

R. Mikut, M. Reischl (Hrsg.)

Institut für Angewandte Informatik /
Automatisierungstechnik



Proceedings

17. Workshop Computational Intelligence

Dortmund,
5.-7. Dezember 2007



R. Mikut, M. Reischl (Hrsg.)

Proceedings 17. Workshop Computational Intelligence

Dortmund, 5.-7. Dezember 2007

Schriftenreihe des
Instituts für Angewandte Informatik / Automatisierungstechnik
an der Universität Karlsruhe (TH)
Band 20

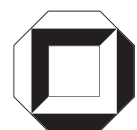
Eine Übersicht über alle bisher in dieser Schriftenreihe erschienenen
Bände finden Sie am Ende des Buchs.

Proceedings

17. Workshop Computational Intelligence

Dortmund, 5.-7. Dezember 2007

R. Mikut
M. Reischl
(Hrsg.)



universitätsverlag karlsruhe

Impressum

Universitätsverlag Karlsruhe
c/o Universitätsbibliothek
Straße am Forum 2
D-76131 Karlsruhe
www.uvka.de



Dieses Werk ist unter folgender Creative Commons-Lizenz
lizenziert: <http://creativecommons.org/licenses/by-nc-nd/2.0/de/>

Universitätsverlag Karlsruhe 2007
Print on Demand

ISSN: 1614-5267
ISBN: 978-3-86644-191-0

VORWORT

Dieser Tagungsband enthält die Beiträge des 17. Workshops „Computational Intelligence“ des Fachausschusses 5.14 der VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik (GMA) und der Fachgruppe „Fuzzy-Systeme und Soft-Computing“ der Gesellschaft für Informatik (GI), der vom 5. – 7. Dezember 2007 im Haus Bommerholz bei Dortmund stattfindet.

Der GMA-Fachausschuss 5.14 „Computational Intelligence“ entstand 2005 aus den bisherigen Fachausschüssen „Neuronale Netze und Evolutionäre Algorithmen“ (FA 5.21) sowie „Fuzzy Control“ (FA 5.22). Der Workshop steht in der Tradition der bisherigen Fuzzy-Workshops, hat aber seinen Fokus in den letzten Jahren schrittweise erweitert.

Die Schwerpunkte sind Methoden, Anwendungen und Tools für

- Fuzzy-Systeme,
- Künstliche Neuronale Netze,
- Evolutionäre Algorithmen und
- Data-Mining-Verfahren

sowie der Methodenvergleich anhand von industriellen und Benchmark-Problemen.

Die Mitglieder des Programmkomitees sind Dr. Mikut (Forschungszentrum Karlsruhe GmbH), Prof. Dr. H.-G. Beyer (FH Vorarlberg), Prof. Dr. A. Kistner (Universität Stuttgart), Prof. Dr. F. Klawonn (FH Braunschweig/Wolfenbüttel), Prof. Dr. A. Kroll (Universität Kassel) und PD Dr. Th. Runkler (Siemens AG).

Die Ergebnisse werden von Teilnehmern aus Hochschulen, Forschungseinrichtungen und der Industrie in einer offenen Atmosphäre intensiv diskutiert. Dabei ist es gute Tradition, auch neue Ansätze und Ideen bereits in einem frühen Entwicklungsstadium vorzustellen, in dem sie noch nicht vollständig ausgereift sind.

Nähere Informationen zum GMA-Fachausschuss erhalten Sie unter

<http://www.iai.fzk.de/medtech/biosignal/gma/index.html>.

Die Herausgeber bedanken sich an dieser Stelle bei allen Autoren und Rednern sowie bei den Mitgliedern des Programmkomitees, die maßgeblich an der Vorbereitung des Workshops beteiligt waren.

Ralf Mikut und Markus Reischl

INHALTSVERZEICHNIS

T. Fober, E. Hüllermeier, M. Mernberger (Philipps-Universität Marburg): Evolutionary Construction of Multiple Graph Alignments for the Structural Analysis of Biomolecules	1
G. Heidemann, S. Klenk (Universität Stuttgart): Visual Analytics for Image Retrieval	15
F. Rügheimer (OvG-Universität Magdeburg): A Condensed Representation for Distributions over Set-Valued Attributes	25
T. Mrziglod (Bayer Technology Services GmbH, Leverkusen): Mit datenbasierten Technologien und Versuchsplanung zu erfolgreichen Produkten	35
H. Schulte (Bosch Rexroth AG, Elchingen): Approximationsgenauigkeit und dynamisches Fehlerwachstum der Modellierung mit Takagi-Sugeno Fuzzy Systemen	42
C. Burghart, R. Mikut, T. Asfour, A. Schmid, F. Kraft, O. Schrempf, H. Holzapfel, R. Stiefelhagen, A. Swerdlow, G. Bretthauer, R. Dillmann (Universität Karlsruhe, Forschungszentrum Karlsruhe GmbH): Kognitive Architekturen für humanoide Roboter: Anforderungen, Überblick und Vergleich	57
R. Mikut, C. Burghart, A. Swerdlow (Forschungszentrum Karlsruhe GmbH, Universität Karlsruhe): Ein Gedankenexperiment zum Entwurf einer integrierten kognitiven Architektur für humanoide Roboter	74
G. Milighetti, H.-B. Kuntze (FhG IITB Karlsruhe): Diskret-kontinuierliche Regelung und Überwachung von Robotern basierend auf Aktionsprimitiven und Petri-Netzen	89
N. Rosemann, W. Brockmann (Universität Osnabrück): Kontrolle dynamischer Eigenschaften des Online-Lernens in Neuro-Fuzzy-Systemen mit dem SILKE-Ansatz	103
A. Hans, D. Schneegaß, A. Schäfer, S. Udluft (Siemens AG, TU Ilmenau): Sichere Exploration für Reinforcement-Learning-basierte Regelung	117

Th. Bartz-Beielstein, M. Bongards, C. Claes, W. Konen, H. Westenberger (FH Köln): Datenanalyse und Prozessoptimierung für Kanalnetze und Kläranlagen mit CI-Methoden	132
S. Nusser, C. Otte, W. Hauptmann (Siemens AG, OvG-Universität Magdeburg): Learning Binary Classifiers for Applications in Safety-Related Domains	139
W. Jakob, A. Quinte, K.-U. Stucky, W. Süß, C. Blume (Forschungszentrum Karlsruhe GmbH; FH Köln, Campus Gummersbach) Schnelles Resource Constrained Project Scheduling mit dem Evolutionären Algorithmus GLEAM	152
M. Preuß, B. Naujoks (Universität Dortmund): Evolutionäre mehrkriterielle Optimierung bei Anwendungen mit nichtzusammenhängenden Pareto-Mengen	165
G. Rudolph, M. Preuß (Universität Dortmund): Ein mehrkriterielles Evolutionsverfahren zur Bestimmung des Phasengleichgewichts von gemischten Flüssigkeiten	177
Y. Chen, O. Burmeister, C. Bauer, R. Rupp, R. Mikut (Universität Karlsruhe, Forschungszentrum Karlsruhe GmbH, Orthopädische Universitätsklinik Heidelberg): First Steps to Future Applications of Spinal Neural Circuit Models in Neuroprostheses and Humanoid Robots	186
F. Hoffmann, J. Braun, T. Bertram, S. Hölemann (Universität Dortmund, RWTH Aachen): Multikriterielle Optimierung mit modellgestützten Evolutionsstrategien	200
S. Piana, S. Engell (Universität Dortmund): Evolutionäre Optimierung des Betriebs von rohrlosen Chemieanlagen	215
T. Runkler (Siemens AG, CT IC 4): Pareto Optimization of the Fuzzy c–Means Clustering Model Using a Multi– Objective Genetic Algorithm	229
H. J. Rommelfanger (J.W. Goethe-Universität Frankfurt am Main): Die Optimierung von Fuzzy-Zielfunktionen in Fuzzy (Mehrziel-) LP- Systemen - Ein kritischer Überblick	242
D. Gamrad, D. Söffker (Universität Duisburg-Essen): Formalisierung menschlicher Interaktionen durch Situations-Operator- Modellbildung	253

S. Ritter, P. Bretschneider (FhG AST Ilmenau): Optimale Planung und Betriebsführung der Energieversorgung im liberalisierten Energiemarkt	260
R. Seising (Medizinische Universität Wien): Heinrich Hertz, Ludwig Wittgenstein und die Fuzzy-Strukturen - Eine kleine „Bildergeschichte“ zur Erkenntnisphilosophie	268
J. Limberg, R. Seising (Medizinische Universität Wien): Sequenzvergleiche im Fuzzy-Hypercube	283
M. Steinbrecher, R. Kruse (OvG-Universität Magdeburg): Visualisierung temporaler Abhängigkeiten in Bayesschen Netzen	298
M. Schneider, R. Tillmann, U. Lehmann, J. Krone, P. Langbein, U. Stark, J. Schrickel, Ch. Ament, P. Otto (FH Südwestfalen, Airbus Deutschland GmbH, Hamburg, TU Ilmenau): Künstliches Neuronales Netz zur Analyse der Geometrie von großflächig gekrümmten Bauteilen	299
C. Frey (FhG IITB Karlsruhe): Prozessdiagnose und Monitoring feldbusbasierter Automatisierungs- anlagen mittels selbstorganisierender Karten	306

Evolutionary Construction of Multiple Graph Alignments for the Structural Analysis of Biomolecules

Thomas Fober, Eyke Hüllermeier, Marco Mernberger

FB Mathematik und Informatik

Philipps-Universität Marburg

Hans-Meerwein-Straße

D-35032 Marburg

Tel.: +49(0)6421 28 - 21586

Fax: +49(0)6421 28 - 21573

E-Mail: {thomas, eyke, mernberger}@mathematik.uni-marburg.de

Abstract

The concept of multiple graph alignment has recently been introduced as a novel method for the structural analysis of biomolecules. Using inexact, approximate graph-matching techniques, this method enables the robust identification of approximately conserved patterns in biologically related structures. In particular, using multiple graph alignments, it is possible to characterize functional protein families independent of sequence or fold homology. This paper first recalls the concept of multiple graph alignment and then addresses the problem of computing optimal alignments from an algorithmic point of view. In this regard, a method from the field of evolutionary algorithms is proposed. Empirically, it is shown that the former yields significantly better results than the latter, albeit at the cost of an increased complexity.

1 Introduction

In the field of bioinformatics, multiple sequence alignment is an established approach for the identification of residues that are conserved across many members of a gene or protein family [13, 4, 12]. However, as this approach relies on evolutionary conserved sequences in DNA or protein chains to detect similarities between different molecules, it is only capable of detecting functional similarities based on heredity. Consequently, sequence analysis is not optimally suited for the identification of *functional* similarities between molecules, as functionality is more closely associated with structural than with sequential features. In fact, it is well-known that there is no strong correspondence between sequence similarity and structural similarity.

Focusing on the the identification of *structural* similarities of biomolecules, this paper presents the concept of *multiple graph alignment* (MGA) as a structural counterpart to sequence alignment. As opposed to homology-based methods, this approach allows one to capture non-homologous molecules with similar functions as well as evolutionary conserved functional domains. Accordingly, MGA offers a much wider field of application, as the method is generally applicable to different types of biological or biochemical objects (and, in principle, even to graph-based models beyond the bioinformatics domain). In this regard, our special interest concerns the analysis of protein structures or, more specifically, protein binding sites. However, graph alignments can also be used for analyzing other types of biomolecules, e.g., to detect similarities between chemical molecules such as protein ligands (see Section 4.2).

Our previous approach to the MGA problem makes use of an algorithm which employs a simple greedy strategy to construct alignments in an incremental way [14]. In this paper, we present an alternative method using evolutionary strategies. As will be shown experimentally, the latter significantly outperforms the former with regard to the quality of alignments, albeit at the cost of an increased runtime.

The paper is organized as follows: In Section 2, we introduce the concept of a multiple graph alignment. The problem of computing an MGA is then addressed in Section 3, where an evolutionary algorithm is proposed for this purpose. Section 4 is devoted to the experimental validation of the approach, and Section 5 concludes the paper.

2 Graph-Based Modeling and Multiple Graph Alignment

2.1 Graph-Based Modeling of Biomolecules

Graph models are often used to represent and analyze three-dimensional biomolecules, i.e., single biomolecules are represented in terms of a graph G consisting of a set of (labeled) nodes V and (weighted) edges E . As mentioned previously, our special interest concerns protein binding pockets and chemical compounds.

2.1.1 Modeling Chemical Compounds

In the case of chemical compounds, atoms are represented as nodes labeled with their corresponding atom type, e.g., using the SYBYL atom type notation. The edges between nodes are labeled by the Euclidean distance between the respective atoms. This way, important geometric properties of the compound are captured. Alternatively, it is possible to use a representation in which edges correspond to molecular bonds and are weighted by the bond order.

2.1.2 Modeling Protein Binding Sites

Regarding the modeling of protein structures, the present work builds upon Cavbase [11, 9], a database system for the fully-automated detection and extraction of protein binding pockets from experimentally determined protein structures (available through the database PDB [6]). In Cavbase, graphs are used as a first approximation to describe binding pockets. The database currently contains 113,718 hypothetical binding pockets that have been extracted from 23,780 publicly available protein structures using the LIGSITE-algorithm [10].

To model a binding pocket as a graph, the geometrical arrangement of the pocket and its physicochemical properties are first represented by pre-defined *pseudocenters* – spatial points that represent the center of a particular property. The type and the spatial position of the centers depend on the amino acids that border the binding pocket and expose their functional groups. They are derived from the protein structure using a set of pre-defined rules [11]. As possible types for pseudocenters, hydrogen-bond donor, acceptor, mixed donor/acceptor, hydrophobic aliphatic and aromatic properties are considered. Pseudocenters can be regarded as a compressed representation of areas on the cavity surface

where certain protein-ligand interactions are experienced. Consequently, a set of pseudocenters is an approximate representation of a spatial distribution of physicochemical properties.

The assigned pseudocenters form the nodes $v \in V$ of the graph representation, and their properties are modeled in terms of node labels $l(v) \in \{P1, P2 \dots P5\}$, where P1 stands for donor, P2 for acceptor, etc. Two centers are connected by an edge in the graph representation if their Euclidean distance is below 11.0 Å and each edge $e \in E$ is labeled with the respective distance $w(e) \in \mathbb{R}$.¹ The edges of the graph thus represent geometrical constraints among points on the protein surface.

In Cavbase, a graph representation of a binding pocket has around 85 nodes on average; however, also graphs with several hundred nodes are frequently detected and extremes with thousands of nodes exist. The graphs are rather dense as approximately 20 percent of all pairs of nodes are connected by an edge.

2.2 Multiple Graph Alignment

In the following, we assume a set $\mathcal{G} = \{G_1(V_1, E_1) \dots G_n(V_n, E_n)\}$ of connected, node-labeled and edge-weighted graphs to be given, each of which represents a biomolecule. To make the discussion more concrete, we subsequently consider the case of protein binding sites.

When comparing homologs from different species in protein cavity space, one has to deal with the same mutations that are also given in sequence space. Corresponding mutations, in conjunction with conformational variability, strongly affect the spatial structure of a binding site as well as its physicochemical properties and, therefore, its graph descriptor. Thus, one cannot expect that the graph descriptors for two related binding pockets match exactly. Our approach includes the following types of edit operations to account for differences between a graph $G_1(V_1, E_1)$ and another graph $G_2(V_2, E_2)$:

1. Insertion or deletion of a node $v_1 \in V_1$ (“InDel”). A pseudocenter can be deleted or introduced due to a mutation in sequence space. Alternatively, an insertion or deletion in the graph descriptor can result from a conformational difference that affects the exposure of a functional group toward the binding pocket.
2. Change of the label $l(v_1)$ of a node $v_1 \in V_1$ (“Node Mismatch”). The assigned physicochemical property (“type”) of a pseudocenter can change if a mutation replaces a certain functional group by another type of group at the same position.
3. Change of the weight $w(e_1)$ of an edge $e_1 \in E_1$ (“Edge Mismatch”). The distance between two pseudocenters can change due to conformational differences.

By assigning a cost value to each of these edit operations, it becomes possible to define an edit distance for a pair of graph descriptors. The edit distance of two graphs G_1, G_2 is defined as the cost of a cost-minimal sequence of edit operations that transforms graph G_1 into G_2 . As in sequence analysis, this allows for defining the concept of an alignment

¹An interaction distance of 11.0 Å is typically enough to capture the geometry of a binding site, and ignoring larger distances strongly simplifies the graph representation.

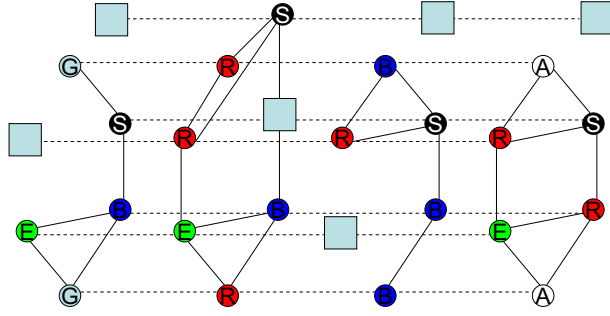


Figure 1: An alignment of four similar, but not identical graphs. The node labels are indicated by the letters assigned to the nodes (shown as circles). The edge labels are omitted for simplification. The assignments among the different graph nodes are indicated by the dashed lines. Large boxes in gray represent dummy nodes that have been introduced in the alignment to represent “missing” nodes.

of two (or more) graphs. The latter, however, also requires the possibility to use dummy nodes \perp that serve as placeholders for deleted nodes. They correspond to the gaps in sequence alignment (cf. Figure 1).

Let $\mathcal{G} = \{G_1(V_1, E_1) \dots G_n(V_n, E_n)\}$ be a set of graphs. Then

$$\mathcal{A} \subseteq (V_1 \cup \{\perp\}) \times \dots \times (V_n \cup \{\perp\})$$

is an alignment of the graphs in \mathcal{G} if and only if

1. for all $i = 1 \dots n$ and for each $v \in V_i$ there exists exactly one $a = (a_1 \dots a_n) \in \mathcal{A}$ such that $v = a_i$ (i.e., each node of each graph occurs exactly once in the alignment).
2. for each $a = (a_1 \dots a_n) \in \mathcal{A}$ there exists at least one $1 \leq i \leq n$ such that $a_i \neq \perp$ (i.e., each tuple of the alignment contains at least one non-dummy node).

Each tuple in the alignment contains at least one node from each graph and indicates that these nodes are matched with each other. If there is no matching partner in a certain graph, a dummy node \perp must be inserted.

To assess the quality of a given alignment, a scoring function is needed. This scoring function corresponds to the above-mentioned edit distance, as each graph alignment defines a set of edit operations that have to be performed to transform one of the aligned graphs into another entry of the alignment. Here, a scoring function that follows a sum-of-pairs scheme is proposed, i.e., the score s of a multiple alignment $\mathcal{A} = (a^1 \dots a^m)$ is defined by the sum of scores of all induced pairwise alignments:

$$s(\mathcal{A}) = \sum_{i=1}^m \text{ns}(a^i) + \sum_{1 \leq i < j \leq m} \text{es}(a^i, a^j), \quad (1)$$

Parameter	Value
ns _m	1.0
ns _{mm}	-5.0
ns _{dummy}	-2.5
es _m	0.2
es _{mm}	-0.1

Table 1: Parameter setting used in the experiments.

where the *node score* (ns) is given by

$$\text{ns} \left(\begin{pmatrix} a_1^i \\ \vdots \\ a_n^i \end{pmatrix} \right) = \sum_{1 \leq j < k \leq n} \begin{cases} \text{ns}_m & l(a_j^i) = l(a_k^i) \\ \text{ns}_{mm} & l(a_j^i) \neq l(a_k^i) \\ \text{ns}_{dummy} & a_j^i = \perp, a_k^i \neq \perp \\ \text{ns}_{dummy} & a_j^i \neq \perp, a_k^i = \perp \end{cases}$$

Comparing two edges is somewhat more difficult than comparing two nodes, as one cannot expect to observe edges of exactly the same lengths. We consider two edges as a match if their respective lengths, a and b , differ by at most a given threshold t_{max} , and as a mismatch otherwise. The *edge score* (es) is then given by

$$\text{es} \left(\left(\begin{pmatrix} a_1^i \\ \vdots \\ a_n^i \end{pmatrix}, \begin{pmatrix} a_1^j \\ \vdots \\ a_n^j \end{pmatrix} \right) \right) = \sum_{1 \leq k < l \leq n} \begin{cases} \text{es}_{mm} & (a_k^i, a_l^j) \in E_k, (a_l^i, a_k^j) \notin E_l \\ \text{es}_{mm} & (a_k^i, a_l^j) \notin E_k, (a_l^i, a_k^j) \in E_l \\ \text{es}_m & d_{kl}^{ij} \leq t_{max} \\ \text{es}_{mm} & d_{kl}^{ij} > t_{max} \end{cases}$$

where $d_{kl}^{ij} = \|w(a_k^i, a_l^j) - w(a_l^i, a_k^j)\|$. The scores for the different cases (i.e., ns_m, ns_{mm}, ns_{dummy}, es_m, es_{mm}) have to be defined by the user. Table 1 shows the parameter setting that was used for the experiments in Section 4.

2.3 Computing Multiple Graph Alignments

The problem of calculating an optimal MGA, that is, an alignment with maximal score for a given set of graphs, is computationally very complex. The subgraph isomorphism problem (which is known to be NP-complete [1]) can be seen as a special case of the graph alignment problem where the cost for mismatches is set prohibitively high. Thus, one cannot expect to find an efficient algorithm that is guaranteed to find an optimal alignment for a given set of graphs. In [14], simple and effective heuristics for the MGA problem have been described that were found to be useful for the problem instances that were examined. The main idea of these methods is to reduce the multiple alignment problem to the problem of pairwise alignment (i.e., calculating an optimal graph alignment for only two graphs) in a first step; these pairwise alignments are subsequently merged into a multiple alignment in a second step.

Both steps, finding optimal pairwise alignments and merging them into a multiple alignment, are only heuristics that do not guarantee to find a good solution on every problem

instance. Even worse, as there is no information about the distance to the optimum, it is hardly possible to get an idea about the quality a solution produced by this strategy.

Motivated by these problems, we investigated the use of evolutionary algorithms as an alternative approach. As will be detailed in Section 3, we resorted to an evolution strategy with a non-standard representation and examined two different approaches: While the first approach seeks to optimize a complete multiple alignment directly, the second approach sticks to the decomposition strategy that is also used by the greedy algorithm and, hence, uses evolutionary optimization only for the pairwise case.

On the one hand, evolutionary optimization is of course more expensive from a computational point of view. On the other hand, the hope is that this approach will be able to improve the solution quality, i.e., to produce alignments that are better than those obtained by a simple greedy strategy. To see if the increased cost of an evolutionary algorithm is justified, we have also compared it with a simple hill-climbing strategy.

3 An Evolutionary Algorithm for Multiple Graph Alignment

Evolution strategies (ES), which are often used to solve real-valued optimization problems, simulate the evolution process by executing a loop [5]. Among the well-defined representations and genetic operators in ES, we are mainly interested in the selection operator, while the recombination and mutation operators have to be redefined as we make use of a non-standard representation. The deterministic plus-selection is very convenient for our purpose. Since the search space of an MGA problem has a size of $\mathcal{O}(m!^{n-1})$, with n the number of structures and m the number of nodes of a single structure, it would be very unfortunate to lose a current best solution. Therefore, other selection techniques such as fitness-proportional or simulated-annealing selection are not applicable in our case.

A first requirement for evolutionary optimization is an appropriate representation of individuals, i.e., candidate solutions. In this regard, note that, given a fixed numbering from 1 to $|V_i|$ (not to be confused with the labeling) of the nodes of each graph G_i , an MGA can be represented in a unique way by a two-dimensional matrix, where the rows correspond to the graphs and the columns to the aligned nodes (possibly a dummy) of these graphs. Fig. 2 shows an example of such a matrix. The first column indicates a mutual assignment of the first node of graph A, the sixth node of graph C, and the fourth node of graph D, while there is no matching partner other than a dummy in graph B.

For simplicity, the number of columns of the matrix is held fixed as alignments of different size are difficult to compare using the scoring function defined above. If a column of the alignment contains only dummy nodes, it is excluded from the scoring and considered as neutral. Such improper columns are allowed only during the optimization procedure but are discarded for the final solution. To guarantee that no constraints are enforced on the solutions, the number of columns is set to a sufficiently high number (otherwise, it could be necessary to map nodes in a suboptimal way).

Initially, each row of the matrix is filled with a random permutation over the specified column length (which, for the i -th row, corresponds to the number of nodes $|V_i|$ of graph G_i), and the remaining entries (exceeding $|V_i|$) are filled with a dummy. Additionally, the fitness and a strategy component are added to the individual. Storing the fitness has

A:	1	2	3	4	.	6	
B:	.	3	.	4	5	.	1	6	.	.	2
C:	6	2	.	.	5	.	.	.	1	.	.	3
D:	4	.	.	1	5	6	.	.	3

Figure 2: Matrix representation of an MGA. Dummies are represented by a dot.

runtime benefits as it avoids re-calculation. The strategy component is introduced to enable the adaptation of the mutation rates.

The overall structure of the evolutionary algorithm is as follows:

- Initially, a population consisting of μ random graph alignments is generated; μ is an external parameter that specifies the *population size* per generation.
- In each generation, $\lambda = \nu \cdot \mu$ offspring individuals are created; the parameter ν is called *selective pressure*. To generate a single offspring, the mating-selection operator chooses two parent individuals at random and submits them to the *recombination* operator. This operator generates an offspring by exchanging the information of both individuals. The new individual is further modified by the *mutation* operator.
- The offsprings are evaluated and added to the parent population. Among the individuals in this temporary population T , the plus-selection chooses the best μ candidates, which then form the population of the next generation.
- The whole procedure is repeated until no further improvements are achieved during several generations.

As mentioned above, the recombination and mutation operators must be adapted to the representation, which is in our case a matrix scheme. Our recombination operator chooses a row r of the matrix scheme at random and cuts both individuals at this position, thereby producing four sub-matrices. An offspring is then created by combining the upper block of the first individual (first sub-matrix), consisting of r rows, with the lower block of the second individual (fourth sub-matrix), consisting of $n - r$ rows. Simply switching the complete blocks is not possible, however, since the nodes are not ordered in a uniform way. Instead, to assign the columns of the first sub-matrix to columns of the fourth sub-matrix in a correct way, we use the elements of the selected row as pivot elements: If the k -th node of the r -th graph appears in column i in the first individual and in column j in the second one, the i -th row of the first sub-matrix is combined with the j -th row of the fourth sub-matrix. General experience has shown that recombination increases the speed of convergence, and this was also confirmed by our experiments (see Section 4).

The mutation operator selects one row and two columns at random and swaps the entries in the corresponding cells. To enable large mutation steps, this procedure is repeated multiple times for each individual. As the optimal number of repetitions was unknown in the design phase of the algorithm, it was specified as a strategy component [5]; see Section 4.1.

3.1 Combining Evolutionary Optimization and Pairwise Decomposition

As mentioned above, the search space of an MGA problem grows exponentially with the number of graphs. Moreover, the evaluation time of the fitness function grows quadratically with the number of graphs. Therefore, the application of our EA algorithm becomes extremely expensive for large problems.

One established strategy to reduce complexity is to decompose a multiple alignment problem into several pairwise problems and to merge the solutions of these presumably more simple problems into a complete solution. This strategy has already been exploited in our greedy approach, where the merging step has been realized by means of the star-alignment algorithm [14]. In star-alignment, a center structure is first determined, and this structure is aligned with each of the other $n - 1$ structures. The $n - 1$ pairwise alignments thus obtained are then merged by using the nodes of the center as pivot elements. As the quality of an MGA derived in this way critically depends on the choice of a suitable center structure, one often tries every structure as a center and takes the best result. In this case, all possible pairwise alignments are needed, which means that our evolutionary algorithm must be called $\frac{1}{2}(n^2 - n)$ times.

As star-alignment is again a purely heuristic aggregation procedure, the gain in efficiency is likely to come along with a decrease in solution quality, compared with the original EA algorithm. This is not necessarily the case, however. In fact, a decomposition essentially produces two opposite effects, a positive one due to a simplification of the problem and, thereby, a reduction of the search space, and a negative one due to a potentially suboptimal aggregation of the partial solutions. For a concrete problem, it is not clear in advance which among these two effects will prevail. Roughly speaking, it is well possible that constructing good pairwise alignments and aggregating them in an ad-hoc way is better than getting astray in a huge search space of multiple alignments.

4 Experimental Results

This section presents two experimental studies. The purpose of the first study was to optimize our evolutionary algorithm by means of a proper adjustment of its exogenous parameters, which have a strong influence on performance and runtime. In our case, the following parameters are concerned:

- the population size μ and the selective pressure ν ;
- selfadaption and recombination, which can assume values {on, off}, and allow the automatic step size control and the recombination operator to be enabled or disabled, respectively;
- initial step size, which defines the initial step size for the mutation; if the automatic step size control is disabled, this parameter is ignored and a constant step size of 1 is used for the mutation.

In the second study, we used our algorithms to analyze two data sets with different characteristics, the first consisting of a set of small molecules (benzamidine) and the second of a set of protein binding sites (thermolisin). For comparison purpose, we also included a simple hill-climbing strategy, namely a $(1 + 1)$ -EA.

4.1 Optimizing the Evolutionary Algorithm

The main goal of this study was to reduce the runtime of the EA without sacrificing solution quality. To this end, we have experimented with the *sequential parameter optimization toolbox* (SPOT) [2] that enables a quasi-automatic adjustment of exogenous parameters. The standard parametrization by Bäck [3] gives a first idea of a useful range. We have chosen a range between 1 and 50 for the population size μ and a range between 1 and 20 for the selective pressure ν . For initial step size, we allowed values between 1 and $\frac{1}{2} \times \text{columnlength}$.

Test problems were produced by generating a random graph first, and replicating this graph (with renumbered nodes) $n-1$ times afterward. Obviously, the optimal solution of a problem of that kind is a perfect match of these m graphs; we modified the scoring system (by a linear transformation) such that the score of this solution is 0. SPOT generates a sequence of design points in the parameter space with the goal to minimize the number of required function evaluations for reaching a fitness of zero. The experiments were conducted for alignment problems of size $n \in \{2, 4, 8, 16\}$ which, however, all yielded similar results. According to these results, the parameter configuration shown in Table 2 seems to be well-suited for the problem.

parameter	μ	ν	selfadaption	recombination	initial step size
value	4	20	off	on	21

Table 2: Optimal setting of the exogenous parameters of the EA.

As can be seen, a small value for the population size (only large enough to enable recombination) is enough, probably due to the fact that local optima do not cause a severe problem. On the other hand, as the search space is extremely large, a high selective pressure is necessary to create offsprings with improved fitness. The self-adaptation mechanism is disabled and, hence, the mutation rate is set to one (only two cells are swapped by mutation). This appears reasonable, as most swaps do not yield an improvement and instead may even produce a deterioration, especially during the final phase of the optimization. Thus, an improvement obtained by swapping two cells is likely to be annulled by a second swap in the same individual. Finally, our experiments suggest that a recombination is very useful and should therefore be enabled.

4.2 Mining Molecular Fragment Data

As a first proof-of-concept for the algorithms presented in the previous section, we analyzed a data set consisting of 87 compounds that belong to a series of selective thrombin inhibitors and were taken from a 3D-QSAR study [7]. The data set is suitable for conducting experiments in a systematic way, as it is quite homogeneous and relatively small (the graph descriptors contain 47–100 nodes, where each node corresponds to an atom). Moreover, as the 87 compounds all share a common core fragment (which is distributed over two different regions with a variety of substituents), the data set contains a clear and unambiguous target pattern. From this data set, 100 subsets of 2, 4, 8 and 16 compounds have been selected at random, and for each subset, an MGA has been calculated using the greedy heuristic (Greedy), our evolutionary algorithm with standard parametrization

(EA_{std}), optimized parametrization (EA_{opt}), and in combination with a star-alignment procedure (EA_{opt}^{*}).

Before performing these experiments, we have compared the optimized EA with a simple hill-climbing strategy, namely a (1 + 1)-EA. To ensure a fair comparison, each search strategy was allowed a fixed number of fitness function evaluations. The results, which are summarized in Table 3, are clearly in favor of EA_{opt} and, therefore, justify the use of an EA instead of a conceptually more simple hill-climber.

	2 graphs		4 graphs	
	EA _{opt}	(1 + 1)-EA	EA _{opt}	(1 + 1)-EA
best	46.1	-143.8	95.5	-1069.1
median	-41.1	-213.4	-93.7	-1129.8
worst	-132.6	-257.9	-304.3	-1307.0
mean	-22.2	-191.425	-98.4125	-1152.4
std	74.7106	47.6079	144.6097	87.4351

	8 graphs		16 graphs	
	EA _{opt}	(1 + 1)-EA	EA _{opt}	(1 + 1)-EA
best	-128.8	-5300.4	-3854.3	-29597
median	-776.2	-6157.9	-6043.2	-35780
worst	-971.8	-8361.6	-8355.8	-36577
mean	-539.25	-6484.2	-5888.0	-33004
std	352.1576	1156.4	1626.4	3249.4

Table 3: Comparison between a simple hill-climbing strategy and an optimized EA.

The results of the main experiment, comparing the three EA-variants with the greedy strategy, are shown in Table 4. As a measure of comparison, we derived the relative improvement of the fitness value, defined as

$$\frac{\text{fitness(EA)} - \text{fitness(Greedy)}}{|\text{fitness(Greedy)}|} \quad (2)$$

The results confirm our expectations: Even in its standard setting, the EA significantly outperforms the greedy algorithm, and the optimized version even improves upon these results. In general, the full EA-variants also perform better than EA_{opt}^{*}, even though the 4-graph case is a notable exception.

The runtimes, shown in Table 5, confirm that EA_{opt} does not only produce better results but is also more efficient than EA_{std}. Of course, the smallest runtime among all alternatives is still produced by the greedy strategy. A good compromise between solution quality and efficiency is achieved by EA_{opt}^{*}.

Regarding the aforementioned benzamidine core fragment, which consists of 25 atoms (11 hydrogens), it turned out that it was fully conserved, throughout all experiments, in all the solutions produced by our EA. For the greedy strategy, the corresponding degree of conservation was significantly smaller.

4.3 Mining Protein Binding Pockets

As our main interest concerns the characterization of protein binding pockets, we subsequently examined the performance of our algorithms on a second data set consisting of 74

	2 graphs	4 graphs	8 graphs	16 graphs
best	2.6522	2.4931	0.9819	0.9061
median	0.5798	0.7641	0.8960	0.8551
worst	0.3183	0.5536	0.8517	0.8298
mean	1.0286	1.0245	0.9198	0.8692
std	0.7279	0.6150	0.0437	0.0289
best	3.2271	2.1952	0.9668	0.9060
median	0.5798	0.8798	0.9136	0.8616
worst	0.5260	0.5536	0.8721	0.8350
mean	1.3853	1.2043	0.9174	0.8721
std	1.1326	0.5673	0.0313	0.0264
best	3.2271	11.9850	1.1423	0.6881
median	0.5798	1.0864	0.7441	0.5423
worst	0.5260	0.7967	0.7074	0.5350
mean	1.3853	3.7989	0.8761	0.6042
std	1.1326	4.8849	0.1758	0.0678

Table 4: Relative improvement for EA_{std} (top), EA_{opt} (middle), and EA_{opt}^* (bottom) in comparison with Greedy.

	2 graphs				4 graphs			
	Greedy	EA_{std}	EA_{opt}	EA_{opt}^*	Greedy	EA_{std}	EA_{opt}	EA_{opt}^*
best	0.219	9.766	7.703	6.934	0.953	86.255	78.692	57.383
median	0.250	14.094	11.953	12.540	1.141	129.740	106.624	80.394
worst	0.296	24.735	25.015	27.548	1.406	197.837	230.401	105.536
mean	0.252	14.804	12.651	12.874	1.1601	137.33	126.10	79.938
std	0.0241	4.177	4.7445	5.151	0.165	30.955	44.896	13.103
	8 graphs				16 graphs			
	Greedy	EA_{std}	EA_{opt}	EA_{opt}^*	Greedy	EA_{std}	EA_{opt}	EA_{opt}^*
best	4.687	1042.92	768.299	320.456	18.609	6568.8	3630.5	693.83
median	5.141	1978.23	1048.41	375.024	20.563	8643.8	5249.2	775.74
worst	6.766	2957.48	4009.34	474.978	28.407	21411.8	16008.5	1731.6
mean	5.553	1945.35	1305.6	383.86	22.296	11275	7102.9	999.69
std	0.790	533.08	784.74	42.494	3.7992	4904.3	3870.9	407.61

Table 5: Comparison of runtimes of the search strategies.

structures derived from the Cavbase database. Each structure represents a protein binding pocket belonging to the protein family of thermolysine, bacterial proteases frequently used in structural protein analysis and annotated with the E.C. number 3.4.24.27 in the ENZYME classification database. The data set is suited for our purpose, as all binding pockets belong to the same enzyme family and, therefore, should share evolutionary related, highly conserved substructures. On the other hand, with binding pockets ranging from about 30 to 90 pseudocenters, the data set is also diverse enough to present a real challenge for graph matching techniques.

Again, 100 graph alignments of size 2, 4, 8, and 16 were produced, respectively, for randomly chosen structures, and the results are summarized in Table 6. The improvements achieved by the EA are not as high as in the previous experiment, but still significant. The most important difference is observed for the star-alignment approach, which now performs rather poorly. This is probably due to the much higher diversity of the structures in this experiment, which makes an aggregation of sub-alignments more difficult, and a heuristic like star-alignment more likely to fail.

	2 graphs		4 graphs		8 graphs		16 graphs	
	EA _{opt}	EA _{opt} [*]	EA _{opt}	EA _{opt} [*]	EA _{opt}	EA _{opt} [*]	EA _{opt}	EA _{opt} [*]
best	0.753	0.749	0.739	0.1391	0.786	0.226	0.7887	0.1713
median	0.576	0.576	0.732	0.0898	0.725	0.070	0.7523	0.1013
worst	0.434	0.431	0.713	0.0375	0.721	0.047	0.7474	0.0812
mean	0.595	0.594	0.728	0.0869	0.755	0.145	0.7613	0.1228
std	0.117	0.115	0.01	0.0344	0.032	0.077	0.0171	0.0342
best	14.1	13.01	895.3	517.9	8971.7	2147.1	16005.0	2269.0
median	65.3	71.05	2109.1	713.6	21878.9	2861.0	51198.7	3039.7
worst	267.9	259.18	3112.8	970.7	39681.1	5207.7	93706.2	3493.1
mean	95.4	93.48	2041.5	736.83	21929.0	3438.3	50665.5	2972.1
std	85.1	78.17	679.46	103.15	9765.7	1097.6	25488	383.98

Table 6: Relative improvement (upper block) and runtime (lower block) of EA_{opt} and EA_{opt}^{*} in comparison with Greedy.

5 Conclusions

Multiple graph alignment (MGA) has recently been introduced as a novel method for analyzing biomolecules on a structural level. Using robust, noise-tolerant graph matching techniques, MGA is able to discover approximately conserved patterns in a set of graph-descriptors representing a family of evolutionary related biological structures. As the computation of optimal alignments is a computationally complex problem, this paper has proposed an evolutionary algorithm (EA) as an alternative to a hitherto existing greedy strategy.

Our experiments, carried out on several data sets with different characteristics, have shown the high potential of this approach and give rise to the following conclusions: The EA significantly outperforms the greedy strategy and is able to produce alignments of much higher quality, albeit at the cost of a considerable increase in runtime. In fact, as runtime seems to increase exponentially with the number of graphs to be aligned, we also considered a combination of evolutionary optimization with decomposition techniques,

namely the star-alignment method. This approach appears to be a reasonable compromise as it achieves a good trade-off between solution quality and runtime.

As part of ongoing work, we currently elaborate on decomposition techniques in more detail. In particular, we investigate alternative decomposition schemes and aggregation procedures. An especially interesting idea in this regard is to replace a heuristic aggregation strategy again by an evolutionary optimization step. This line of research is complemented by more standard means to improve evolutionary optimization, such as alternative representations and more sophisticated genetic operators.

References

- [1] Atallah, M. J. (editor): *Algorithms and Theory of Computation Handbook*. CRC Press LLC. 1999.
- [2] Bartz-Beielstein, T., Lasarczyk, C. and Preuß, M.: Sequential Parameter Optimization Toolbox. *Technical Report, Universität Dortmund, Germany*. 2006.
- [3] Bäck, T.: Evolutionary Algorithms in Theory and Practise. *Dissertation, Universität Dortmund, Germany*. 1994.
- [4] Bateman, A., Coin, L., Durbin, R., Finn, R. D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E. L. L., Studholme, D. J., Yeats, C. and Eddy, S. R.: The Pfam protein families database. *Nucl. Acids. Res.*, 32(90001):D138–141. 2004.
- [5] Beyer, H.-G. and Schwefel, H.-P.: Evolution strategies – A comprehensive introduction. *Natural Computing*, 1(1):3–52. 2002.
- [6] Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. and Bourne, P.E.: The protein data bank. *Nucleic Acids Research*, 28:235–242. 2000.
- [7] Böhm, M., Stürzebecher, J. and Klebe, G.: Three-dimensional quantitative structure-activity relationship analyses using comparative molecular field analysis and comparative molecular similarity indices analysis to elucidate selectivity differences of inhibitors binding to trypsin, thrombin, and factor xa. *Journal of Medicinal Chemistry*, 42(3):458–477. 1999.
- [8] Bron, C. and Kerbosch, J.: Algorithm 457: Finding All Cliques of an Undirected Graph. *Communications of the ACM*, 16(9):575–577. 1973.
- [9] Hendlich, M., Bergner, A., Günther, J. and Klebe, G.: Relibase: Design and Development of a Database for Comprehensive Analysis of Protein-Ligand Interactions. *Journal of Molecular Biology*, 326:607–620. 2003.
- [10] Hendlich, M., Rippmann, F. and Barnickel, G.: LIGSITE: Automatic and efficient detection of potential small molecule-binding sites in proteins. *Journal of Molecular Graphics and Modelling*, 15:359–363. 1997.

- [11] Schmitt, S., Kuhn, D. and Klebe, G.: A New Method to Detect Related Function Among Proteins Independent of Sequence and Fold Homology. *J. Mol. Biol.*, 323(2):387–406. 2002.
- [12] Servant, F., Bru, C., Carrère, S., Courcelle, E., Gouzy, J., Peyruc, D. and Kahn, D.: Prodom: Automated clustering of homologous domains. *Briefings in Bioinformatics*, 3(3):246–251. 2002.
- [13] Thompson, J. D., Higgins, D. G. and Gibson T. J.: Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22:4673–4680. 1994.
- [14] Weskamp, N., Hüllermeier, E., Kuhn, D. and Klebe, G.: Graph Alignments: A New Concept to Detect Conserved Regions in Protein Active Sites. *German Conference on Bioinformatics 2004*, 131–140. 2004.

Visual Analytics for Image Retrieval

Gunther Heidemann and Sebastian Klenk

Intelligent Systems Group, University of Stuttgart
Universitätsstrasse 38
D-70569 Stuttgart, Germany
E-Mail: ais@vis.uni-stuttgart.de

Abstract

Content Based Image Retrieval (CBIR) applies pattern recognition methods to search for images within a database. The main problem of CBIR is that to date only low level image features are available as similarity measures. Therefore, it is not yet possible to search for scenes or objects defined on the level of world knowledge. The application of current similarity measures is based just on the hope for a correlation of low level features with the expectations of the user. While most research concentrates on overcoming this so called “semantic gap” by systems that work on a higher level of abstraction, we point out an alternative. Instead of improving the intelligence of the machine, we want to draw upon the intelligence of the user by advanced user–system interaction. A key component in this is the visualization of the low-level features CBIR is based on to provide the user with an understanding what is actually being searched for. Motivated by first experiments, we describe a research agenda for the incorporation of visualization methods in CBIR systems.

1 Introduction

The overwhelming number of digital images available today requires powerful management and retrieval system. But manual indexing or categorization of images is not only extremely labor intensive, but also problematic from a principled point of view, since categories are difficult to establish. In particular, the definition of dominant categories are difficult to find for a given image. Imagine a scene of a street in a city: What image components are relevant for indexing? Street? Cars, people, shops, a dog, traffic lights, the gray sky? Even more challenging is the question about more abstract categories the image might stand for, such as *traffic*, *daily life*, *shopping* or *pollution*.

To avoid manual indexing, most retrieval systems are content-based, such as QBIC [1], VisualSEEK [2], Photobook [3], Virage [4], Blobworld [5], PicToSeek [6] or SIMPLIcity [7], to name but a few. The field is called Content Based Image Retrieval (CBIR) since the content of the pixel image itself is exploited. Most popular are query-based retrieval systems, where the user specifies one or more sample images and the CBIR system returns those images from a database which are judged to be similar. In this paper we will first outline the fundamental problems of this popular approach. We will argue that there is no realistic chance to overcome the so called *semantic gap* within the foreseeable future. As a consequence, we propose a novel approach, which keeps the basic idea of CBIR but puts a much stronger focus on Human–Computer Interaction (HCI) and visualization. This approach is in the line of the upcoming field of Visual Analytics, which aims at a “synergy” of visualization techniques and human pattern recognition capabilities.

Section 2 describes the existing CBIR approach and its specific problems, section 3 discusses two alternatives of future development. Motivated by earlier results (section 4), section 5 suggests how Visual Analytics might fill the semantic gap, section 6 outlines the corresponding user–system interaction.

2 Problems of today’s CBIR approach

The majority of contemporary CBIR systems is query-image based, i.e. the request of the user is expressed in terms of one or more sample images. This approach leads to three different problems: The specification of queries, the semantic gap, and the lack of system feedback. We will discuss these problems separately.

2.1 Query specification

Specifying the goal of a search using query images entails some obvious difficulties: It is assumed that the user already has similar images. This assumption is somewhat absurd since if he/she has a perfect image, it’s unlikely he/she is looking for just another one. In other words: Most likely the user does *not* have an ideal representation of the scene in question, only an image that resembles the one searched for in certain aspects. So the target has to be specified using multiple images, but now the question arises, which components are really relevant? Picking relevant components from different images is not only difficult to implement but also problematic since a user may refer to entirely different levels of abstraction by selecting components.

As an example, let’s assume the user is looking for a red sports car in front of a beach. Even if a query image of exactly this kind is available, how should a CBIR system know what is being asked? The exact scene (then why not use the query image?) ? Or a red car, a beach, a good-weather image? Abstract categories such as *vacation* or *luxury*? Let’s assume it is the exact scene that is being searched for, but the user has only an image of a blue sports car in a city, an empty beach, and a red truck. In this case, both physical and abstract components (color) have to be combined. Though this can be realized by software such as Photoshop, composing a search on the image level would be off the mark, since a CBIR system does not search for images but for features. It should be noted that the problem of appropriate queries is not only one of human-machine interaction, caused by a lack of understanding on the side of the machine — a human would experience the same difficulties when a scene of multiple, equally dominant components is presented.

2.2 The semantic gap

The *semantic gap* denotes the number one problem of CBIR: Users are thinking in high level categories while machines can only recognize low level image features. Even if we neglect really high level concepts such as *luxury*, the most down-to-earth level a non-expert user is ready to think of is the level of physical objects, such as *red car*, *sand* or *tree*. But a machine has no understanding whatsoever of cars — this is the semantic gap. The CBIR system “thinks” on the level of features such as colored regions with shape feature, textures, color histograms or local projections onto selected function systems.

To make the difficulties caused by the semantic gap more clear, let's consider an attempt to "bridge the gap" between features and physical objects. Modeling a physical object from low level features requires considerable effort, since mere 2D geometric relations among features are, evidently, insufficient in most cases. Rather, a 3D object model is required which relates the object geometry and surface properties to its appearance in a 2D projection. Such models are extremely difficult to obtain even for rigid objects, since not only the shape but the surface reflectance properties have to be modeled. But even if a complete 3D-surface model exists, the 2D appearance still depends on the illumination conditions of the scene. For these reasons, a geometry-based bridge between features and models is difficult to build. Therefore, the appearance-based approach has become popular in object recognition, which relies on capturing the pixel appearance (e.g. [8, 9]). The same idea — specifying a scene from appearance properties instead of models — underlies the query-image approach of CBIR, so we are back at the semantic gap.

After this outline of the two basic CBIR problems, we will discuss future perspectives in the next sections.

3 Parting of the ways in CBIR

There are two ways how CBIR research can deal with the above mentioned issues:

1. We can hope for the solution of the general computer vision problem, i.e. the elimination of the semantic gap. Once a computer has an understanding (whatever this means) of real world images, its internal representation is probably sufficiently advanced that also query specification is no longer a problem.
2. We can accept the semantic gap for the time being and look for ways to exploit best existing low level features.

We did not mention an ever further development of low level pattern recognition as a third option, since, in our opinion, this will not bridge the semantic gap and not lead to solutions differing significantly from the state of the art.

As artificial intelligence is probably decades (or more) from solving the general computer vision problem, we prefer the second option. So the question is: How can a user get the most out of a CBIR system based on today's features and pattern recognition methods? In other words, what are the main obstacles on the way from query images to a result?

To date, a CBIR system acts practically as a black box. Its input are some query images and parameters, its output is a set of ranked result images. The user is mostly left in the dark *why* a result image has been chosen and how it is connected to the input images. Even worse, many non-expert users do not have the slightest idea what kind of features the computer is looking for. We will describe an experiment which shows that this situation can be helped by even very simple means.

4 Results of the ColoSeek approach

In [10], the query based CBIR system ColoSeek was introduced. We will go into the details of this system as far as necessary to motivate the Visual Analytics approach in

CBIR. ColoSeek resembles most other CBIR systems in the way that low-level color, texture and form features are extracted from images in a database and are saved in an index. For retrieval, the same features are extracted from query images and the best match features are searched for in the index to retrieve the best match images. There are two major differences between ColoSeek and other retrieval systems: Vision is focused from the very beginning of the processing, and there is a search space visualization.

4.1 Focused search

ColoSeek does not search for images, but for *regions*. For each image of the database, in the first processing stage Interest Points (IPs) are generated using standard IP-detectors, the Harris-detector [11] and a detector proposed in [12]. IPs denote regions of an image which are prominent or salient with respect to a certain feature (here: corners and symmetry). The subsequent extraction of color, texture and form features takes place within the regions selected by the detectors plus some features extracted from the overall appearance of the image. It is therefore only these selected regions a user can search for (or alternatively the overall structure of the image), while other regions are not available.

Correspondingly, the user input is not a set of images, but a set of regions he or she selects manually from query images. The system then tries to find an image which has a possibly big number of matching regions. If there is an image with a red car in front of a beach, the user should crop a red car from an arbitrary scene. In addition, he or she can specify the overall appearance of a beach (sand, horizon, blue sky) from another query image. This procedure solves at least part of the specification problem, it does not solve, however, the task to specify a red car when we have only a blue version.

The reasoning behind focused search is not only rooted in the specification problem but also in the abilities of pattern matching. IPs denote regions which are more suitable for the extraction of stable and distinctive features than other regions. This supports a clear ranking of similar / non-similar regions. The price, of course, is that the system may “overlook” parts of an image. So the idea can be summarized as “realize the possible instead of trying the impossible”. Further, focused search facilitates system feedback as described in the next section.

4.2 System feedback

To avoid the CBIR system acting as an unpredictable black box, the system must give feedback to the user. Basically, system feedback can be given at two points in the search: When images have been retrieved from the database, or already in the preceding stage when the user specifies the query. ColoSeek gives both types of feedback. For the result images, it is straight forward to show how or why they were selected by indicating the IP-centered regions which were best match to the query. Thus, the user immediately knows which components are the actual answer to the query and which are irrelevant.

The second type of feedback is less simple and will be the subject of the proposed Visual Analytics approach. In ColoSeek, the features the user has specified by input regions are visualized. As the feature extraction itself is somewhat complex, we will outline only the main idea in the present paper and refer to [10] for details. The feature extraction

employs spatio-chromatic filters which combine color, texture and form in a single feature. A spatio-chromatic filter can be visualized as a colored patch, i.e. a local distribution of different colors with defined spatial relations. For the motivation of spatio-chromatic features see also [13].

When the user specifies a query by several regions cut from sample images, the feature extraction boils them down to a representation by spatio-chromatic features. As a consequence, only certain aspects of the local color distribution are captured while others are lost. ColoSeek visualizes the captured color distribution as a weighted superposition of the spatio-chromatic filters. This way of visualization is, plainly, not a very good idea for several reasons. Firstly, a lot of information is lost in the superposition, a single image is not enough to visualize the search space. Secondly, the resulting image is blurred and tells little about the color patterns the system is looking for. So does this kind of system feedback make sense at all?

Surprisingly, the system feedback of ColoSeek proved to be extremely useful in a user study, in spite of its limited quality. When users were asked to search for objects in the image database, search times and results improved considerably when the system feedback was used. The effect was twofold:

1. Inexperienced users were surprised about what the CBIR system actually did. While in the beginning they assumed the system was searching for *objects* (in a high level sense), they realized it could do no more than looking for color patterns. This came as a disillusionment but helped them select fitting regions.
2. The displayed color patterns were sufficient to guide user: For example, to solve the task “find a tomato” the user realized they had to pick red, round objects (even if they were not tomatoes!) to maximize hits in the output.

Though the ColoSeek system is relatively simple and its visualization has drawbacks, the results motivate the Visual Analytics approach to CBIR outlined in the following.

5 Visual Analytics for CBIR

The emerging field of Visual Analytics aims at solving highly complex problems of pattern recognition and data mining by combining the benefits of computer visualization powers with the visual analytical skills of humans. Computational pattern recognition has limited ability but immense throughput, whereas humans are extremely skilled in analyzing patterns but have rather limited capacity. So, by a “synergy” of both, tasks of unknown size and complexity can be tackled. Visual Analytics is not so much an attempt to build more intelligent machines but machines that exploit better the intelligence of humans.

While Visual Analytics was in its beginnings primarily seen in the context of surveillance (in particular in the US) [14], the scientific community takes a much broader view today [15]. Visual Analytics has applications ranging from image browsing [16] over the analysis of social networks [17] to systems for the pharmaceutical industry [18] to name but a few. The success of Visual Analytics roots in the approach to bridge the “semantic gap” between a signal and its high level interpretation not by a machine, but by the user

supported by the machine. Due to its interactive nature, visual analytics is one of the rare technical fields where extensive user studies are conducted (e.g. [19]).

CBIR can profit from Visual Analytics mainly in two aspects: Search space visualization with interactive query improvement, and visualization of the way results were achieved. Both will be described in the following.

5.1 Search space visualization

Feature extraction of CBIR systems maps regions or even entire images onto a vector of real numbers (feature vector). Usually, this mapping can not be inverted, i.e. it is impossible to reconstruct the image out of the features. The idea is to capture the relevant information while filtering out irrelevant information. It is therefore not a straight forward task to visualize what a particular feature vector describes since it is a good match to many different patterns. There are basically two ways of visualization: Using natural or artificial patterns.

5.2 Visualization using natural images

A single feature or a vector of features can be visualized by a set of image patches cropped from natural images which are similar with respect to the chosen similarity measure (e.g. the euclidean distance). For a vector consisting of several features, this is essentially what common CBIR systems do. However, if we are not after obtaining a search result at once but to optimize features first, it is more suitable to visualize features in isolation. Let us consider a color histogram as an example, since color histograms are not only one of the first [20] and most intensively researched features in CBIR (e.g. [21, 22]), but remain until today one of its most frequently used features, in particular in sophisticated combinations with color texture [23, 24, 25]. As histograms discard all spatial information, there is a wide variety of image patches with similar histograms that can be found in a database. Which should we choose as a subset for visualization of the histogram? To visualize the variety of images covered by the histogram, a good choice will be a subset with a large variance with respect to one or more of the other features.

To be more precise, let $\vec{f}^Q = (f_1^Q \dots f_{N_f}^Q)^T \in \mathbb{R}^{N_f}$ denote a feature vector extracted from a query image Q , consisting of different components. For example, components $f_a^Q \dots f_b^Q$, $a < b$, might denote the bins of a color histogram. Let $\vec{f}^1 \dots \vec{f}^{N_D}$ be the feature vectors of the set of images $S = \{I_1 \dots I_{N_D}\}$ from the database which are similar to Q with respect to a suitable distance measure $\|\cdot\|$: $\|\vec{f}^Q - \vec{f}^i\| < \varepsilon$, $i = 1 \dots N_D$ for a similarity threshold ε . CBIR usually returns a set result images $R \subset S$ with smallest $\|\vec{f}^Q - \vec{f}^i\|$, $i = 1 \dots |R|$, as a result. But to visualize the effect of a subset of features (e.g. $f_a^Q \dots f_b^Q$), we display an image set V_{ab} instead, such that $\|\vec{f}^Q - \vec{f}^i\|_{ab}$, $i = 1 \dots |V_{ab}|$, is minimized. The modified distance measure $\|\cdot\|_{ab}$ evaluates only components $f_a \dots f_b$ of the feature vectors and disregards all others.

Since the visualization should be compact and give an idea of the search space given by the features $f_a^Q \dots f_b^Q$, we select a subset $V_{ab}^* \subset V_{ab}$ such that pairs \vec{f}^j, \vec{f}^k , $j \neq k$, of its feature vectors maximize $\|\vec{f}^j - \vec{f}^k\|_{\overline{ab}}$, $i = 1 \dots N_{ab}$. Here $\|\cdot\|_{\overline{ab}}$ is a distance measure

which evaluates all components except for $f_a \dots f_b$. Thus, the user can compare different features.

There are many other ways in which features can be visualized using natural images. Natural images have the advantage that the user knows from his/her own experience the objects represented by the features. However, this connection might lead to false assumptions on the nature of the feature. Therefore alternatives to natural imagery are interesting.

5.3 Visualization using artificial images

If we do not use a set of images V_{ab} similar to Q considering features $f_a \dots f_b$, then artificial samples have to be synthesized. For example, if $f_a \dots f_b$ denote a color histogram, then the task is to “invent” the missing spatial structure for the display of the color distribution. Such structure does not necessarily resemble physical objects, though certain properties of natural imagery should be observed. For example, the spectral power of natural images falls with spatial frequency f according to a power law $1/f^p$ [26]. Selecting a spatial frequency distribution of this kind would therefore be a good starting point for artificial images. But apart from that, we still lack methods to visualize color histograms such that the user gets an intuition of what the histogram describes. Likewise, the visualization of other low level vision features is an open question of research. Examples are texture features (such as the co-occurrence matrix) or local pattern descriptors (e.g. wavelets [27] or Gaussian derivatives [28]).

To guide the design of feature visualization, user experiments will be essential. A straightforward experiment can test whether a visualization helps the user identify an image from, e.g., a color histogram: Given the visualization, the test person has to find the best match from a collection of images. However, this is only part of the real task — the user has not only to judge whether a feature he or she has designed characterizes the target image well. Rather, the user must get help from the system in the design of the feature. This leads to the aspect of user–system interaction.

6 User–System Interaction

There have been several attempts to improve user–system interaction in CBIR by advanced system feedback. The most common system feedback is *backpropagation* of the results, i.e. display of the parts of an image which have led to its selection. Components which are responsible for the image being chosen are highlighted, for an overview see [29] and references therein. However, backprojection yields only part of the information: Though the user sees the key components of an image, he or she can not know why, i.e., based on which feature, they were chosen. Consequently, a first step will be labeling the key image components with the respective features.

More generally speaking, the development of user–system interaction is, in our opinion, still stuck on the wrong side of the semantic gap (the side of semantics), since system feedback still aims primarily at pointing out semantically meaningful components of images that caused their selection. By contrast, we suggest to visualize not only the components but also the features, both in isolation (as described in the previous section) and in the

result images. For the latter, novel visualization methods have to be found. For example, colors represented by a histogram may be visualized in an image using increased saturation while other colors are suppressed. But while the visualization of a simple color histogram might not put up big difficulties, more complex features will require more innovative methods.

Once methods for the visualization of features *as a such* (section 5) and features *within* images have been developed, novel query specification methods become possible. We envisage composing a query in the way of working with a drawing program: The user can click and drop pre-defined components with his or her own choice of parameters, in addition, parts from natural images may be used. The idea to compose a search from different components with defined spatial relations has long since been realized (e.g. [30]), however, on the level of concepts, not the *feature level*. An example is the use of pre-defined Color Names [31, 32] which does indeed resemble a drawing program, but in effect appeals to the users high level understanding of colors. For example, *skin color* is mainly a high level concept and extremely difficult to find on the pixel level without context information about faces etc. Therefore, the components from which a user can build a query should be visualizations of features, not concepts. Thus the CBIR interface should allow the user to build an object or scene by defining the spatial relations between patches, where the patches might be color histograms, texture features or characteristic local frequencies, each represented by an appropriate visualization. In other words, the interface should be coupled directly to the feature level, without any assumed “jump” over the semantic gap.

7 Conclusion

In this paper we have discussed how CBIR might profit from a visualization of the low level features it builds on. The underlying assumption of our approach is that closing the semantic gap by “intelligence” of machines is not realistic to date. Rather, machines should put humans in a position to design queries on a lower level. Let us assume we were successful in visualizing low level features, could we profit from this development apart from CBIR? In fact, CBIR is just one among many disciplines of Computer Vision facing the semantic gap. For practically every recognition task, the system designer can do no more than apply features known from experience, test the correlation of features and high level concepts, and hope for these features (a) to hold the correlation against unforeseen disruption of the object while (b) not to correlate with unknown objects. To improve this situation, feature visualization might find its way in every day Computer Vision engineering.

Literatur

- [1] Faloutsos, C.; Barber, R.; Flickner, M.; Hafner, J.; Niblack, W.; Petkovic, D.; Equitz, W.: Efficient and Effective Querying by Image Content. *J. Intell. Inf. Sys.* 3 (1994) 3, S. 231–262.
- [2] Smith, J. R.; Chang, S.-F.: VisualSEEK: A Fully Automated Content-Based Image Query System. In: *Proc. ACM Multimedia 96*, S. 87–98. 1996.

- [3] Pentland, A.; Picard, R. W.; Sclaroff, S.: Photobook: Content-Based Manipulation for Image Databases. *Int'l J. of Computer Vision* 18 (1996), S. 233–254.
- [4] Gupta, A.; Jain, R.: Visual Information Retrieval. *Commun. ACM* 40 (1997) 5, S. 70–79.
- [5] Carson, C.; Thomas, M.; Belongie, S.; Hellerstein, J. M.; Malik, J.: Blobworld: A System for Region-Based Image Indexing and Retrieval. In: *Proc. Visual Information Systems*, S. 509–516. 1999.
- [6] Gevers, T.; Smeulders, A. W. M.: PicToSeek: Combining Color and Shape Invariant Features for Image Retrieval. *IEEE Trans. on Image Processing* 9 (2000) 1, S. 102–119.
- [7] Wang, J. Z.; Li, J.; Wiederhold, G.: SIMPLicity: Semantics-Sensitive Integrated Matching for Picture Libraries. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 23 (2001) 9, S. 947–963.
- [8] Pope, A.; Lowe, D.: Probabilistic Models of Appearance for 3-D Object Recognition. *Int'l J. of Computer Vision* 40 (2000) 2, S. 149–167.
- [9] Borotschnig, H.; Paletta, L.; Prantl, M.; Pinz, A.: Appearance-based active object recognition. *Image and Vision Computing* 18 (2000), S. 715–727.
- [10] Heidemann, G.: Combining spatial and colour information for content based image retrieval. *Computer Vision and Image Understanding, Special Issue on Colour for Image Indexing and Retrieval* 94 (2004) 1-3, S. 234–270.
- [11] Harris, C.; Stephens, M.: A Combined Corner and Edge Detector. In: *Proc. 4th Alvey Vision Conf.*, S. 147–151. 1988.
- [12] Reissfeld, D.; Wolfson, H.; Yeshurun, Y.: Context-Free Attentional Operators: The Generalized Symmetry Transform. *Int'l J. of Computer Vision* 14 (1995), S. 119–130.
- [13] Heidemann, G.: The Principal Components of Natural Images Revisited. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 28 (2006) 5, S. 822–826.
- [14] Thomas, J.; Cook, K.: A visual analytics agenda. *IEEE Computer Graphics and Applications* 26 (2006) 1, S. 10–13.
- [15] Keim, D.; Robertson, G.; Thomas, J.; van Wijk, J.: Guest Editorial: Special Section on Visual Analytics. *IEEE Trans. on Visualization and Computer Graphics* 12 (2006) 6, S. 1361–1362.
- [16] Yang, J.; Fan, J.; Hubball, D.; Gao, Y.; Luo, H.; Ribarsky, W.; Ward, M.: Semantic Image Browser: Bridging Information Visualization with Automated Intelligent Image Analysis. In: *IEEE Symp. on Visual Analytics Science and Technology*. 2006.
- [17] Shen, Z.; Ma, K.-L.; Eliassi-Rad, T.: Visual Analysis of Large Heterogeneous Social Networks by Semantic and Structural Abstraction. *IEEE Trans. on Visualization and Computer Graphics* 12 (2006) 6, S. 1427–1439.

- [18] Saffer, J.; Burnett, V.; Chen, G.; van der Spek, P.: Visual analytics in the pharmaceutical industry. *IEEE Computer Graphics and Applications* 24 (2004) 5, S. 10–15.
- [19] Saraiya, P.; North, C.; Lam, V.; Duca, K.: An Insight-Based Longitudinal Study of Visual Analytics. *IEEE Trans. on Visualization and Computer Graphics* 12 (2006) 6, S. 1511–1522.
- [20] Swain, M.; Ballard, D.: Color Indexing. *Int'l J. of Computer Vision* 7 (1991) 1, S. 11–32.
- [21] Stricker, M. A.; Orengo, M.: Similarity of Color Images. In: *Proc. SPIE Storage and Retrieval for Image and Video Databases*, S. 381–392. 1995.
- [22] Funt, B. V.; Finlayson, G. D.: Color Constant Color Indexing. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 17 (1995) 5, S. 522–529.
- [23] Manjunath, B. S.; Ohm, J.-R.; Vasudevan, V. V.; Yamada, A.: Color and Texture Descriptors. *IEEE Trans. on Circuits and Systems for Video Technology* 11 (2001) 6, S. 703–715.
- [24] Mojsilovic, A.; Kovacevic, J.; Hu, J.; Safranek, R. J.; Ganapathy, S. K.: Matching and Retrieval Based on the Vocabulary and Grammar of Color Patterns. *IEEE Trans. on Image Processing* 9 (2000) 1, S. 38–54.
- [25] Qiu, G.; Feng, X.; Fang, J.: Compressing histogram representations for automatic colour photo categorization. *Pattern Recognition* 37 (2004), S. 2177–2193.
- [26] Tolhurst, D. J.; Tadmor, Y.; Chao, T.: Amplitude spectra of natural images. *Ophthalmic & Physiological Optics* 12 (1992) 2, S. 229–232.
- [27] Tian, Q.; Sebe, N.; Lew, M. S.; Louprias, E.; Huang, T. S.: Image Retrieval Using Wavelet-Based Salient Points. *J. of Electronic Imaging* 10 (2001) 4, S. 835–849.
- [28] Schmid, C.; Mohr, R.: Local Grayvalue Invariants for Image Retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 19 (1997) 5, S. 530–535.
- [29] Smeulders, A. W. M.; Worring, M.; Santini, S.; Gupta, A.; Jain, R.: Content-Based Image Retrieval at the End of the Early Years. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 22 (2000) 12, S. 1349–1380.
- [30] Smith, J. R.; Chang, S.-F.: Integrated Spatial and Feature Image Query. *Multimedia Systems* 7 (1999) 2, S. 129–140.
- [31] Ogle, V. E.: CHABOT – Retrieval from a Relational Database of Images. *Computer* 28 (1995) 9, S. 40–48.
- [32] Liu, Y.; Zhang, D.; Lu, G.; Ma, W.-Y.: Region-Based Image Retrieval with High-Level Semantic Color Names. In: *Proc. 11th Int'l Multimedia Modelling Conference MMM'05*, S. 180–187. Washington, DC, USA: IEEE Computer Society. 2005.

A Condensed Representation for Distributions over Set-Valued Attributes

Frank Rügheimer

Department of Knowledge Processing and Language Engineering
Otto-von-Guericke-University of Magdeburg
Germany
E-Mail: ruegheim@iws.cs.uni-magdeburg.de

Abstract

Set-valued attributes are routinely applied in knowledge modeling to represent imprecision. Other uses of set-valued data are encountered in Information Retrieval. A text, for instance, may cover topics from more than just one domain, even though we are able to clearly distinguish between the domains themselves. Thus, assigning a single “dominant” domain, would result in information loss and adversely affect recall rates.

In this work I present a condensed representation for multi-dimensional distributions over set-valued data. The representation allows access to both the probability of singleton values and one-point-coverages for each element in the attribute’s base domain. After presenting the underlying semantic framework, the approach is formalized and extended to the multi-dimensional case. It is shown how marginal distributions can be computed. Because the supplied interpretation of marginals is in line with the probabilistic approach, both calculi can be combined in applications. Finally a brief prospect to conditioning operations explores an extension of the field of application from the representation of static distributions to dynamic scenarios where distributions are changed in the light of new observations.

The suggested approach is distinguished from other information-compressed representations such as possibility distributions as equivalents of mass distributions with consonant focal sets [1] and the the context model [2], in that it supplies an intuitive probability-based aggregation operator without imposing restrictions on the admissible set-valued outcomes.

1 Introduction

The analysis of collected data and subsequent modeling constitute essential steps to solving many non-trivial classification, prediction or planning problems. Probabilistic models have shown to provide a powerful tool for capturing attribute interaction in such tasks by using approximates of their joined distribution. However, to successfully apply such approaches, a sample space of disjoint elementary events, must be identified.

The concept of a set, on the other hand, allows to naturally describe many phenomena in the real world. Additionally, sets provide a way of expressing imprecise knowledge with respect to an attribute value, by listing candidates for the unknown value.

Unfortunately set-valued data are not easily integrated into the probabilistic framework. Because values in the basic domain of a set-valued attribute may be expressed concurrently, the basic domain cannot serve as a sample space. For finite domains, a straightforward approach to that problem consists in choosing the power set of the attribute domain

as the sample space forming a Random Set [4]. Alas, the resulting sample spaces are very large. For example, for a basic domain with 30 elements the cardinality of the power set is already 2^{30} ($> 10^9$). Considering, that probabilistic models are build using conditional or joint distributions involving several of such attributes, the resulting distributions soon become unmanageable. Moreover, even if the representation-related issues could be overcome, the acquisition of sufficient data to reliably estimate the distributions would be infeasible.

To avoid these problems in applications, one may introduce assumptions that lead to more compact representations. In the last decades a large number of such representations have been suggested in the literature. Depending on underlying assumptions and interpretational framework these approaches allow probabilistic processing of set-valued data in specific, but always clear-cut scenarios.

Interpreting set-valuedness as imprecision, Dempster investigated upper and lower probabilities induced by set-valued mappings from a probability space [5]. In such a scenario the distribution is given a-priori and the set-valued outcomes (images under the mapping) are often called focal sets. For the multivariate case Dempster relied on independence assumptions and the heuristic known today as Dempster's Rule.

In his theory of evidence [6] Shafer extended Dempsters ideas employing mass distributions to quantify evidence. Later suggestions include operators and principles, which rely on a non-statistical, strictly subjectivist interpretation of probability assignments and are subsumed under belief function theory [7, 8, 9].

For the statistic-oriented approaches the problem of distribution size lead to information compressed representations. Possibility distributions, for instance, can be interpreted as a representation of a random set with consonant, i.e. nested, focal sets [1]. Due to consonance, information about both the extent of each focal set and its respective probability is fully preserved, so the original random set could still be reconstructed. The consonance requirement is relaxed in the context model interpretation of possibility [2], though at the expense of reduced representation power and an undefined aggregation operator [3]. Notably, different random sets correspond to the same representation in the context model. This information loss impedes a semantically sound definition of interpretable marginals as disjoint events can no longer be clearly differentiated. The context model approach also suggest one-point-coverages as information summaries. This work proposes a condensed representation for managing the interaction within groups of set-valued attributes.

In the following section I provide definitions for set-valued attributes and distributions of set-valued outcomes. It also addresses the problems of exponentially growing distribution size and low support for individual set-valued outcomes when estimating the respective distributions from data. In section 3 I suggest a condensed representation for set-valued attributes that avoids those difficulties by grouping multi-valued outcomes. The representation is first introduced for the one-dimensional case. That approach is extended to multi-dimensional domains in section 4, which includes detail on the computation of marginal and suggestions for conditioning operators.

2 Set-Valued Attributes

As this paper deals with several classes of attributes some notational conventions are introduced. Attributes are designated with capital letters from the beginning of the alphabet,

whereas letters from the second half are used for sets. Unless stated otherwise, I use superscripts to indicate the class of attributes and associated distributions or special symbols, and subscripts for indices. When applied to symbols that denote a distribution, subscripts usually list the set of attributes a distribution is defined over.

In the following an attribute A is seen as a random variable over a sample space $\text{dom}(A)$. A probability distribution p over $\text{dom}(A)$ is based on a probability measure P and defined as $p : \text{dom}(A) \rightarrow [0, 1]$, $p(a) \stackrel{\text{def}}{=} P(A = a)$. The distribution p suffices to recover the complete probability measure P on any σ -algebra over $\text{dom}(A)$.

As an extension of that definition, consider a set-valued attribute A^* over the sample space $\text{dom}(A^*) = 2^{\text{dom}(A)} \setminus \emptyset^1$. As before, a probability distribution p^* over $\text{dom}(A^*)$, $p^* : \text{dom}(A^*) \rightarrow [0, 1]$ is defined by $p^*(S) \stackrel{\text{def}}{=} P(A^* = S)$, but because $\text{dom}(A^*) = 2^{\text{dom}(A)} \setminus \emptyset$, the potential outcomes are now sets. Because the empty set is not among the values of A^* , p^* is a mass distribution².

Whereas direct reasoning with mass distributions has previously been proposed for small attribute domains [10], the growth in distribution size renders that approach inapplicable for larger domains. The difficulties with mass distributions stem from having to store probabilities for a large number of set-valued outcomes. In particular, even if all values can be represented in memory, providing reliable estimations of these probabilities would require unrealistically large samples. However, in application scenarios, one is rarely interested in detailed probability values for each of the set-valued outcomes. Instead, probabilities of singletons and the probability of coverage of elements by set-valued events (one-point-coverages) are useful as information summaries. By focussing on these pieces of information a compact representation is achieved.

3 Condensed Representation

To provide an information compressed representation of a random set two principles are employed. The first approach relies on coarsening of the sample space, i.e. giving up the distinction between set-valued outcomes within certain subgroups. The induced probability distribution on the resulting coarser frame is less informative, but can be more easily estimated. The second principle consists in switching to an element-oriented view. The idea behind this representation is that one may not be interested in the focal sets themselves, but rather in the probability that a given element from the basic domain is an element of the random set (element coverage). The resulting coverage probabilities are not associated with disjoint events and thus cannot be aggregated in a meaningful way.

For the proposed condensed representation both approaches are employed. To reduce distribution size, all multi-valued outcomes are merged in a coarsening step. In addition to that, the proportion of coverage by the original multi-valued outcomes is recorded for every element of the basic domain. Thus although coarsening removes information about individual multi-valued outcomes, some of that information is preserved in the form of summarized element coverage. In this work the presentation focuses on discrete distribution on finite domains, though the ideas can be transferred to the continuous case.

¹The empty set is usually not considered a valid value of a set-valued attribute.

²In the framework of belief theory [6] this concept is known as a basic probability assignment

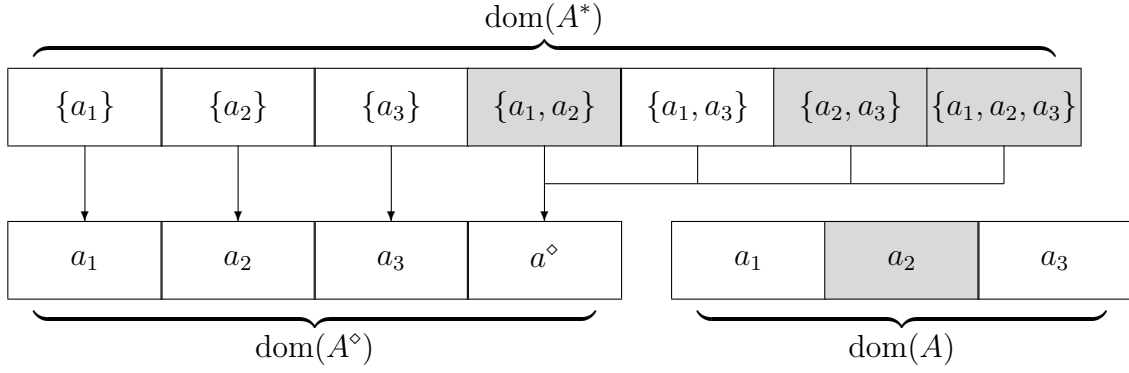


Figure 1: Domains of a set-valued attribute A^* , the induced condensed set-valued attribute A^\diamond and underlying base attribute A . Arrows indicate contributions of probability mass. Shaded elements of $\text{dom}(A^*)$ mark multi-valued outcomes covering a_2 .

Definition 1: Let A be a conventional attribute with values from $\text{dom}(A)$ and A^* the a set-valued extension of A . We can define a condensed set-valued attribute A^\diamond over the sample space $\text{dom}(A^\diamond) = \text{dom}(A) \cup \{a^\diamond\}$, where $a^\diamond \notin \text{dom}(A)$ by relating outcomes ω_{A^\diamond} from $\text{dom}(A^\diamond)$ to outcomes ω_{A^*} from $\text{dom}(A^*)$ as follows:

$$\omega_{A^\diamond} \stackrel{\text{def}}{=} \begin{cases} a, & \text{if } \omega_{A^*} = \{a\}, \\ a^\diamond & \text{if } |\omega_{A^*}| > 1. \end{cases} \quad (1)$$

The relation between the attribute domains is illustrated in figure 1. As a consequence of the equation 1, a probability distribution p^* over $\text{dom}(A^*)$ induces a coarsened probability distribution p^\diamond over $\text{dom}(A^\diamond)$:

$$p^\diamond(a) = \begin{cases} p^*(\{a\}) & \text{if } a \in \text{bdom}(A^\diamond), \\ \sum_{S \in \text{dom}(A^*), |S| > 1} p^*(S) & \text{if } a = a^\diamond. \end{cases} \quad (2)$$

A is called the *base attribute* and the underlying domain $\text{bdom}(A^\diamond) \stackrel{\text{def}}{=} \text{dom}(A)$ is referred to as the *basic domain* of the condensed set-valued attribute A^\diamond . For $a \neq a^\diamond$, the value $p^\diamond(a)$ quantifies not a probability of occurrence, but rather the probability of observing a as the *sole* element in a set-valued outcome from $A^*(\omega)$. The originally multi-valued outcomes are mapped to the new attribute value a^\diamond . Since p^\diamond is a probability distribution, standard operations like conditioning and marginalization can still be employed with this representation. Moreover, equation 1 can be used to estimate the condensed probability distributions directly from data. A distribution with $p^\diamond(a^\diamond) = 0$ corresponds to the limiting case of a conventional attribute.

Because of $p^*(\{a\}) = p^\diamond(a)$, the probabilities of singleton outcomes are directly represented in distributions over condensed set-valued attributes. But to reconstruct one-point coverages the representation has to be further enriched.

Definition 2: Let A^\diamond be a condensed set-valued attribute based on the set-valued attribute A^* . Furthermore let p^* denote a distribution over A^* and p^\diamond the distribution obtained by applying equation 2. Consider now the coverage function $c^\diamond : \text{dom}(A) \rightarrow [0, 1]$,

$$c^\diamond(a) = \begin{cases} \frac{\sum_{S \in \text{dom}(A^*), a \in S, |S| > 1} p^*(S)}{p^\diamond(a^\diamond)} & \text{if } p^\diamond(a^\diamond) > 0, \\ 1 & \text{otherwise.} \end{cases} \quad (3)$$

The value $c^\diamond(a)$ denotes the conditional probability of a being *among* the elements of a multi-valued outcome. The one-point-coverage $\text{opc}(a)$ of a value $a \in \text{dom}(A)$ is computed as $\text{opc}(a) = \sum_{a \in S} p(S) = p^\diamond(a) + p^\diamond(a^\diamond) \cdot c^\diamond(a)$. Of course one could store the values for the one-point-coverage directly. But defining coverage factors relative to the proportion of multi-valued outcomes has the advantage, that their value does not have to be adapted when conditioning p^\diamond .

In the following, the term *condensed distribution* is understood as a tuple (p^\diamond, c^\diamond) that is formed by a condensed probability distribution and corresponding coverage function referring to a condensed set-valued attribute. As a convention I will use corresponding attribute names to relate conventional, sets-valued and condensed set-valued attributes that are defined with respect to the same base domain.

4 Reasoning with Condensed Set-Valued Attributes

So far only distributions over single attributes have been considered. Shifting attention to reasoning, one needs to examine the interaction between attributes. A direct way of representing attribute interaction is by means of joint distributions, which, for instance, can be estimated from contingency tables.

For of clarity I have introduced the condensed distribution for the one-dimensional case first, but working with multidimensional distributions, it is more comfortable to switch to a function-based formalization of sample spaces, so to avoid having to track indices. The formalization was previously used by Borgelt in [3], where it is explicated in detail. Tuples or instantiations over a set X of attributes are defined as mappings $t_X : X \rightarrow \bigcup_{A \in X} \text{dom}(A)$ with the constraint, that each attribute $A \in X$ may only be mapped to elements of its original domain $\text{dom}(A)$. A tuple t over $\{A, B, C\}$, which maps attribute A to a_1 , B to b_2 and C to c_2 is written $t = (A \mapsto a_1, B \mapsto b_2, C \mapsto c_2)$. This is shortened to $t = (a_1, b_2, c_2)$, if an implicit order is fixed. Tuples t_Y over subsets $Y \subseteq X$ of attributes can be seen as restrictions $t_{X|Y}$ of the functions t_X to Y . In particular, $t_{X|\emptyset}$ yields the empty tuple $()$. When speaking about sets S of tuples over an attribute set X , the projection of S to Y is defined as $\text{proj}_Y^X(S) = \{t_Y \in T_Y \mid \exists t_X \in S : t_Y \equiv t_{X|Y}\}$.

Probability distributions and coverage functions, referring an underlying set X of attributes are defined over the set T_X of all possible tuples or instantiations t_X over X . For a more elaborate introduction to the formalism the reader is referred to BorKru2002. Converting the notation used in section 3 to the above formalism, the sample space used for p^\diamond would be written as $T_{\{A^\diamond\}}$ and the sample space formed by the original set-valued outcomes is expressed as $2^{T_{\{A\}}} \setminus \emptyset$.

4.1 Joint Distributions

Turning to the multidimensional case, it is no longer sufficient to only distinguish between singletons and multi-valued outcomes. The reason is that potential outcomes that are multi-valued when considered in a high dimensional space may be represented by singletons when projected to a lower dimensional space. Consequently, to be able to compute marginal distributions, it is necessary to consider the cardinality of an outcomes projections when selecting the coarsening.

Given a set $X^\diamond = A_1^\diamond, \dots, A_n^\diamond$ of condensed set-valued attributes this distinction is comfortably achieved by selecting T_{X^\diamond} as the sample space. The equivalent of the full sample space for set-valued outcomes is formed by the non-empty subsets of the combined base domains, i.e. $2^{T_X} \setminus \emptyset$.

The mapping σ_X serves to map individual set-valued outcomes to tuples in the coarsened sample space:

Definition 3: Consider a set $X = \{A_1, \dots, A_n\}$ of attributes and the set T_X of tuples over X . Let X^\diamond be an associated set of condensed set-valued attributes $X^\diamond = \{A_1^\diamond \dots A_n^\diamond\}$, such that $\text{dom}(A_i) = \text{bdom}(A_i^\diamond)$. The mapping $\sigma_X : 2^{T_X} \setminus \emptyset \rightarrow T_{X^\diamond}$ is defined by

$$\sigma_X(S)(A^\diamond) \stackrel{\text{def}}{=} \begin{cases} t_{\{A\}}^X(A), & \text{where } \{t_{\{A\}}^X\} = \text{proj}_{\{A\}}^X(S), \text{ if } |\text{proj}_{\{A\}}^X(S)| = 1, \\ a^\diamond & \text{if } |\text{proj}_{\{A\}}^X(S)| > 1. \end{cases} \quad (4)$$

If S is a singleton, so are all its projections. Conversely, if the cardinality of S is greater than one, the tuple $\sigma_X(S)$ must assign a the symbol for a multi-valued outcome to at least one of the attributes. Again the distribution p^* of set-valued outcomes, induces the probability distribution p^\diamond via the mapping σ_X .

Definition 4: Let $X^\diamond = \{A_1^\diamond, \dots, A_n^\diamond\}$ be a finite nonempty set of condensed set-valued attributes with respective domains $\text{dom}(A_1^\diamond \dots A_n^\diamond)$ and $X = \{A_1, \dots, A_n\}$ be a set of associated base attributes, such that $\text{dom}(A_i) = \text{bdom}(A_i^\diamond)$. A distribution p^* over $2^{T_X} \setminus \emptyset$ is summarized by a condensed probability distribution $p^\diamond : T_{X^\diamond} \rightarrow [0, 1]$ with

$$p^\diamond(t_{X^\diamond}) = \sum_{\substack{S \in 2^{T_X} \setminus \emptyset \\ \sigma_X(S) \equiv t_{X^\diamond}}} p^*(S) \quad (5)$$

For each tuple t_{X^\diamond} , the inverse image under σ_X identifies the set-valued outcomes that contribute to its probability mass in the condensed joint distribution. The definition assigns exactly those outcomes S to t_{X^\diamond} that for all attributes $A \in X$:

- *either* contain only tuples, which assign the same value to A as t_{X^\diamond} does to A^\diamond (the outcome is single-valued w.r.t. A)
- *or* have a projection to $\{A\}$ with cardinality greater than one i.e., the outcome is multi-valued w.r.t. A .

In other words, a tuple represents a class of set-valued outcomes, that are equal with respect to which of their projections result in singletons and the particular instantiation in those projections. Due to this distinction, the representation retains information required for marginalization operations to different subsets of attributes. The contribution to element coverage are subdivided in parallel to that. For each tuple from T_{X^\diamond} that represents a class of multi-valued outcomes over T_X the relative element coverage is recorded. For tuples that assign more than just one attribute the multivalued symbol additional lower-dimensional coverage functions are stored. The latter are required only to compute coverage factors in marginal distributions.

The relation of coverage values to a specific class of multi-valued outcomes is established via subscripts, leading to designations of the form $c_{X,[Y],t_{Z^\diamond}}^\diamond(t_Y)$. As before, the first subscript X denotes the attribute set under consideration. The second subscript (given in squared brackets) indicates a subset $Y \subseteq X$ of attributes that are distinguished when computing element coverage within the class. To fully identify the class the third subscript is a tuple over $Z^\diamond = X^\diamond \setminus Y^\diamond$.

Definition 5: Let $X^\diamond = \{A_1^\diamond, \dots, A_n^\diamond\}$ a finite nonempty set of condensed set-valued attributes and X the set of respective base attributes. A distribution p^* over a $2^{T_X} \setminus \emptyset$ is summarized by the condensed set-valued distribution p^\diamond and a family of functions $c_{X,[Y],t_{Z^\diamond}}^\diamond : T_Y \rightarrow [0, 1]$ such, that for all $Y \subseteq X$ and for all t_{Z^\diamond} with $Z^\diamond = X^\diamond \setminus Y^\diamond$:

$$c_{X,[Y],t_{Z^\diamond}}^\diamond(t_Y) \stackrel{\text{def}}{=} \begin{cases} \frac{\sum_{S \in \mathcal{S}} p^*(S)}{p_{X^\diamond}^\diamond(t_{X^\diamond})} & \text{if } p_{X^\diamond}^\diamond(t_{X^\diamond}) > 0 \\ 1 & \text{otherwise,} \end{cases} \quad (6)$$

where t_{X^\diamond} is the unique tuple from T_{X^\diamond} that satisfies $t_{X^\diamond|Z^\diamond} \equiv t_{Z^\diamond}$ and maps the remaining attributes Y^\diamond to the respective symbol for grouped multi-valued outcomes, whereas $\mathcal{S} = \{S_1 : S_1 \in 2^{T_X} \setminus \emptyset, \sigma_X(S_1) \equiv t_{X^\diamond}, t_Y \in \text{proj}_Y^X(S_1)\}$.

If $Y = \emptyset$, the third parameter is a tuple t_{X^\diamond} that for each attribute is instantiated with a value from the respective from base domain. The inverse image t_{X^\diamond} of under σ_X contains only the singleton corresponding $\{t_X\}$. The only valid function argument is the empty tuple $t_Y = ()$ such that the projection condition trivially holds. Together with equation 5 and $t_X \in \text{proj}_X^X(\{t_X\}) = \{t_X\}$ we obtain $\forall t_X \in T_X$:

$$\begin{aligned} c_{X,[\emptyset],t_{X^\diamond}}^\diamond(()) &= \begin{cases} \frac{\sum_{S \in 2^{T_X} \setminus \emptyset, \sigma_X(S) \equiv t_{X^\diamond}} p^*(S)}{p^\diamond(t_{X^\diamond})} & \text{if } p^\diamond(t_{X^\diamond}) > 0 \\ 1 & \text{otherwise} \end{cases} \\ &= \begin{cases} \frac{p^*(\{t_X\})}{p^*(\{t_X\})} & \text{if } p^\diamond(t_{X^\diamond}) > 0 \\ 1 & \text{otherwise} \end{cases} = 1. \end{aligned}$$

Of course, no explicit representation is required in that case.

Before turning to the computation of one-point-coverage, it is useful to introduce the notion of tuple compatibility. Given a set X^\diamond of condensed set-valued attribute and the set X of their respective base attributes, I call a tuple $(t_{X^\diamond}) \in T_{X^\diamond}$ compatible to a tuple $t_X \in T_X$ if $\forall A \in X$ either $t_{X^\diamond}(A^\diamond) = t_X(A)$ or $t_{X^\diamond}(A^\diamond)$ assumes the surrogate value for a multi-valued result. For traversing compatible tuple it is useful to define an auxiliary function

Definition 6: Let $X = \{A_1, \dots, A_n\}$ be a set of attributes. The function $\text{mv}_X : T_X \times 2^X \rightarrow T_{X^\diamond}$ is defined as

$$\text{mv}_X(t, Y)(A^\diamond) = \begin{cases} t(A), & \text{if } A \notin Y \\ a^\diamond, & \text{if } A \in Y. \end{cases} \quad (7)$$

where A^\diamond denotes the condensed set-valued attribute with base attribute A .

The second argument Y determines the attributes for which the resulting tuple returns the surrogate symbol for multi-valued instantiation.

The one-point-coverage of a tuple $t_X \in T_X$ is computed from the probability assigned to the compatible tuples (t_{X^\diamond}) from T_{X^\diamond} and weighting them with the respective coverage factor. The compatible tuples are traversed by using mv in on t_X with all selections from the attributes in X^\diamond .

$$\text{opc}(t_X) = \sum_{Y \subseteq X, Z = X \setminus Y} p^\diamond(\text{mv}_X(t_X, Y)) \cdot c_{X, [Y], \text{mv}_Z(t_X|Z, \emptyset)}^\diamond(t_X|Y). \quad (8)$$

An example of a two-dimensional condensed joint distribution is given in figure 2.

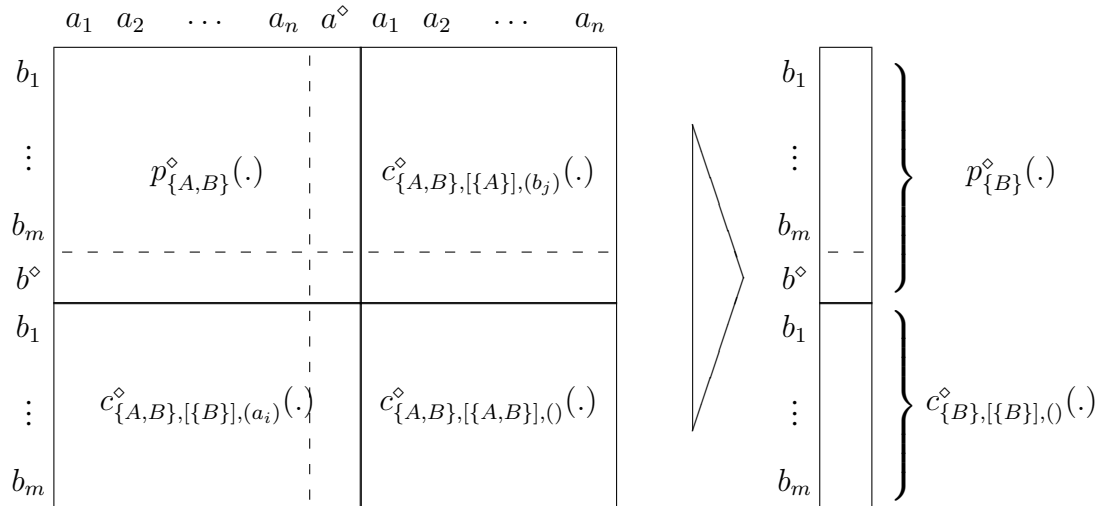


Figure 2: Joint distribution with coverage values for two condensed set-valued attributes and one of its marginal distributions.

4.2 Marginal Distributions

Whereas high dimensional distributions allow to encode complicated relations between attributes, one is usually interested in only a subset of these attribute at a given time. Marginal distributions allow to summarize higher dimensional distributions by only admitting distinctions w.r.t. a subset of the attributes.

For the probabilistic part of the condensed set-valued distribution over a set of attributes X^\diamond , the marginal distribution with respect to any subset of attributes $Y^\diamond \subseteq X^\diamond$ is computed by assigning the aggregated probability mass of higher dimensional tupels to their common projections in the lower dimensional space.

$$\begin{aligned} p_{Y^\diamond}^\diamond(t_{Y^\diamond}) &= P_{Y^\diamond}^\diamond(\{t_{Y^\diamond}\}) = P_{X^\diamond}^\diamond(\{t_{X^\diamond} \mid t_{X^\diamond}|_{Y^\diamond} \equiv t_{Y^\diamond}\}) \\ &= \sum_{t_{X^\diamond} \in T_{X^\diamond}, t_{X^\diamond}|_{Y^\diamond} \equiv t_{Y^\diamond}} p_{X^\diamond}^\diamond(t_{X^\diamond}). \end{aligned} \quad (9)$$

To provide meaningful coverage factors for the projection recall equations 1 and 2. As before, for each aggregated probability value associated values for relative coverage with respect to the projected space need to be aggregated. Because coverage values specify a proportion relative to the a tuple t_X^\diamond that represents a class of set-valued outcomes,

the correct way to aggregate their contribution to marginal coverage factors it to use a weighted average. Assuming $p^\diamond_{Y^\diamond}(\text{mv}_Y(t_Y, Z)) \neq 0$ this yields

$$c^\diamond_{Y, [Z], t_{Y^\diamond \setminus Z^\diamond}}(t_Z) \stackrel{\text{def}}{=} \frac{\sum_{\substack{t_{X^\diamond} \in T_{X^\diamond}, t_{X^\diamond | Z^\diamond} \equiv \text{mv}(t_Z, Z) \\ t_{X^\diamond | Y^\diamond \setminus Z^\diamond} \equiv t_{Y^\diamond \setminus Z^\diamond}}} p^\diamond_{X^\diamond}(t_{X^\diamond}) \cdot c^\diamond_{X, [Z], t_{X^\diamond | X^\diamond \setminus Z^\diamond}}(t_Z)}{p^\diamond_{Y^\diamond}(\text{mv}_Y(t_Y, Z))}. \quad (10)$$

For $p^\diamond_{Y^\diamond}(\text{mv}_Y(t_Y, Z)) = 0$ the value is defined as 1. Because the coverage value is weighted with the respective class probability, this arbitrary definition has no adverse effect, but it ensures that subsequent operations have defined inputs.

By construction the marginal of a condensed distribution coincides with a condensed distribution computed directly from projections of the original set-valued outcomes. This allows to relate observations that are made with respect to different but overlapping sets of attributes. Moreover, because the coarsened distribution p^\diamond is a regular probability distributions, the suggested representation is embedded in the probabilistic framework. Conventional probability distributions can be viewed as a limiting case of the condensed set-valued distribution. This allows to mix conventional and set-valued attributes in applications.

5 Future Work

Due to the representation of the coverage function, one-point coverages are implicitly updated when the distribution p^\diamond is altered. This property is useful for applications that use condensed set-valued or hybrid distributions to represent generic knowledge. By conditioning, such a model can later be enriched with information from case-specific observations to adapt the model to a situation at hand. But due to the coarsening step, information from observations w.r.t. multi-valued outcomes is only used indirectly, when the probability distribution is conditioned. In addition working with the coarsened distributions, modified conditioning schemes that better exploit the coverage information could be devised and tested for such applications.

Finally, even with a information-compressed representation, joint distributions used to represent attribute interaction may become very large if the number of attributes is increased. To counter that problem, the integration of condensed set-valued attributes with graphical models will be further explored.

References

- [1] Dubois, D.; Prade, H.: *Possibility Theory*. New York, New York: Plenum Press. Translation of: *Théorie des possibilités*. 1988.
- [2] Gebhardt, J.; Kruse, R.: The Context Model – An Integrating View of Vagueness and Uncertainty. *Int. Journal of Approximate Reasoning* (1993) 9, S. 283–314.
- [3] Borgelt, C.; Kruse, R.: *Graphical Models—Methods for Data Analysis and Mining*. Chichester: J. Wiley & Sons. 2002.

- [4] Nguyen, H. T.: On random sets and belief functions. *Journal Math. Anal. Appl.* 65 (1978), S. 531–542.
- [5] Dempster, A.: Upper and Lower Probabilities Induced by a Multivalued Mapping. *Ann. Math. Statist.* 38 (1967), S. 325–339.
- [6] Shafer, G.: *A Mathematical Theory of Evidence*. Princeton University Press. 1976.
- [7] Shafer, G.: *The analysis of fuzzy Information*, Bd. 1, Kap. Belief functions and possibility measures. Boca Raton, FL: CRC Press. 1986.
- [8] Smets, P.: Belief functions: the disjunctive rule of combination and the generalized bayesian theorem. *International Journal of Approximate Reasoning* 9 (1993), S. 1–35.
- [9] Smets, P.; Kennes, R.: The transferable belief model. *Artificial Intelligence* 66 (1994), S. 191–234.
- [10] Kruse, R.; Nauck, D.; Klawonn, F.: Reasoning with Mass Distributions. In: *Uncertainty in Artificial Intelligence* (D'Ambrosio, B. D.; Smets, P.; Bonissone, P. P., Hg.). San Mateo, California: Morgan Kaufmann. 1991.

Mit datenbasierten Technologien und Versuchsplanung zu erfolgreichen Produkten

Thomas Mrziglod

Bayer Technology Services GmbH

PT-AS-CS

51368 Leverkusen

Tel. (0214) 3027516

E-Mail: Thomas.Mrziglod@bayertechnology.com

Zusammenfassung

Zur systematischen Unterstützung eines datenbasierten Produktentwicklungs-Prozesses ist die so genannte Product Design Workbench entwickelt worden. Das integrierte Softwaresystem unterstützt dabei den typischen Arbeitsablauf bestehend aus Versuchsplanung, Datenanalyse, Modellerstellung und Rezepturoptimierung. Zur präzisen Modellierung des oft komplexen Rezeptur-Eigenschafts-Zusammenhangs wird dabei auf Künstliche Neuronale Netze oder Hybridmodelle zurückgegriffen.

1 Einleitung

Die zunehmende Komplexität der Prozesse sowie der Innovations- und Kostendruck stellt die industrielle Produktentwicklung vor ganz neue Herausforderungen. Die Integration von Datenerfassung, Analyse, Modellierung und Optimierung sowie maßgeschneiderter Versuchsplanungsmethoden sind die neuen Anforderungen des Marktes an die Prozessabläufe. Alle Integrationsbemühungen bleiben jedoch wirkungslos, wenn die Verfahren der wachsenden Komplexität der Produktentwicklung nicht gerecht werden.

Mit Produktentwicklung ist hierbei das Auffinden der richtigen Formulierung gemeint, so dass das resultierende Produkt ein vorgegebenes Eigenschaftsprofil möglichst gut erfüllt. Dabei sind sowohl die einzusetzenden Komponenten als auch deren Konzentration sowie optimale Werte für Produktionsparameter zu bestimmen. Ziel kann dabei die Entwicklung völlig neuer Rezepturen oder aber auch die Modifikation bestehender Rezepturen sein, bei denen beispielsweise teure Einsatzstoffe durch kostengünstigere ersetzt werden ohne die resultierenden Produkteigenschaften wesentlich zu verändern.

Produktentwicklung beinhaltet in der Regel eine Mehrzieloptimierung, wobei 10, 20 und mehr Zielgrößen keine Seltenheit sind. Um den daraus resultierenden Schwierigkeiten einer systematischen Produktentwicklung gerecht zu werden, hat Bayer Technology Services das System einer Product Design Workbench entwickelt, die alle wesentlichen Komponenten für eine datenbasierte Produktentwicklung integriert, und in zahlreichen Projekten innerhalb und außerhalb des Bayerkonzerns eingesetzt. Industrielle Anwendungen mit großem Potential liegen im Bereich der Rezepturoptimierung z.B. für die Polymer-, Waschmittel-, Kosmetik- und Nahrungsmittelindustrie.

2 Das Konzept

Bei der Product Design Workbench handelt es sich um ein integriertes Softwaresystem zur Unterstützung der Arbeitsschritte *Versuchsplanung*, *Datenanalyse*, *Modellerstellung* und *Rezepturoptimierung*. Dabei können sowohl die Auswahl der bereitgestellten Module als auch deren Funktionalitäten für die speziellen Erfordernisse des jeweiligen Einsatzbereichs maßgeschneidert werden.

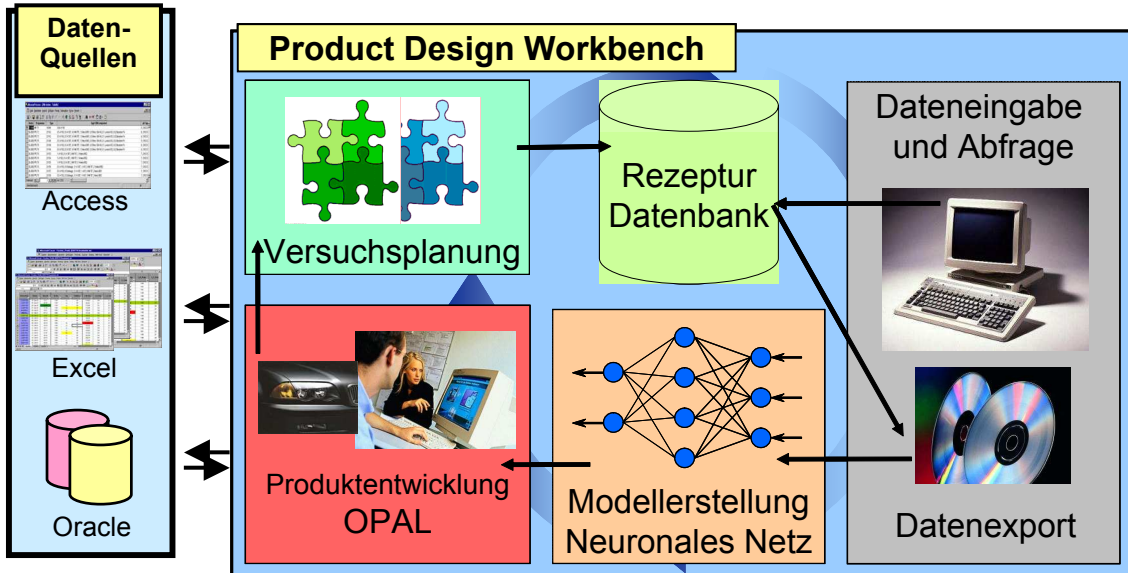


Abbildung 1

Der typische Arbeitsablauf sieht folgendermaßen aus (Abbildung 1): Alle in einem Bereich anfallenden Rezepturen und die vermessenen Eigenschaften der entstehenden Mischungen werden in einer internen Datenbank abgelegt. Für eine konkrete Fragestellung können die dafür relevanten Datensätze mit Hilfe einer Abfragekomponente aus der Datenbank extrahiert werden. Für diesen keineswegs trivialen Arbeitsschritt ermöglicht das System die Berücksichtigung von Expertenwissen.

Die Modellerstellung selbst läuft durch Anwendung von Künstlichen Neuronalen Netzen (Multilayer Perceptrons) oder Hybridmodellen. Dabei kombiniert ein Hybridmodell Neuronale Netz-Modelle mit physikalisch-chemischen Modellen. Solche datenbasierten Modellierungsmethoden sind sehr gut geeignet, um systematisch versteckte Informationen in vorhandenen Messdaten in Prognosemodelle zu extrahieren. Dabei sind je nach Fragestellung Modelle unterschiedlicher Komplexität erforderlich, um ans Ziel zu gelangen. Gerade bei der Produktentwicklung ist es wichtig, mit Modellen zu arbeiten, die auf den jeweils zu einer speziellen Fragestellung passenden Datensätzen basieren. Mit Hilfe der Modell-Anwendungskomponente OPAL (Optimierungs-Prognose-Analyse-Labor) wird das Modell zur Prognose von Eigenschaftsprofilen für neue virtuelle Rezepturen und zur Berechnung neuer Rezepturen nach vorgegebenen Profilen herangezogen. Für den Fall, dass die dem Modell zugrundeliegende Datenbasis noch Lücken aufweist, können diese mit Hilfe der Versuchsplanungs-Komponente systematisch geschlossen werden. Dabei werden die Versuchsvorschläge nicht nur einfach breit im Rezeptorraum verteilt, sondern können auch nach Maßgabe eines bereits vorliegenden Modells verstärkt in Bereiche mit hoher zu erwartender Produktqualität gelegt werden.

Nach Vermessung der Eigenschaften der neuen Rezepturen werden die Qualitäts- und Rezeptur-Daten wiederum in die Datenbank importiert, um sie für eine Modellaktualisierung heranzuziehen. Der gesamte Vorgang wird nun solange fortgesetzt, bis eine hinreichend gute Rezeptur gefunden ist.

3 Die Module

Zentrales Element des Systems ist die interne Datenbank. Hier wurde ein allgemeines Datenmodell gewählt, um Rezepturdaten aus den verschiedensten Bereichen erfassen zu können. Dies ermöglicht auch den Einsatz als zentrales Datenhaltungssystem für einen kompletten Einsatzbereich. Grundsätzlich ist der Datenbanktyp flexibel gehalten; Standard ist je nach Anforderung Oracle oder Access. Für den Datenimport und -export sind Standardkomponenten vorhanden. Üblicherweise wird auch eine Anbindung an vorhandene Datenquellen vorgenommen.

3.1 Relevante Rezepturen

The screenshot displays a software interface for database queries. The top section, titled 'Database Query', includes a toolbar, a search criteria field (set to '1'), and a status bar indicating 'Query has 611 hits (after 0 iterations)'. Below this is a tabbed interface with 'Object Names' selected, showing a tree view of the database structure. The right side of this section shows a table with columns 'Class', 'Name', and 'Name', containing one entry: '1 Prozess Compounding Matrix Condition_1'. Below the table are 'Logical Operations' and a field containing 'Condition_1'. The bottom section, 'Query Results', features a tree view on the left and a table with columns: 'Export', 'Name', 'Class', 'Distribution', 'Count', 'Min', 'Mean', 'Max', and 'STD'. The table lists four quality attributes: 'Biegefestigkeit', 'Schlagzähigkeit', 'Zugfestigkeit', and 'Hydrolysebeständig'. A 'Distribution Visualization' plot is shown below the table, displaying a histogram of values for 'Biegefestigkeit'.

Export	Name	Class	Distribution	Count	Min	Mean	Max	STD
<input checked="" type="checkbox"/>	Biegefestigkeit	Qualität	Plot	610	202,30	281,55	336,35	14,83
<input checked="" type="checkbox"/>	Schlagzähigkeit	Qualität						6,16
<input checked="" type="checkbox"/>	Zugfestigkeit	Qualität						10,13
<input checked="" type="checkbox"/>	Hydrolysebeständig	Qualität						19,47

Abbildung 2

Die Abfragekomponente ermöglicht es dem Produktentwickler die passenden Rezepturen in der Datenbank zu finden. Typische Vorgaben sind: Sollen bestimmte Komponenten enthalten sein oder nicht, in wie viel Datensätzen soll eine Einstellgrößen mindestens vorhanden sein, damit sie in die Betrachtung einbezogen wird. Gegebenenfalls können Parameter, Komponenten oder allgemein Einflussgrößen gruppiert werden. Genauso kann aber auch die Einhaltung von Spezifikationsbereichen in den Datensätzen eine Rolle spielen. Abfragen werden mit Hilfe einer grafischen Benutzeroberfläche zusammengestellt, im Hintergrund wird automatisch ein SQL-

Statement generiert. Als Ergebnis werden zunächst statistische Informationen über den sich ergebenden Datensatz ausgegeben: Anzahl der Rezepturen, Liste der enthaltenen Komponenten, Konzentrationsbereiche, gemessene Eigenschaften, Histogramme (Abbildung 2). Auf Basis dieser Informationen wird die Abfrage nun so lange verfeinert, bis sich ein brauchbares Verhältnis von Problemkomplexität (entspricht Anzahl der Einflussgrößen) zu Anzahl der Datensätze ergibt.

3.2 Modellerstellung

Der resultierende Datensatz wird per Knopfdruck als flache Tabelle exportiert und an die Modellerstellungskomponente weitergereicht. Je nach Anforderung des jeweiligen Einsatzbereichs wird mit Neuronalen Netzen oder Hybridmodellen gearbeitet. In vielen Fällen sind Neuronale Netze ein hinreichend leistungsstarkes Instrument, um die Rezeptur-Eigenschafts-Beziehung mit guter Genauigkeit abzubilden. Für diesen Fall wurde eine kommerziell verfügbare Neuronale Netz Komponente zur Unterstützung des Produktentwicklers integriert. Diese ist soweit vorparametrisiert, dass in der Regel auch von einem Nicht-Experten automatisiert ein Neuronales Netzmodell erstellt werden kann.

In komplexeren Situationen kann das Modellierungsverhalten Neuronaler Netze unter Einbeziehung von strukturellem Vorwissen über den zu Grunde liegenden Zusammenhang nochmals deutlich verbessert werden. Zentrale Idee dabei ist, dass die Dimension des Lösungsraums durch Einbeziehung von Vorwissen reduziert wird. So ein Strukturiertes Hybridmodell besteht daher aus drei Komponenten:

- Rigorosen Teilmodellen zur Beschreibung des Eingangs-Ausgangs-Verhaltens für die Teilprozesse, für die entsprechendes Prozessverständnis verfügbar ist,
- Neuronalen Netz Teilmodellen für die Teilprozesse, für die nur unzureichendes Prozessverständnis verfügbar ist,
- sowie dem Verfahrensfließbild zur Verschaltung der Teilmodelle.

Dieser Ansatz ist seit 1995 sowohl theoretisch, als auch praktisch überprüft worden [3]. Dabei konnte gezeigt werden, dass sich durch Verwendung Strukturierter Hybridmodelle das Extrapolationsverhalten gegenüber der Verwendung Neuronaler Netze erheblich verbessern lässt. Mittlerweile stehen entsprechende Werkzeuge zur standardisierten Erstellung von strukturierten Hybridmodellen zur Verfügung [2].

3.3 Prognose und Optimierung

Auf Basis des erstellten Neuronalen Netzes oder Hybridmodells werden mit Hilfe der Komponente OPAL (Optimierungs-Prognose-Analyse-Labor, Abbildung 3) virtuelle Rezepturen berechnet. Das System erlaubt eine Visualisierung der hochdimensionalen Abhängigkeiten und damit auch eine visuelle Optimierung.

Darüber hinaus ist es möglich auf Basis einer Kombination aus Gradientenverfahren und Monte-Carlo-Simulation, neue Rezepturen zu berechnen, die ein vorgegebenes Eigenschaftsprofil möglichst gut erfüllen. Dazu wird im Falle einer Mehrzieloptimierung mit Gleichungsnebenbedingungen, diese in ein Einzel-Optimierungsproblem im Sinne der „goal attainment“-Methode umformuliert. In der Regel sind bei der industriellen Produktentwicklung komplexe Randbedingungen zu berücksichtigen. Rezepturbestandteile sind häufig gruppiert und es sind beispielsweise Beschränkungen bzgl. der Gesamtzahl zu verwendender Einsatzstoffe und der Anzahl

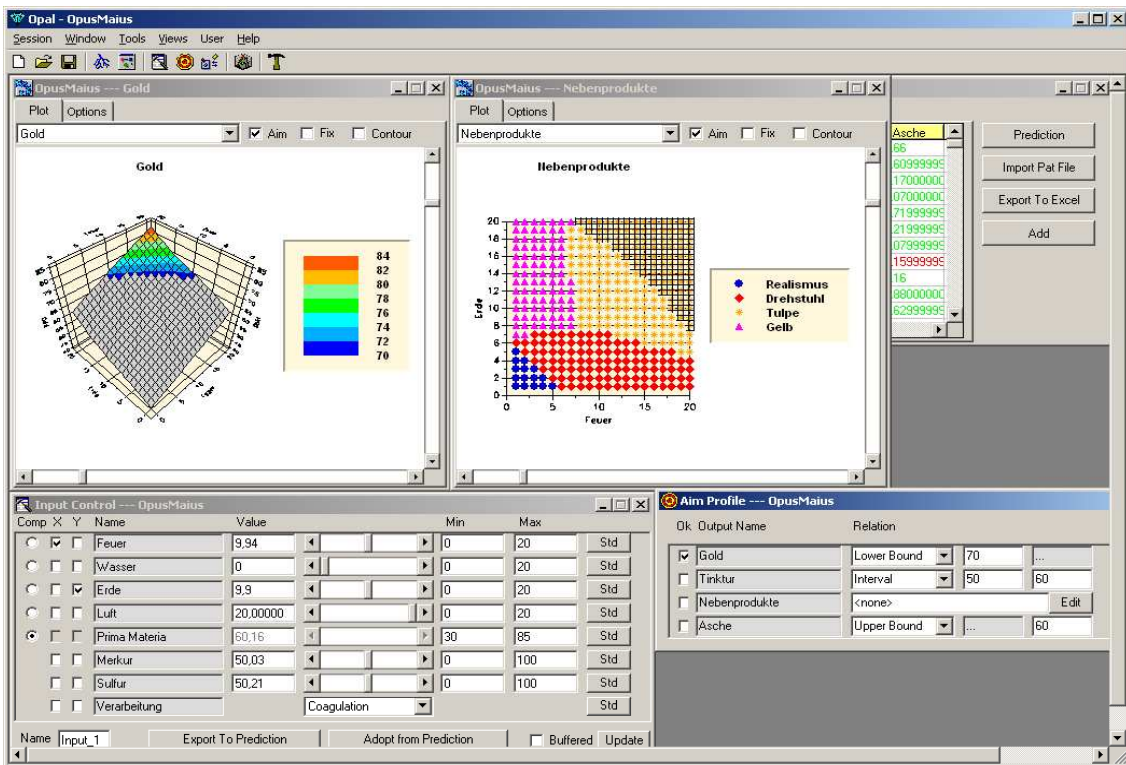


Abbildung 3

zu verwendender Einsatzstoffe aus verschiedenen Gruppen zu berücksichtigen. Des weiteren können nichtnumerische Eingangsgrößen auftreten. Mit Hilfe des Monte-Carlo-Verfahrens können solche komplexe diskrete Randbedingungen bei der Optimierung berücksichtigt werden. Allerdings waren zur Erzielung einer gleichmäßigen Verteilung der zu testenden potentiellen Rezepturen in dem auf Grund der Randbedingungen in der Regel nicht quaderförmigen Parameterraum erhebliche Anstrengungen erforderlich. Eine vorgegebene Anzahl guter Zwischenergebnisse wird anschließend als Startwerte für das lokal konvergente Gradientenverfahren verwendet. Dabei werden alle diskreten Parameter fixiert und nur kontinuierliche Nebenbedingungen berücksichtigt.

3.4 Datenlücken

Die Modellprognose eines Neuronalen Netzes kann immer nur so gut sein wie die zu Grunde liegenden Daten. In eingeschränkter Form gilt dies natürlich auch für Hybridmodelle. Zum Schließen vorhandener Datenlücken ist eine speziell entwickelte Komponente zur Planung neuer Versuchsrezepturen in das Gesamtsystem integriert: Versuche werden so geplant, dass sie untereinander und zu den schon vorhandenen Versuchen möglichst verschieden sind. Dabei können Bereiche im Rezepturraum bevorzugt werden, die nach Maßgabe eines schon vorhandenen Modells gute Produktqualität erwarten lassen. Genau wie im Optimierungsfall sind hier bei realen Produktentwicklungsfragestellungen komplexe Randbedingungen zu berücksichtigen.

4 Anwendungen

4.1 Katalysatorentwicklung

Zur Verbesserung einer Hydrierungsreaktion ist ein neuer Katalysator erforderlich. Dabei soll der Katalysator aus zwei Elementen bestehen, die aus einer Gruppe von insgesamt sechs Elementen auszuwählen sind. Die Konzentrationen der Katalysatorkomponenten können jeweils vier verschiedene Werte annehmen. Weitere Parameter sind die Reaktionstemperatur, die Reaktionszeit, die Kalzinierungstemperatur, der Druck und das Verhältnis zweier Reaktanten. Insgesamt besteht der Parameterraum aus zwölf Variablen in kontinuierlicher oder diskreter Form. Die Zielgrößen bestehen aus fünf Parametern wie Ausbeute, Selektivität, Menge der Nebenprodukte sowie dem Verhältnis spezieller Produkte und Nebenprodukte. Vor Beginn der Versuchsplanung standen insgesamt 14 passende Altdaten aus einem Datenpool von 300 Datensätzen zur Verfügung.

Zur Erstellung eines Neuronalen Netzes mit hoher Prognosegüte konnten die Datenlücken mit Hilfe von drei Versuchsplanungszyklen mit Hilfe des Moduls ECCO [1], das auf einer Evolutionsstrategie basiert, geschlossen werden. Die beiden ersten Iterationen umfassten jeweils 50 Experimente. Für die dritte Iteration wurden 20 weitere Experimente durchgeführt. Basierend auf den 14 Altdaten und den 120 neuen Experimenten wurde für jede Zielgröße ein Neuronales Netzmodell erstellt und darauf basierend mehrere Kandidaten identifiziert, welche die vorgegebenen Ziele erfüllen.

4.2 Verbesserung eines Handelsproduktes

Ein bestehendes Kunststoffprodukt soll bezüglich seines Transparenzverhaltens verbessert werden. Die Frage lautet also: Wie ist die bisherige Rezeptur durch Veränderung der Konzentration der Einsatzstoffe oder durch Austausch von Komponenten zu modifizieren, so dass diese neue Eigenschaft möglichst große Werte annimmt. Dabei sollen natürlich andere Eigenschaften (z.B. Chemikalien- und Temperaturbeständigkeit) weiterhin spezifikationsgerecht bleiben.

Im konkreten Fall basierte die Rezeptur auf insgesamt zehn Komponenten, von denen zwei durch andere austauschbar waren. Bei der Rezepturoptimierung waren sieben Spezifikationen und zwei zusätzliche Nebenbedingungen zu beachten. Die „richtige“ Rezeptur konnte mit zwei Workbench-Iterationen im obigen Sinne basierend auf insgesamt 50 Rezepturvarianten ermittelt werden.

5 Zusammenfassung

Grundgedanke des Workbench Konzepts ist es, dem Anwender z.B. dem Laborleiter, ein System an die Hand zu geben, das ihn bei der Produktentwicklung systematisch unterstützt. Dazu wird die Product Design Workbench in ihrem Umfang und bezüglich der bereitgestellten Funktionalitäten den Erfordernissen des Problems und den Bedürfnissen des Anwenders angepasst. Hierbei beinhaltet die Anpassung der Funktionalitäten sowohl den grundsätzlichen Austausch von Modulen, z.B. die Verwendung eines rigorosen Modells statt eines Neuronalen Netzes, als auch die Konfiguration der Module. Module können beispielsweise so konfiguriert werden, dass der Benutzer bei Routineaufgaben stärker geführt wird. Umgekehrt können für Experten weitergehende Funktionalitäten oder Algorithmen frei geschaltet werden. Letztendlich entscheidet die Aufgabenstellung über die verwendeten Methoden und Module.

Zentraler Bestandteil des Systems zur Unterstützung der Produktentwicklung sind datenbasierte Modelle. Dabei ist es ganz wesentlich, dass auf problemspezifische Modelle, die auf ausgewählten Datensätzen des Gesamt-Datenpools basieren, zurückgegriffen werden kann. Diese Modelle können deshalb als eine Art Straßenkarte zum Auffinden neuer oder verbesserter Produkte verstanden werden. Im Laufe der Produktentwicklungs-Prozedur werden diese Straßenkarten sukzessive zu weiteren Karten verfeinert, die jeweils spezielle Regionen des Rezeptur-Eigenschaftsraums beschreiben. In diesem Sinne ist das vorgestellte System mehr als nur ein „Atlas“ zur Produktentwicklung, es stellt ein komplettes Navigationssystem dar, das den Anwender systematisch zum neuen Produkt führt.

Das System hat sich bereits in zahlreichen Anwendungen bewährt. Es bietet darüber hinaus ein weites Anwendungsspektrum, das sich nicht auf die Polymer- oder Kunststoffentwicklung beschränkt. Gerade auch die Lebensmittel-, Kosmetik- oder Waschmittelindustrie bieten ein hohes Potential zur Effizienzsteigerung auf Basis der Product Design Workbench.

6 Literatur

- [1] Duff, G. D., Ohrenberg, A., Voelkening, S., Boll, M.: *A Screening Workflow for Synthesis and Testing of 10,000 Heterogeneous Catalysts per Day - Lessons Learned*. Macromolecular Rapid Commun. 25, 169-177; 2004
- [2] Mogk, G., Mrziglod, Th., Schuppert, A.: *Application of hybrid models in chemical industry*. European Symposium on Computer Aided Process Engineering - 12, Elsevier Science B.V.; 2002
- [3] Schuppert, A.: *Extrapolability of structured hybrid models: a key to optimization of complex processes*. Proceedings of EquaDiff '99, B.Fiedler, K.Gröger, J.Sprekels Eds., 1135-1151; 2000

Approximationsgenauigkeit und dynamisches Fehlerwachstum der Modellierung mit Takagi-Sugeno Fuzzy Systemen

Horst Schulte

Bosch Rexroth AG, Hydraulics, Glockeraustr.4, 89275 Elchingen
Tel.: 07308 82-2101, horst.schulte@boschrexroth.de

In dieser Arbeit wird der Zusammenhang zwischen der Anzahl der Regeln von Takagi-Sugeno Fuzzy Systemen, der Approximationsgenauigkeit und dem dynamischen Fehlerwachstum dargestellt. Untersucht werden die Approximationseigenschaften von affinen Takagi-Sugeno Fuzzy Systemen zur approximativen Modellierung von nichtlinearen, glatten Zustandsdifferentialgleichungen. Es wird gezeigt, dass für eine gegebene, obere Fehlerschranke ε eines zu approximierenden Gebietes im Zustandsraum die notwendige Anzahl von Fuzzyregeln bzw. lokaler Modelle analytisch bestimmt werden kann. Vorausgesetzt wird zunächst, dass die lokalen Modelle äquidistant über den gesamten Zustandsraum verteilt sind. Da sich diese Art der Fehlerabschätzung nur auf die rechte Seite der Differentialgleichung bezieht und aufgrund der globalen Abschätzung zu stark konservativen Ergebnissen und damit einer hohen Anzahl von lokalen Modellen führt, wird als Basis für eine weiterführende Untersuchung ein iterativer Ansatz zur Konstruktion einer nichtäquidistanten Modellverteilung vorgestellt. Ziel ist es die Anzahl der Fuzzyregeln zur Beschreibung nichtlinearer dynamischer Systems bezogen auf eine gegebene obere Fehlerschranke zu reduzieren.

1 Einleitung

Modelle mit Variablen, die zwischen den Ein- und Ausgangsgrößen eines zu beschreibenden Systems vermitteln und die Systemzustände charakterisieren, werden als Zustandsraummodelle bezeichnet. Diese werden seit Anfang der 60er Jahre neben Ein-Ausgangsmodellen zur Behandlung von Regelungsproblemen verwendet. Zu jedem Zeitpunkt enthalten die Zustandsvariablen alle Informationen, die notwendig sind, um zusammen mit zukünftigen Eingängen die zukünftigen Systemzustände und Ausgänge vorherzusagen zu können. Eine große Anzahl realer Systeme lässt sich durch solch ein Zustandsraummodell, im Folgenden kurz Zustandsmodell genannt, als ein System von Differentialgleichungen und Ausgangsgleichungen darstellen:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}), \\ \mathbf{y} &= \mathbf{g}(\mathbf{x}, \mathbf{u}),\end{aligned}\tag{1}$$

mit dem Zustandsvektor $\mathbf{x} \in X \subset \mathbb{R}^n$, dem Steuer- bzw. Eingangsvektor $\mathbf{u} \in U \subset \mathbb{R}^m$ und dem Ausgangsvektor $\mathbf{y} \in Y \subset \mathbb{R}^p$. Die rechten Seiten der Gleichungen (1) sind im allgemeinen nichtlineare Funktionen in $m + n$ Variablen. In vielen Fällen führt eine Beschränkung auf einen kleinen Arbeitsbereich des realen nichtlinearen Systems zu einem linearen Modell der Form

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u}, \\ \mathbf{y} &= \mathbf{C} \mathbf{x} + \mathbf{D} \mathbf{u}.\end{aligned}\tag{2}$$

Formal lässt sich das Ersatzsystem (2) bei gegebener Systembeschreibung (1) durch Linearisierung mit einer Taylorreihenentwicklung der rechten Seite und Vernachlässigung der höheren Glieder gewinnen. Unter Linearisierung versteht man allgemein das Problem, zu einer gegebenen Differentialgleichung eine andere zu finden, die linear und von möglichst einfacher Bauart ist und deren Lösungen das gleiche asymptotische Verhalten wie die Lösungen der Ursprungsgleichungen haben [2]. Die weite Verbreitung der linearen Form (2) liegt darin begründet, dass die Systemeigenschaften eindeutig durch die zugehörigen Matrizen (A, B, C, D) festgelegt sind. Der Nachweis von Systemeigenschaften, wie z.B. Steuerbarkeit, Beobachtbarkeit und Stabilität, lässt sich hierbei auf einfache algebraische Bedingungen zurückführen. Allerdings gelten diese Eigenschaften nur in einem lokalen Bereich und ob die Systembeschreibung (2) dann tatsächlich das leistet, was man von ihr erwartet, ist eine Frage, auf die es keine generelle Antwort gibt [2].

Anstatt nur ein lineares Ersatzsystem in einem lokalen Bereich (2) zur Beschreibung der Systemdynamik zugrunde zu legen, wird bei Takagi–Sugeno Fuzzy Systemen [5] eine gewichtete Kombination von linearen Zustandsmodellen angesetzt. Die Idee basiert auf der Grundannahme, dass der Arbeitsraum eines allgemein nichtlinearen Systems (1), der in einem begrenzten Gebiet von \mathbb{R}^{n+m} liegt, in einzelne Arbeitsbereiche zerlegbar ist. Die einzelnen Bereiche müssen den Arbeitsraum, der durch die zulässigen Wertebereiche der Systemzustände und Eingänge definiert wird, vollständig überdecken. Die Gültigkeit jedes einzelnen Modells wird dabei durch lokale Gewichtsfunktionen sichergestellt.

2 Approximation nichtlinearer dynamischer Systeme

2.1 Takagi–Sugeno Fuzzy Systeme

Die Modellstruktur, wie sie hier behandelt wird, wurde erstmals von Takagi und Sugeno [5] im Rahmen der Fuzzy Set Theorie als *Fuzzy-Regel basiertes Modell* eingeführt. Die Verwendung von Fuzzy-Regeln mit einer linguistischen Beschreibung ist jedoch für den Einsatz als universeller Approximator von Differentialgleichungssystemen ungeeignet. In dieser Arbeit wird daher eine bekannte *äquivalente defuzzifizierte Form* mit dem Vorteil einer vereinfachten Darstellung und Interpretierbarkeit verwendet. Aus der Literatur sind zwei Arten von Takagi-Sugeno Fuzzy Systemen bekannt:

1. mit statischen Funktionen und
2. mit dynamischen Systemen in der Zustandsraumform.

In dieser Arbeit beschränken wir uns auf die allgemeinere zweite Form. Behandelt werden Kombinationen aus affinen Zustandsmodellen, also linearen Zustandsmodellen plus konstantem Vektor. Wie später gezeigt wird, ergibt sich diese Struktur aus der Linearisierung von (1) um Nichtgleichgewichtspunkte. Die Zustands- und Ausgangsgleichung des affinen Takagi–Sugeno (TS) Fuzzy Systems lautet dabei wie folgt:

$$\dot{\mathbf{x}}(t) = \sum_{i=1}^{N_r} \alpha_i(\mathbf{z}) \mathbf{A}_i \mathbf{x}(t) + \sum_{i=1}^{N_r} \alpha_i(\mathbf{z}) \mathbf{B}_i \mathbf{u}(t) + \sum_{i=1}^{N_r} \alpha_i(\mathbf{z}) \mathbf{a}_i, \quad (3a)$$

$$\mathbf{y}(t) = \sum_{i=1}^{N_r} \alpha_i(\mathbf{z}) \mathbf{C}_i \mathbf{x}(t), \quad (3b)$$

mit $\mathbf{x} \in \mathbb{R}^n$ als Zustandsvektor, $\mathbf{u} \in \mathbb{R}^m$ als Eingangsvektor und $\mathbf{y} \in \mathbb{R}^p$ als Ausgangsvektor. Die Matrizen $\mathbf{A}_i \in \mathbb{R}^{n \times n}$, $\mathbf{B}_i \in \mathbb{R}^{n \times m}$ und $\mathbf{C}_i \in \mathbb{R}^{p \times n}$ mit einem zusätzlichen affinen Anteil $\mathbf{a}_i \in \mathbb{R}^{n \times 1}$ legen die $i = 1, \dots, N_r$ lokalen Zustandsraummodelle fest. Die normalisierte Gewichtsfunktion, gebildet aus dem Produkt der Zugehörigkeitsfunktion $M_{ij}(z_j)$

$$\alpha_i : \quad \mathbb{R}^l \rightarrow \mathbb{R}, \quad \mathbf{z} \mapsto \alpha_i(\mathbf{z}) \quad (3c)$$

$$\alpha_i(\mathbf{z}) = \frac{w_i(\mathbf{z})}{\sum_{k=1}^{N_r} w_k(\mathbf{z})}, \quad w_i(\mathbf{z}) = \prod_{j=1}^l M_{ij}(z_j), \quad (3d)$$

mit den Eigenschaften

$$(i) \quad w_i(\mathbf{z}) \geq 0 \quad \text{und} \quad \sum_{i=1}^{N_r} w_i(\mathbf{z}) > 0 \quad \forall \mathbf{z}, \quad (4)$$

$$(ii) \quad \alpha_i(\mathbf{z}) \geq 0 \quad \text{und} \quad \sum_{i=1}^{N_r} \alpha_i(\mathbf{z}) = 1 \quad \forall \mathbf{z}, \quad (5)$$

definiert den lokalen Gültigkeitsbereich jedes i 'ten Modells in Abhängigkeit von $\mathbf{z} = [z_1, z_2, \dots, z_l]^T$. Die Wahl der Vektordimension l und der Komponenten z_1, \dots, z_l ist abhängig vom zu approximierenden Differentialgleichungssystem und wird aus den Zuständen $\{x_1, \dots, x_n\}$ und den Eingängen $\{u_1, \dots, u_m\}$ gebildet.

2.2 Ableitung von TS Fuzzy Systemen aus nichtlinearen Differentialgleichungen

Ausgangspunkt der Untersuchung ist das Differentialgleichungssystem

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad (6a)$$

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)), \quad (6b)$$

mit

$$\mathbf{f} : \mathbb{R}^{m+n} \rightarrow \mathbb{R}^n, \quad \mathbf{g} : \mathbb{R}^{m+n} \rightarrow \mathbb{R}^p, \quad (6c)$$

wobei $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$ und $\mathbf{y} \in \mathbb{R}^p$. Die partiellen Ableitungen von \mathbf{f} und \mathbf{g} seien bis zur Ordnung zwei auf \mathbb{R}^{m+n} vorhanden und stetig. Gegeben sei eine Lösung $(\mathbf{x}_s(t), \mathbf{y}_s(t))$ von (6) für $t \in [t_0, \infty)$ mit der Eingangsfunktion

$$\mathbf{u}_s : [t_0, \infty) \rightarrow \mathbb{R}^m, \quad (7)$$

deren partiellen Ableitungen bis zur Ordnung zwei ebenfalls vorhanden und stetig sind. Dann lassen sich die rechte Seite von (6a) und die rechte Seite von (6b) nach Taylor in der Umgebung $(\Delta \mathbf{x}, \Delta \mathbf{u})$ dieser Lösung entwickeln:

$$\begin{aligned} \frac{d(\mathbf{x}_s + \Delta \mathbf{x})}{dt} &= \mathbf{f}(\mathbf{x}_s + \Delta \mathbf{x}, \mathbf{u}_s + \Delta \mathbf{u}) \\ &= \mathbf{f}(\mathbf{x}_s, \mathbf{u}_s) + \mathbf{A}(t) \Delta \mathbf{x} + \mathbf{B}(t) \Delta \mathbf{u} + \mathbf{r}_f(\mathbf{x}_s, \mathbf{u}_s, \Delta \mathbf{x}, \Delta \mathbf{u}) \end{aligned} \quad (8a)$$

$$\begin{aligned} \mathbf{y}_s + \Delta \mathbf{y} &= \mathbf{g}(\mathbf{x}_s + \Delta \mathbf{x}, \mathbf{u}_s + \Delta \mathbf{u}) \\ &= \mathbf{g}(\mathbf{x}_s, \mathbf{u}_s) + \mathbf{C}(t) \Delta \mathbf{x} + \mathbf{D}(t) \Delta \mathbf{u} + \mathbf{r}_g(\mathbf{x}_s, \mathbf{u}_s, \Delta \mathbf{x}, \Delta \mathbf{u}) . \end{aligned} \quad (8b)$$

Bekannterweise werden die Matrizen $\mathbf{A}(t)$, $\mathbf{B}(t)$, $\mathbf{C}(t)$ und $\mathbf{D}(t)$ durch die Funktionalmatrizen von \mathbf{f} und \mathbf{g} längs der Lösungstrajektorie \mathbf{x}_s bei gegebener Eingangsfunktion \mathbf{u}_s bestimmt:

$$\mathbf{A}(t) := \mathbf{A}(\mathbf{x}_s(t), \mathbf{u}_s(t)) = \left[\frac{\partial \mathbf{f}(\mathbf{x}_s(t), \mathbf{u}_s(t))}{\partial \mathbf{x}(t)} \right] , \quad (9a)$$

$$\mathbf{B}(t) := \mathbf{B}(\mathbf{x}_s(t), \mathbf{u}_s(t)) = \left[\frac{\partial \mathbf{f}(\mathbf{x}_s(t), \mathbf{u}_s(t))}{\partial \mathbf{u}(t)} \right] , \quad (9b)$$

$$\mathbf{C}(t) := \mathbf{C}(\mathbf{x}_s(t), \mathbf{u}_s(t)) = \left[\frac{\partial \mathbf{g}(\mathbf{x}_s(t), \mathbf{u}_s(t))}{\partial \mathbf{x}(t)} \right] , \quad (9c)$$

$$\mathbf{D}(t) := \mathbf{D}(\mathbf{x}_s(t), \mathbf{u}_s(t)) = \left[\frac{\partial \mathbf{g}(\mathbf{x}_s(t), \mathbf{u}_s(t))}{\partial \mathbf{u}(t)} \right] , \quad (9d)$$

mit $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^T$ und $\mathbf{u} = [u_1 \ u_2 \ \dots \ u_p]^T$. Vernachlässigt man auf der rechten Seite von (8) die Restterme \mathbf{r}_f und \mathbf{r}_g , so erhält man bezogen auf die Ausgangssituation (6) ein neues System mit der Zustandsgleichung

$$\begin{aligned} \dot{\boldsymbol{\xi}}(t) &= \mathbf{f}(\mathbf{x}_s(t), \mathbf{u}_s(t)) + \mathbf{A}(t) (\boldsymbol{\xi}(t) - \mathbf{x}_s(t)) + \mathbf{B}(t) (\mathbf{u}(t) - \mathbf{u}_s(t)) , \\ \boldsymbol{\xi}(t_0) &= \mathbf{x}_s(t_0) , \end{aligned} \quad (10a)$$

mit dem Zustandsvektor $\boldsymbol{\xi}(t) \in \mathbb{R}^n$, der im allgemeinen $\boldsymbol{\xi}(t) \neq \mathbf{x}_s(t)$ für $t > t_0$ ist und der Ausgangsgleichung

$$\tilde{\mathbf{y}}(t) = \mathbf{g}(\mathbf{x}_s(t), \mathbf{u}_s(t)) + \mathbf{C}(t) (\boldsymbol{\xi}(t) - \mathbf{x}_s(t)) + \mathbf{D}(t) (\mathbf{u}(t) - \mathbf{u}_s(t)) \quad (10b)$$

wobei $\tilde{\mathbf{y}}(t) \in \mathbb{R}^p$.

Gegeben sei nun eine endliche Menge \mathbb{G}_s von diskreten Punkten

$$\mathbb{G}_s = \{\mathbf{x}_i, \mathbf{u}_i\} \in (\mathbf{x}_s(t), \mathbf{u}_s(t)) , \quad i = 1, \dots, N_r , \quad t \in [t_0, \infty) \subset \mathbb{R} \quad (11)$$

mit $\mathbb{G}_s \subset \mathbb{G} \subseteq \mathbb{R}^n \times \mathbb{R}^m$. Zur Approximation der rechten Seite von (10a) wird nun eine gewichtete Kombination von Zustandsmodellen angesetzt:

$$\dot{\tilde{\boldsymbol{\xi}}}(t) = \sum_{i=1}^{N_r} \alpha_i(\tilde{\boldsymbol{\xi}}(t), \mathbf{u}(t)) \left[\mathbf{A}(\mathbf{x}_i, \mathbf{u}_i) (\tilde{\boldsymbol{\xi}}(t) - \mathbf{x}_i) + \mathbf{B}(\mathbf{x}_i, \mathbf{u}_i) (\mathbf{u}(t) - \mathbf{u}_i) + \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i) \right] , \quad (12)$$

$$\tilde{\boldsymbol{\xi}}(t_0) = \mathbf{x}_s(t_0)$$

mit dem Zustandsvektor $\tilde{\boldsymbol{\xi}}(t) \in \mathbb{R}^n$, der im allgemeinen $\tilde{\boldsymbol{\xi}}(t) \neq \boldsymbol{\xi}(t)$ ist für $t > t_0$ und der Gewichtsfunktion $\alpha_i(\tilde{\boldsymbol{\xi}}(t), \mathbf{u}(t))$. Das heißt, in der folgenden Untersuchung wird vorausgesetzt, dass

$$\mathbf{z} = \left[\tilde{\boldsymbol{\xi}}^T \ \mathbf{u}^T \right]^T \quad (13)$$

ist. Fassen wir nun die statischen Terme aus (12) zusammen

$$\begin{aligned} \dot{\tilde{\boldsymbol{\xi}}}(t) = & \sum_{i=1}^{N_r} \alpha_i(\tilde{\boldsymbol{\xi}}(t), \mathbf{u}(t)) \mathbf{A}(\mathbf{x}_i, \mathbf{u}_i) \tilde{\boldsymbol{\xi}}(t) + \sum_{i=1}^{N_r} \alpha_i(\tilde{\boldsymbol{\xi}}(t), \mathbf{u}(t)) \mathbf{B}(\mathbf{x}_i, \mathbf{u}_i) \mathbf{u}(t) \\ & + \sum_{i=1}^{N_r} \alpha_i(\tilde{\boldsymbol{\xi}}(t), \mathbf{u}(t)) [\mathbf{f}(\mathbf{x}_i, \mathbf{u}_i) - \mathbf{A}(\mathbf{x}_i, \mathbf{u}_i) \mathbf{x}_i - \mathbf{B}(\mathbf{x}_i, \mathbf{u}_i) \mathbf{u}_i] , \end{aligned}$$

dann ergibt sich mit den Abkürzungen

$$\mathbf{A}_i := \mathbf{A}(\mathbf{x}_i, \mathbf{u}_i), \quad \mathbf{B}_i := \mathbf{B}(\mathbf{x}_i, \mathbf{u}_i), \quad (14a)$$

$$\mathbf{a}_i := \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i) - \mathbf{A}(\mathbf{x}_i, \mathbf{u}_i) \mathbf{x}_i - \mathbf{B}(\mathbf{x}_i, \mathbf{u}_i) \mathbf{u}_i \quad (14b)$$

eine Zustandsdifferentialgleichung der Form

$$\dot{\tilde{\boldsymbol{\xi}}}(t) = \sum_{i=1}^{N_r} \alpha_i(\tilde{\boldsymbol{\xi}}(t), \mathbf{u}(t)) \mathbf{A}_i \tilde{\boldsymbol{\xi}}(t) + \sum_{i=1}^{N_r} \alpha_i(\tilde{\boldsymbol{\xi}}(t), \mathbf{u}(t)) \mathbf{B}_i \mathbf{u}(t) + \sum_{i=1}^{N_r} \alpha_i(\tilde{\boldsymbol{\xi}}(t), \mathbf{u}(t)) \mathbf{a}_i, \quad (14c)$$

die der Form eines TS Fuzzy Systems (3) entspricht. Von Interesse ist nun die Abschätzung des maximalen Fehlers, der durch die Vernachlässigung der lokal gültigen Restterme \mathbf{r}_f und \mathbf{r}_g in (8) und der Diskretisierung mit (11) zwangsläufig auftritt.

2.3 Fehleranalyse

Basierend auf den Untersuchungen in [3] mit der Erweiterung für den MIMO Fall in [4] wird der Zusammenhang zwischen der Anzahl der lokal gewichteten Zustandsmodelle und der Approximationsgenauigkeit dargestellt. Zur Quantifizierung der Genauigkeit wird durchgehend die euklidische Norm

$$\|\mathbf{e}\|_2 := \sqrt{\sum_{i=1}^n e_i^2} \quad (15)$$

vom Fehlervektor \mathbf{e} verwendet. Vorgestellt werden zwei analytische Beziehungen zur Bestimmung der notwendigen Anzahl lokaler Modelle bei einer gegebenen globalen oberen Fehlerschranke $\varepsilon \in \mathbb{R}_+$ für ein begrenztes zusammenhängendes Gebiet. Für die weiteren Betrachtungen werden vorab noch die folgenden Definitionen eingeführt:

1. $\text{ceil}(\cdot)$ ist eine Funktion

$$\text{ceil} : \mathbb{R} \rightarrow \mathbb{N}, \quad x \mapsto \text{ceil}(x), \quad (16)$$

die den kleinsten ganzzahligen Wert liefert, der nicht kleiner als x ist. Zum Beispiel ist $\text{ceil}(2.567) = 3$.

2. $\mathbf{f}_{TS}(\mathbf{x}, \mathbf{u})$ ist eine gewichtete Summe

$$\mathbf{f}_{TS}(\mathbf{x}, \mathbf{u}) := \sum_{i=1}^{N_r} \alpha_i(\mathbf{x}, \mathbf{u}) \mathbf{f}_i(\mathbf{x}, \mathbf{u}) \quad (17)$$

von affinen vektorwertigen Funktionen

$$\mathbf{f}_i(\mathbf{x}, \mathbf{u}) = \mathbf{A}_i \mathbf{x} + \mathbf{B}_i \mathbf{u} + \mathbf{a}_i, \quad i = 1, \dots, N_r \quad (18)$$

mit $\mathbf{x} \in X \subset \mathbb{R}^n$ und $\mathbf{u} \in Y \subset \mathbb{R}^m$. Die Funktion (17) entspricht dabei der rechten Seite von (3a).

3. Die Distanz $d(\mathbf{f}, \mathbf{f}_{TS})$ zwischen der Originalfunktion $\mathbf{f}(\mathbf{x}, \mathbf{u})$ und der Approximation mit $\mathbf{f}_{TS}(\mathbf{x}, \mathbf{u})$ ist definiert durch das Supremum der euklidischen Fehlernom auf $\mathbb{G} \subseteq X \times U$

$$d(\mathbf{f}, \mathbf{f}_{TS}) := \sup_{(\mathbf{x}, \mathbf{u}) \in \mathbb{G}} \|\mathbf{f}(\mathbf{x}, \mathbf{u}) - \mathbf{f}_{TS}(\mathbf{x}, \mathbf{u})\|_2 \quad (19)$$

4. λ_H ist der maximale Eigenwert der Hesseschen Matrix:

$$\lambda_H = \max_{i,j} [\lambda_{H_{ij}}] \quad \text{mit} \quad \lambda_{H_{ij}} = \text{Eig} \left[\frac{\partial^2 f_j(\vartheta, \boldsymbol{\psi})}{\partial \boldsymbol{\psi} \partial \boldsymbol{\psi}} \right] \quad (20)$$

5. $C^2(X)$ ist die Menge aller Funktionen, die auf X definiert ist und deren partielle Ableitungen bis zur Ordnung zwei alle auf X vorhanden und stetig sind.

Theorem 1. Gegeben ist $\mathbf{f}(\mathbf{x}, \mathbf{u}) \in C^2(\mathbb{G})$. $\mathbb{G} \subseteq X \times U$ sei eine kompakte Menge (der zu approximierende Bereich), definiert durch

$$\mathbb{G} = \left\{ \boldsymbol{\psi} = [\psi_1 \ \psi_2 \ \dots \ \psi_{n+m}]^T \in \mathbb{R}^{n+m} \mid |\psi_j - \zeta_j| \leq \frac{\beta_j}{2}, \ j = 1, \dots, n+m \right\}, \quad (21)$$

wobei der Zustands- und Eingangsvektor mit

$$\boldsymbol{\psi} := [\mathbf{x}^T \ \mathbf{u}^T]^T, \quad \mathbf{x}^T = [x_1 \ x_2 \ \dots \ x_n], \quad \mathbf{u}^T = [u_1 \ u_2 \ \dots \ u_m]$$

zusammengefasst wird. \mathbb{G} wird durch die Lage (Lage des Zentrums) $\boldsymbol{\zeta} = [\zeta_1 \ \zeta_2 \ \dots \ \zeta_{n+m}]^T$ und die Breite β_j pro Dimension mit $j = 1, \dots, n+m$ festgelegt. Für eine absolute Fehlerschranke von $\varepsilon > 0$ erfüllt der Ansatz $\mathbf{f}_{TS}(\mathbf{x}, \mathbf{u})$ nach (17) mit N_r Gewichtsfunktionen $\alpha_i(\mathbf{x}, \mathbf{u}) : \mathbb{R}^{n+m} \rightarrow \mathbb{R}_+$, die durch 1 beschränkt sind mit den Bedingungen

$$\forall (\mathbf{x}, \mathbf{u}) : \sum_{i=1}^{N_r} \alpha_i(\mathbf{x}, \mathbf{u}) = 1, \quad \alpha_i(\mathbf{x}, \mathbf{u}) \equiv 0 \text{ falls } \|\boldsymbol{\psi} - \boldsymbol{\psi}_i\|_2 > R \text{ und sonst } \alpha_i(\mathbf{x}, \mathbf{u}) \geq 0$$

und der Modellanzahl

$$N_r = \prod_{i=1}^{n+m} \text{ceil} \left(\frac{\beta_i}{2\sqrt{2\varepsilon}} \sqrt{\lambda_H (n+m) \sqrt{n}} \right) \quad (22)$$

und der Distanz (19) als Fehlerkriterium die Ungleichung

$$d(\mathbf{f}, \mathbf{f}_{TS}) \leq \varepsilon.$$

Anmerkung zur Beweisführung des Theorems: Wegen der gleichmäßigen Stetigkeit der stetigen Funktion \mathbf{f} auf dem Kompaktum X kann ein L gefunden werden, so dass

$$\|\mathbf{r}_i(\mathbf{x})\|_2 = \|\mathbf{f}(\mathbf{x}) - \mathbf{f}_i(\mathbf{x})\|_2 \leq L_i \|\mathbf{x} - \mathbf{x}_i\|_2 \leq L_i R_i = \varepsilon_i \quad (23)$$

mit $L_i = L$ für jedes $\mathbf{x} \in X$ und einem dazu gewählten \mathbf{x}_i richtig ist. Wähle $R := \frac{\varepsilon}{L}$ und eine endliche Überdeckung von X mit N_r Hyperkugeln $B_{R_i}[\mathbf{x}_i]$ mit dem Mittelpunkt \mathbf{x}_i und den gleichen Radien $R_i = R$, so dass (23) mit $\varepsilon_i = \varepsilon$ gilt für jedes $\mathbf{x} \in X$ und einem dazu gewählten \mathbf{x}_i . Mit der Identität von $\mathbf{f}(\mathbf{x}) = \sum_{i=1}^{N_r} \alpha_i(\mathbf{x}) \mathbf{f}(\mathbf{x})$, welche sich aus der Voraussetzung $\sum_{i=1}^{N_r} \alpha_i(\mathbf{x}) = 1$ ergibt, erfolgt der Beweis in [4] durch eine Fehlerabschätzung nach mehrmaliger Anwendung der Dreiecksungleichung und der Beziehung

$$\mathbf{x}^T \mathbf{A} \mathbf{x} = \mathbf{x}^T \sum_{i=1}^n \mathbf{p}_i \lambda_i \mathbf{x}_i \leq \mathbf{x}^T \lambda \sum_{i=1}^n \mathbf{p}_i \mathbf{x}_i, = \lambda \mathbf{x}^T \mathbf{x}$$

mit $\mathbf{p}_i \in \mathbb{R}^n$ und $\lambda = \max_i [\lambda_i]$ als maximaler Eigenwert von \mathbf{A} .

Die Fehlerabschätzung, die der Berechnung der hinreichenden Modellanzahl N_r mit (22) zugrunde liegt, berücksichtigt nicht die spezielle Struktur des nichtlinearen Systems (6a). Die bisherige Fehleranalyse geht von dem ungünstigsten Fall aus, dass alle n Zustände und m Eingänge nichtlinear eingehen. Dies kann schon bei relativ einfachen Systemen zu sehr konservativen Abschätzungen führen [4]. Falls das zu approximierende System in einen linearen und nichtlinearen Anteil zerlegbar ist, muss dagegen nur noch der nichtlineare Anteil bei der Berechnung der Modellanzahl berücksichtigt werden. Dieser Sachverhalt bildet das folgende Theorem ab:

Theorem 2. Gegeben ist $\mathbf{f}(\boldsymbol{\psi}) = \mathbf{F} \boldsymbol{\psi}_L + \mathbf{f}_{nl}(\boldsymbol{\psi}_N)$ mit $\boldsymbol{\psi}_L \in \mathbb{G}_L$, $\boldsymbol{\psi}_N \in \mathbb{G}_N$ und der kompakten Menge $\mathbb{G} = \mathbb{G}_L \times \mathbb{G}_N$. Die Vektoren $\boldsymbol{\psi}_L$ und $\boldsymbol{\psi}_N$ setzen sich aus einzelnen Komponenten der Vektoren \mathbf{x} und \mathbf{u} zusammen. Es sei $\boldsymbol{\psi}_L := [\mathbf{x}_L^T \mathbf{u}_L^T]^T$ und $\boldsymbol{\psi}_N := [\mathbf{x}_N^T \mathbf{u}_N^T]^T$. Der nichtlineare Anteil von $\mathbf{f}(\boldsymbol{\psi})$ ist $\mathbf{f}_{nl} : \boldsymbol{\psi}_N \rightarrow \mathbb{R}^n$ mit $n_N = \dim(\boldsymbol{\psi}_N)$. Der lineare Anteil wird beschrieben durch $\mathbf{F} \boldsymbol{\psi}_L$ mit $\mathbf{F} \in \mathbb{R}^{n \times n_L}$ und $n_L = \dim(\boldsymbol{\psi}_L)$ als die Anzahl der Komponenten, die nur linear in $\mathbf{f}(\boldsymbol{\psi})$ eingehen. Für eine absolute Fehlerschranke von $\varepsilon > 0$ erfüllt der Ansatz

$$\mathbf{f}_{TS}(\boldsymbol{\psi}) = \mathbf{F} \boldsymbol{\psi}_L + \sum_{i=1}^{N_r} \alpha_i(\boldsymbol{\psi}_N) \mathbf{f}_i(\boldsymbol{\psi}_N), \quad \boldsymbol{\psi}_N = \mathbf{z} \in \mathbb{Z} = \mathbb{G}_N, \quad (24a)$$

$$\mathbf{f}_i(\boldsymbol{\psi}_N) = \mathbf{f}_i(\mathbf{x}_N, \mathbf{u}_N) = \mathbf{A}_i \mathbf{x}_N + \mathbf{B}_i \mathbf{u}_N + \mathbf{a}_i \quad (24b)$$

mit N_r Gewichtsfunktionen $\alpha_i(\boldsymbol{\psi}_N) : \mathbb{R}^{n_N} \rightarrow \mathbb{R}_+$, die durch 1 beschränkt sind mit den Bedingungen

$$\forall \boldsymbol{\psi}_N : \sum_{i=1}^{N_r} \alpha_i(\boldsymbol{\psi}_N) = 1, \quad \alpha_i(\boldsymbol{\psi}_N) \equiv 0 \text{ falls } \|\boldsymbol{\psi}_N - \boldsymbol{\psi}_{N_i}\|_2 > R \text{ und sonst } \alpha_i(\boldsymbol{\psi}_N) \geq 0$$

und der Modellanzahl

$$N_r = \prod_{i=1}^{n_N} \text{ceil} \left(\frac{\beta_i}{2 \sqrt{2} \varepsilon} \sqrt{\lambda_H n_N \sqrt{n}} \right) \quad (25)$$

und der Distanz (19) als Fehlerkriterium, die Ungleichung

$$d(\mathbf{f}, \mathbf{f}_{TS}) \leq \varepsilon .$$

In (25) ist $n_N = \dim(\psi_N)$ und $\lambda_H = \max_{i,j} [\lambda_{H_{ij}}]$, vgl. (20).

Anmerkungen: 1.) Im Gegensatz zum Theorem 1 wird der maximale Eigenwert der Hesseschen Matrix nur aus dem nichtlinearen Anteil \mathbf{f}_{nl} und nicht aus dem gesamten System gebildet. 2.) Der Beweis von Theorem 2 wird ausgeführt in [4].

3 Approximative Modellierung eines nichtlinearen Feder-Dämpfer-Masse Systems

Das mathematische Modell eines mechanischen Oszillators mit nichtlinearer Feder- und nichtlinearer Dämpfercharakteristik sei gegeben durch

$$m \ddot{x} + d(\dot{x}) + k(x) = F, \quad x(t_0) = 0, \quad \dot{x}(t_0) = 0, \quad (26)$$

mit x als Auslenkung des starren Körpers aus der vorgegebenen Lage und einer in x -Richtung auf den starren Körper einwirkende äußere Kraft $F(t)$. Die mechanische Eigenschaft der Feder wird durch eine wegabhängige Funktion $k(x)$ modelliert mit

$$k(x) = c_0 x + c_1 x^3, \quad c_0 > 0, \quad c_1 > 0 \quad . \quad (27)$$

Der Dämpfercharakteristik $d(\dot{x})$ wird durch

$$d(\dot{x}) = d_0 \dot{x} + \operatorname{sgn}(\dot{x}) d_1 \sqrt{|\dot{x}|} = \begin{cases} d_0 \dot{x} - d_1 \sqrt{|\dot{x}|} & \text{falls } \dot{x} < 0 \\ d_0 \dot{x} + d_1 \sqrt{|\dot{x}|} & \text{sonst} \end{cases} \quad (28)$$

festgelegt. Die Stetigkeit an der Stelle $\dot{x} = 0$ ergibt sich aus

$$\lim_{\dot{x} \rightarrow +0} d(\dot{x}) = \lim_{\dot{x} \rightarrow -0} d(\dot{x}) = 0 .$$

Die Modellparameter in (27), (26) und (28) seien mit

$m = 2$ als Masse des starren Körpers in kg,

$d_0 = 0.5$ als Dämpfungskoeffizient des geschwindigkeitsproportionalen Terms von (28) in N/m/s,

$d_1 = 0.8$ als Dämpfungskoeffizient des nichtlinearen Terms von (28) in N/(m/s)^{1/2},

$c_0 = 20$ als Steifigkeitskoeffizient des wegproportionalen Terms der Federkennlinie (27) in N/m,

$c_1 = 215$ als Steifigkeitskoeffizient des nichtlinearen Terms der Federkennlinie (27) in N/m³

bekannt. Die Differentialgleichung zweiter Ordnung (26) lässt sich nach Einführung der Zustandsvariablen

$$x_1 := x, \quad x_2 := \dot{x}$$

und der Festlegung der Eingangs- und Ausgangsvariable mit

$$u := F, \quad y := x$$

in die Zustandsraumdarstellung

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u) = \begin{bmatrix} f_1(\mathbf{x}, u) \\ f_2(\mathbf{x}, u) \end{bmatrix}$$

überführen. Nach kurzer Umrechnung erhält man die Form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{c_0}{m} & -\frac{d_0}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{c_1}{m} x_1^3 - \frac{d_1}{m} \operatorname{sgn}(x_2) \sqrt{|x_2|} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u, \quad (29a)$$

$$y = [1 \ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \quad (29b)$$

Im *ersten Schritt* muss zur Approximation von (29a) mit dem Ansatz (3) die maximal zulässige Fehlerschranke ε und der Arbeitsbereich \mathbb{G} festgelegt werden. Die Fehlerschranke sei $\varepsilon = 0.2$. Der maximale Betrag der äußeren Kraft sei $F_{max} = 3.5$ N, der maximale Betrag der Auslenkung sei $x_{max} = 0.19$ m und die maximal zulässige Geschwindigkeit des Starrkörpers ist durch $\dot{x}_{max} = 0.3$ m/s begrenzt. Die Koordinaten von \mathbb{G} werden ohne Beschränkung der Allgemeinheit so gewählt, dass das Zentrum $\zeta = [\zeta_1 \ \zeta_2 \ \zeta_3]^T$ von \mathbb{G} im Ursprung liegt. Aus den zuvor festgelegten Schranken $|x_1| \leq x_{max}$, $|x_2| \leq \dot{x}_{max}$, $|u| \leq F_{max}$ lassen sich die β_j bestimmen:

$$\beta_1 = 2 x_{max} = 0.38, \quad \beta_2 = 2 \dot{x}_{max} = 0.6, \quad \beta_3 = 2 F_{max} = 7. \quad (30)$$

Im *zweiten Schritt* wird die notwendige lokale Modellanzahl N_r unter Ausnutzung der Modellstruktur berechnet. Aus (29a) folgt direkt

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} f_1(\mathbf{x}, u) \\ f_2(\mathbf{x}, u) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ -\frac{c_0}{m} & -\frac{d_0}{m} & \frac{1}{m} \end{bmatrix}}_{\mathbf{F}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ u \end{bmatrix}}_{\psi_L} + \underbrace{\begin{bmatrix} 0 \\ -\frac{c_1}{m} x_1^3 - \frac{d_1}{m} \operatorname{sgn}(x_2) \sqrt{|x_2|} \end{bmatrix}}_{\mathbf{f}_{nl}(\psi_N)} \quad (31)$$

mit $\psi_N = [x_1 \ x_2]^T$. Dies ergibt sich aus dem Sachverhalt, dass nur die Zustände und nicht der Eingang nichtlinear in die Zustandsgleichung eingehen. Aus dem nichtlinearen Anteil $\mathbf{f}_{nl} = [f_{nl_1} \ f_{nl_2}]^T$ von (31) werden die Hesseschen Matrizen

$$\mathbf{H}_{ij} = \begin{bmatrix} \frac{\partial^2 f_j}{\partial x_1^2} & \frac{\partial^2 f_j}{\partial x_2 \partial x_1} \\ \frac{\partial^2 f_j}{\partial x_1 \partial x_2} & \frac{\partial^2 f_j}{\partial x_2^2} \end{bmatrix}$$

für $j = 1, 2$ bestimmt und wir erhalten

$$\mathbf{H}_{i1} = 0_{2 \times 2}, \quad \mathbf{H}_{i2} = \begin{bmatrix} -6 \frac{c_1}{m} x_{1_i} & 0 \\ 0 & \frac{d_1}{4} \operatorname{sgn}(x_{2_i}) |x_{2_i}|^{-\frac{3}{2}} \end{bmatrix}.$$

Die zugehörigen Eigenwerte lauten

$$\lambda_{H_{i1}} = \operatorname{Eig}[\mathbf{H}_{i1}] = [0, 0], \quad \lambda_{H_{i2}} = \operatorname{Eig}[\mathbf{H}_{i2}] = \left[-6 \frac{c_1}{m} x_{1_i}, \quad \frac{d_1}{4} \operatorname{sgn}(x_{2_i}) |x_{2_i}|^{-\frac{3}{2}} \right].$$

Der maximale Wert der größten Eigenwerte von $\lambda_{H_{i1}}$ und $\lambda_{H_{i2}}$ ist

$$\begin{aligned}\lambda_H &= \max \left[\max_i [0, 0] ; \max_i \left[-6 \frac{c_1}{m} x_{1_i} , \frac{d_1}{4} \operatorname{sgn}(x_{2_i}) |x_{2_i}|^{-\frac{3}{2}} \right] \right] \\ &= \max \left[0, 0, -6 \frac{c_1}{m} (-x_{max}), \frac{d_1}{4} \operatorname{sgn}(\epsilon_{\dot{x}}) |\epsilon_{\dot{x}}|^{-\frac{3}{2}} \right] \\ &= \max [0, 0, 122.55, 70.711] = 122.55 .\end{aligned}$$

Mit diesem Wert, dem zuvor festgelegten Bereich (30), der Ordnung $n_N = 2$ des nichtlinearen Anteils, der Systemordnung $n = 2$ und der zulässigen Fehlerschranke von $\varepsilon = 0.2$ ist die Modellanzahl nach (25) berechenbar:

$$\begin{aligned}N_r &= \prod_{i=1}^{n_N} \operatorname{ceil} \left(\frac{\beta_i}{2 \sqrt{2\varepsilon}} \sqrt{\lambda_H n_N \sqrt{n}} \right) \\ &= \operatorname{ceil} \left(\frac{0.38}{2 \sqrt{2 \cdot 0.2}} \sqrt{122.55 \cdot 2 \sqrt{2}} \right) \cdot \operatorname{ceil} \left(\frac{0.6}{2 \sqrt{2 \cdot 0.2}} \sqrt{122.55 \cdot 2 \sqrt{2}} \right) \\ &= 6 \cdot 9 = 54 .\end{aligned}$$

Im *dritten Schritt* werden die Matrizen \mathbf{A}_i , \mathbf{B}_i und Vektoren \mathbf{a}_i der $i = 1, \dots, 54$ lokalen Modelle aus der Taylorreihenentwicklung um die festgelegten Arbeitspunkte

$$\begin{aligned}(x_{1_i}, x_{2_i}) &\in \{-0.16, -0.096, -0.032, 0.032, 0.096, 0.16\} \\ &\times \{-0.285, -0.2125, -0.14, -0.0675, 0.005, 0.0775, 0.15, 0.2225, 0.295\} \\ &\quad (32)\end{aligned}$$

für

$$\mathbb{G}_s = \{[x_{1_i} \ x_{2_i} \ u_i]^T\} \quad (33)$$

berechnet. Die x_{1_i} und x_{2_i} Komponenten sind äquidistant über den Definitionsbereich \mathbf{f}_{nl} verteilt mit

$$\mathbf{z} := [z_1 \ z_2]^T \quad \text{wobei} \quad z_1 := x_1, \quad z_2 := x_2 . \quad (34)$$

Aus der Taylorreihenentwicklung von (29a) mit (9a), (9b) und (14a) ergeben sich die Systemmatrizen

$$\begin{aligned}\mathbf{A}_i &= \begin{bmatrix} 0 & 1 \\ a_{21}(x_{1_i}) & a_{22}(x_{2_i}) \end{bmatrix} \quad \text{mit} \quad a_{21}(x_{1_i}) = -\frac{1}{m} (c_0 + 3 c_1 x_{1_i}^2) , \\ &\quad a_{22}(x_{2_i}) = -\frac{1}{m} \left[d_0 + d_1 \left(\frac{\frac{3}{2} \operatorname{sgn}(x_{2_i}) \sqrt{|x_{2_i}|}}{x_{2_i}} - \frac{|x_{2_i}|^{3/2}}{x_{2_i}^2} \right) \right] \\ &\quad (35)\end{aligned}$$

und die Eingangsmatrizen

$$\mathbf{B}_i = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \quad (36)$$

sowie mit (14b) die Vektoren

$$\mathbf{a}_i = \begin{bmatrix} 0 \\ f_2(\mathbf{x}_i, u_i) - a_{21}(x_{1_i}) x_{1_i} - a_{22}(x_{2_i}) x_{2_i} - \frac{1}{m} u_i \end{bmatrix} .$$

Mit den zuvor berechneten Koeffizienten $a_{21}(x_{1_i})$ und $a_{22}(x_{2_i})$ und der zweiten Komponente der nichtlinearen Funktion (29a) am Entwicklungspunkt $f_2(\mathbf{x}_i, u_i)$ erhalten wir dann

$$\mathbf{a}_i = \begin{bmatrix} 0 \\ \frac{2}{m} c_1 x_{1_i}^3 + \frac{d_1}{m} \left(\frac{1}{2} \operatorname{sgn}(x_{2_i}) \sqrt{|x_{2_i}|} - \frac{|x_{2_i}|^{3/2}}{x_{2_i}} \right) \end{bmatrix}. \quad (37)$$

Der affine Anteil (37) ist unabhängig von der Eingangsgröße u . Dies entspricht dem Sachverhalt, dass (34) nur von den Zuständen abhängt. Die Ausgangsmatrizen \mathbf{C}_i sind für alle Modelle gleich: $\mathbf{C}_i = \mathbf{C} = [1 \ 0]$, vgl. (29b). Daraus folgt nun das Approximationsmodell

$$\begin{aligned} \dot{\mathbf{x}} &= \sum_{i=1}^{54} \alpha_i(x_1, x_2) \mathbf{A}_i \mathbf{x} + \mathbf{B} u + \sum_{i=1}^{54} \alpha_i(x_1, x_2) \mathbf{a}_i, \\ \mathbf{y} &= \mathbf{C} \mathbf{x}. \end{aligned} \quad (38)$$

Man beachte, dass die Gleichgewichtslagen des nichtlinearen Systems (31) nicht den Arbeitspunkten der lokalen Modelle entsprechen. Im stationären Fall setzt sich (38) also immer aus mindestens zwei Modellen zusammen. Dies kann zu einem stationären Approximationsfehler führen.

Im *vierten und letzten Schritt* werden die Zugehörigkeitsfunktionen der einzelnen Gewichtsfunktionen $\alpha_i(x_1, x_2)$ für (38) bestimmt. Die genaue Lage der Modellzugehörigkeiten im zweidimensionalen Arbeitsraum wird durch die x_1 -abhängigen Zugehörigkeitsfunktionen M_{i1} und die x_2 -abhängigen Zugehörigkeitsfunktionen M_{i2} festgelegt, vgl. (3d). In Bild 1 werden alle Gewichtsfunktionen $\alpha_i(x_1, x_2)$ der 54 lokalen Modelle über z_1 und z_2 dargestellt. Der maximale Wert jeder einzelnen Gewichtsfunktionen entspricht dem Entwicklungspunkt der Taylorreihenentwicklung. Das TS Fuzzy System (38) ist nun vollständig bestimmt.

Zur *Modellvalidierung* wird das System in einer Simulation exemplarisch mit zwei sinusförmigen Eingangssignalen $u_{1,2}$, mit denen der gesamte Arbeitsraum und damit alle lokalen Modelle während der Berechnung in variierenden Kombinationen sukzessiv angeregt werden, untersucht. In Bild 2 sind für

$$\begin{aligned} u_1 &: u_1(t) = u_0 \sin(2\pi ft) \quad \text{mit} \quad u_0 := 2.5 \text{ N}, \quad f = 0.28 \text{ Hz} \\ u_2 &: u_2(t) = u_0 \sin(2\pi ft) \quad \text{mit} \quad u_0 := 5.0 \text{ N}, \quad f = 0.112 \text{ Hz} \end{aligned} \quad (39)$$

die Verläufe der Fehlernorm $\|\mathbf{f}(\mathbf{x}, u) - \mathbf{f}_{TS}(\mathbf{x}, u)\|_2$ über die gesamte Simulationsdauer dargestellt. Man erkennt, dass der zulässige Fehler von $\varepsilon = 0.2$ nicht überschritten wird. Für die gesamte Simulationsdauer sind die Ungleichungen

$$\begin{aligned} \sup_t \|\mathbf{f}(\mathbf{x}, u_1) - \mathbf{f}_{TS}(\mathbf{x}, u_1)\|_2 &\leq \varepsilon = 0.2 \\ \sup_t \|\mathbf{f}(\mathbf{x}, u_2) - \mathbf{f}_{TS}(\mathbf{x}, u_2)\|_2 &\leq \varepsilon = 0.2 \end{aligned}$$

mit (19) nach Theorem 2 erfüllt.

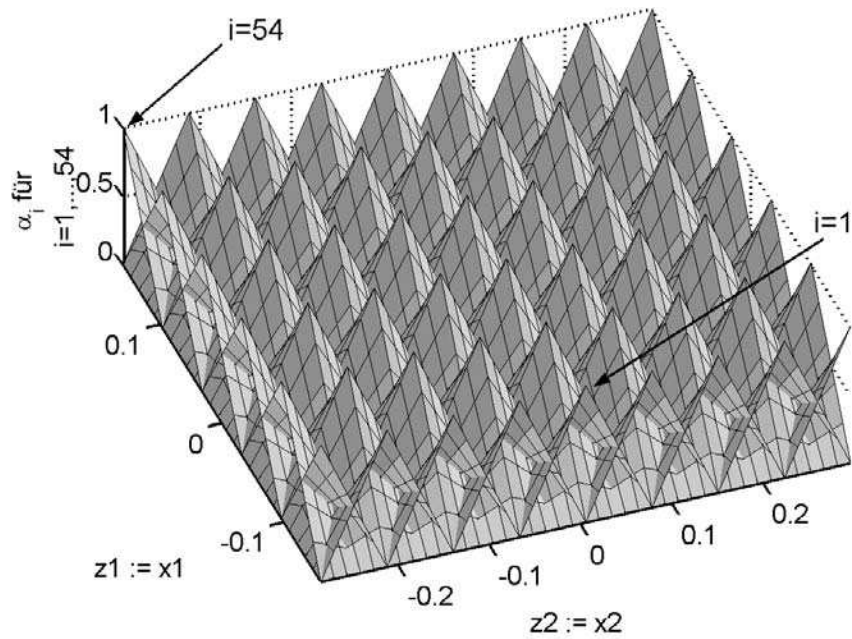


Bild 1: Gewichtsfunktionen $\alpha_i(x_1, x_2)$ der 54 lokalen Modelle

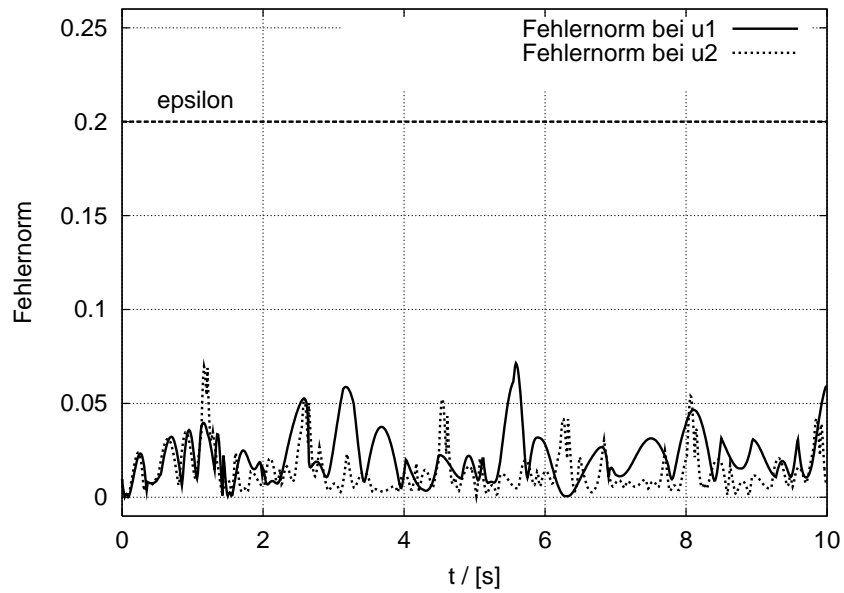


Bild 2: Darstellung des Approximationsfehlers $\|\mathbf{f}(\mathbf{x}, u) - \mathbf{f}_{TS}(\mathbf{x}, u)\|_2$ berechnet für die Eingänge $u_1(t)$ und $u_2(t)$, vgl. (39)

4 Untersuchungen zum dynamischen Fehlerwachstum

4.1 Fehlergleichung

Zur Verdeutlichung des dynamischen Fehlerwachstums wird

$$\dot{e}(t) = \dot{\mathbf{x}}(t) - \dot{\tilde{\boldsymbol{\xi}}}(t) \quad (40)$$

angesetzt, wobei \mathbf{x} die zeitliche Entwicklung des Zustandsvektors vom Originalsystem (6) und $\tilde{\boldsymbol{\xi}}$ die zeitliche Entwicklung des Zustandsvektors vom TS Fuzzy System (14c) repräsentiert. Nach dem Einsetzen und der Integration über eine feste Zeit T erhalten wir unter Ausnutzung der Identität $\mathbf{f}(\mathbf{x}, \mathbf{u}) = \sum_{i=1}^{N_r} \alpha_i(\mathbf{x}, \mathbf{u}) \mathbf{f}_i(\mathbf{x}, \mathbf{u})$ die Beziehung

$$e(t) = \int_0^T \sum_{i=1}^{N_r} \alpha_i(\mathbf{x}, \mathbf{u}) \mathbf{r}_{f_i}(\mathbf{x}, \mathbf{u}, \Delta\mathbf{x}, \Delta\mathbf{u}) dt + e(t_0). \quad (41)$$

Es ist $e(t_0) = \mathbf{x}(t_0) - \tilde{\boldsymbol{\xi}}(t_0)$, da im Allgemeinen der Initialisierungspunkt nicht mit einem Entwicklungspunkt zusammenfällt. Die Restterme \mathbf{r}_{f_i} können abgeschätzt werden mit

$$\|\mathbf{r}_{f_i}(\mathbf{x}, \mathbf{u}, \Delta\mathbf{x}, \Delta\mathbf{u})\|_{\infty} \leq \frac{1}{2} M_{f_i} \left\| \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{u} \end{bmatrix} \right\|_{\infty}^2, \quad (42a)$$

wobei

$$M_{f_i} = \max_{1 \leq j \leq n} \sum_{k_1, k_2=1}^n \sum_{k_3=1}^m \sup_{t \in [t_0, \infty)} \left| \frac{\partial^2 f_j(\mathbf{x}, \mathbf{u})}{\partial x_{k_1} \partial x_{k_2}} + \frac{\partial^2 f_j(\mathbf{x}, \mathbf{u})}{\partial^2 u_{k_3}} + \frac{\partial^2 f_j(\mathbf{x}, \mathbf{u})}{\partial x_{k_1} \partial u_{k_3}} \right| \quad (42b)$$

ist. Falls die Lösungen in einem beschränkten Gebiet liegen, dann wird der Fehlerverlauf ebenfalls beschränkt sein. Aufgrund der lokalen Gewichtung der Restterme ist der Beitrag jedes i 'ten Modells am aktuellen Gesamtfehler $e(t)$ auf das Gebiet um den i 'ten Entwicklungspunkt beschränkt und aufgrund von (42) auch begrenzt. Die wirksame Gebietsgröße ist mit $R = \varepsilon/L$ nach (23) festgelegt.

4.2 Numerische Untersuchungen am Fallbeispiel

Zur Verdeutlichung der bisher gemachten Aussagen wird nun am Beispiel des abgeleiteten TS Fuzzy Systems aus Abschnitt 3 die Fehlergleichung für eine Sprungantwort numerisch gelöst. Die Ergebnisse sind in Bild 3 für die Fehlerkomponente e_2 dokumentiert. Im oberen Diagramm sind die zeitlichen Verläufe von vier zusammenhängenden Gewichtsfunktionen dargestellt. Das untere Diagramm stellt die zeitliche Fehlerentwicklung von $e_2 = x_2 - \xi_2$ dar. Man erkennt deutlich eine Minimierung des Fehlers in der Nähe eines dominanten lokalen Modells. In diesem Fall vom 17'ten Modell. Der Fehler steigt an und erreicht Maxima in Bereichen, in denen zwei dominante lokale Modelle beteiligt sind, siehe die Bereiche um $t = 0.9$ s und $t = 2.6$ s.

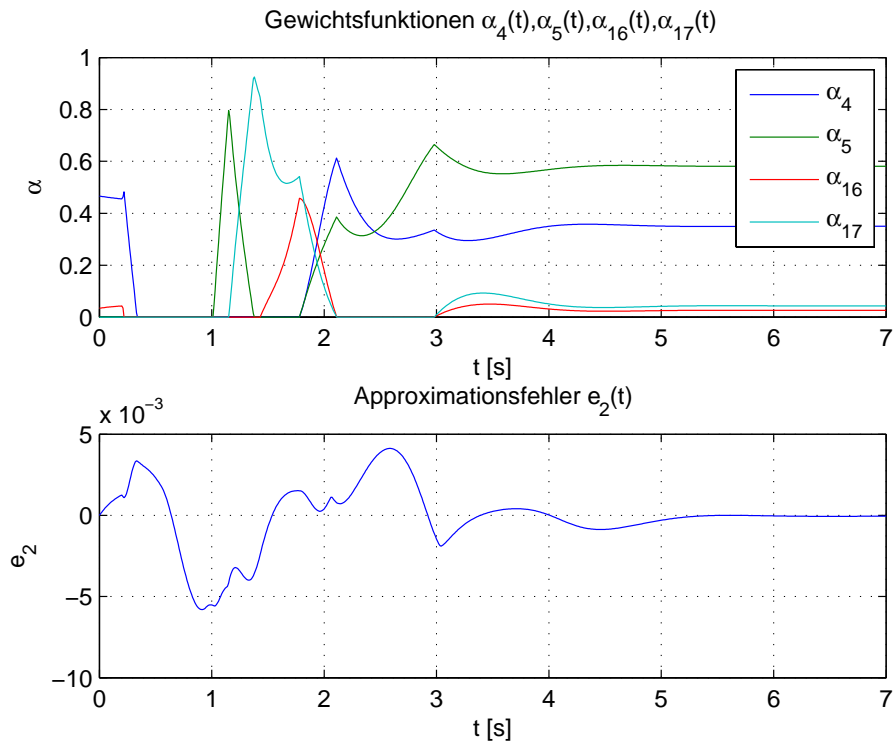


Bild 3: Zusammenhang zwischen dem zeitlichen Verlauf der Gewichtsfunktionen $\alpha_{4,5,16,17}$ und der Fehlerdynamik

4.3 Ansatz zur Reduktion der Modellanzahl

Aus der abgeleiteten Beziehung (25) folgt, dass mit steigender Anforderung an die Approximationsgenauigkeit, mit der Anzahl der Dimension des Vektors ψ_N und dem Wert des maximalen Eigenwerts der Hesseschen Matrix die Anzahl der lokalen Modelle steigt. Ebenso variieren bei steigendem Grad der Nichtlinearitäten in der Originalfunktion (6a) die lokalen Steigungen über den gesamten Arbeitsbereich stärker. Aufgrund der Worst-Case Abschätzung (vgl. die Anmerkung zur Beweisführung des Theorems 1) ist L die obere Schranke der größten lokalen Steigung. Die Radien der Hyperkugeln ergeben sich mit dieser oberen Schranke und dem vorgegebenen Fehler aus $R = \varepsilon/L$. Die Hyperkugeln legen den Gültigkeitsbereich jedes lokalen Modells fest und überdecken aufgrund identischer Radien R das zu approximierende Gebiet gleichmäßig. Wie man anhand der Darstellung des Fehlers in Bild 2 erkennen kann, ist diese Art der Approximation nicht besonders effektiv.

Aus diesem Grund wird zur Zeit ein Ansatz verfolgt, mit dem halbanalytisch mit Hilfe der gegebenen Originalfunktion basierend auf den Eigenwerten der Hesseschen Matrix Bereiche zusammengefasst werden. Damit wird in lokalen Bereichen die Hyperkugelanzahl verringert bzw. die Kugelradien vergrößert, wobei insgesamt die Anzahl der lokalen Modelle verringert wird.

5 Zusammenfassung

In dieser Arbeit wurde die Approximationseigenschaft von affinen Takagi-Sugeno Fuzzy Systemen analysiert. Dabei wurde eine Methode zur Konstruktion von TS Fuzzy Systemen, bei der die notwendige Modellanzahl explizit bestimmt wird, untersucht.

Die Anwendbarkeit wurde anhand eines nichtlinearen Feder-Dämpfer-Masse Systems geprüft. Hierfür wurde eine TS Struktur mit 54 lokalen Modellen abgeleitet und die Dynamik mittels numerischer Integration mit dem Originalsystem verglichen. Zur Reduktion der Modellkomplexität wurde das zu approximierende System in einen linearen und nichtlinearen Anteil zerlegt. Anschließend wurde eine Fehlergleichung zur Beschreibung des dynamischen Fehlerwachstums abgeleitet. Anhand des zuvor behandelten Fallbeispiels wurde diese Gleichung exemplarisch gelöst.

Zum Schluss wurde ein halbanalytischer Ansatz zur Reduktion der Modellanzahl unter Ausnutzung einer nicht-äquidistanten Verteilung der lokalen Gewichtsfunktionen in den Grundzügen vorgestellt.

Literatur

- [1] G. Z. Angelis, *System Analysis, Modelling and Control with Polytopic Linear Models*, PhD thesis, University Eindhoven, Niederlande, Feb. 2001.
- [2] H. W. Knobloch and F. Knappel, *Gewöhnliche Differentialgleichungen*, B.G. Teubner, Stuttgart, 1974.
- [3] R. Rovatti, Takagi-Sugeno models as approximators in Sobolev norms: the SISO case, in *Fifth IEEE International Conference on Fuzzy Systems*, New Orleans, USA, 1996, pp. 1060–1066.
- [4] H. Schulte, *Approximative Modellierung, Systemidentifikation und Reglerentwurf mittels gewichteter Kombination lokaler Zustandsraummodelle am Beispiel fluidischer Antriebe*, PhD thesis, Universität Kassel, 2005.
- [5] T. Takagi and M. Sugeno, Fuzzy identification of systems and its application to modelling in control, *IEEE Transactions on System, Man and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.

Kognitive Architekturen für humanoide Roboter: Anforderungen, Überblick und Vergleich

Catherina Burghart¹, Ralf Mikut², Tamim Asfour¹, Andreas Schmid¹, Florian Kraft¹, Oliver Schrempf¹, Hartwig Holzapfel¹, Rainer Stiefelhagen¹, Alexej Swerdlow¹, Georg Bretthauer^{1,2}, Rüdiger Dillmann¹

¹Universität Karlsruhe (TH), Sonderforschungsbereich 588

²Forschungszentrum Karlsruhe GmbH, Institut für Angewandte Informatik

1 Einleitung und Historie

An humanoide Roboter werden hohe Ansprüche gestellt: Sie sollen über eine menschenähnliche Bewegung, eine menschenähnliche Kommunikation und eine menschenähnliche Intelligenz verfügen. Vor allem die beiden letzten Eigenschaften sind Ausdruck dafür, dass sich ein Mensch einen intuitiven Umgang mit dem Roboter wünscht und typische menschliche kognitive Fähigkeiten in ihn hinein projiziert.

In der Psychologie umfasst die Definition des Begriffs Kognition sowohl bewusste bzw. intentionale als auch unbewusste Prozesse; beispielsweise das aktive Lösen eines Problems als bewusster Vorgang im Gehirn oder das Fahrradfahren als unbewusste Fähigkeit. Des Weiteren werden die kognitiven Funktionen in höhere (wie das Erkennen) und niedrigere Funktionen (beispielsweise motorische Basisfunktionen) eingeteilt. Zur Kognition gehören Funktionen der Wahrnehmung (Perzeption), des Gedächtnisses, des Denkens und Problemlösens, des Lernens, der motorischen Steuerung und des Sprachgebrauchs. Dieser Kognitionsbegriff aus der Psychologie [1] kann gleichermaßen auch auf kognitive Robotersysteme angewandt werden.

Kognitive Architekturen sind Theorien über die Funktionsweise der menschlichen oder tierischen Informationsverarbeitung, wobei versucht wird, das konkrete Verhalten in einer speziellen Situation aus grundlegenden Annahmen abzuleiten. Psychologische Experimente liefern dabei den Grundstein für diese Annahmen. Mithilfe von kognitiven Architekturen lassen sich Theorien über die natürliche Kognition implementieren und testen sowie kontextunabhängige Mechanismen der Intelligenz erforschen. Eine geeignet implementierte kognitive Architektur ermöglicht außerdem ein intelligentes Verhalten eines künstlichen Agenten (z.B. eines intelligenten Roboters) in seiner Umwelt.

Die Arbeiten zu kognitiven Architekturen haben ihren Ursprung in unterschiedlichen Disziplinen, z.B. der kognitiven Psychologie und der Robotik. Ein Vergleich der Architekturen ist relativ schwierig: Erstens sind sie für verschiedene Bedürfnisse entworfen worden, z.B. um menschliches Verhalten zu simulieren oder um als kognitives Rahmenwerk für ein intelligentes künstliches System zu dienen. Zweitens verfolgen sie unterschiedliche Konzepte, beispielsweise eine Multiagentenarchitektur, eine streng hierarchische Architektur oder mehrere verteilte funktionspezifische Subsysteme. Drittens sind nicht alle Fähigkeiten einer kognitiven Architektur im gleichen Detaillierungsgrad implementiert, z.B. in der Bildverarbeitung, Spracherkennung oder Dateninterpretation.

Es gibt schon eine Reihe an Artikeln, die einen Überblick über kognitive Architekturen geben, beispielsweise von Byrne oder Langley [2, 3]. Vor dem Hintergrund der Anforder-

rungen an einen humanoiden Roboter wird in diesem Artikel ein Überblick über kognitive Architekturen gegeben, die aus verschiedenen Feldern wie der kognitiven Psychologie, der Servicerobotik oder der humanoiden Robotik stammen. Dies soll dazu beitragen, geeignete Bestandteile verschiedener Architekturen zu identifizieren, um diese in einer kognitiven Architektur zu integrieren.

Die ersten Architekturen für intelligente Robotersysteme brachte das Forschungsgebiet der Künstlichen Intelligenz hervor (z.B. [4, 5]). Diese Architekturen verwendeten Sensordaten, um ein vollständiges, symbolisches Abbild der Umwelt des Robotersystems zu kreieren und Handlungen auf dieser symbolischen Ebene zu planen und auszuführen, sogenannte Sense-Plan-Act (SPA) Strategien. Der Zyklus wird vom Robotersystem fortlaufend wiederholt. Ein wesentlicher Nachteil dieser Strategie ist die lange Reaktionszeit und die stark vereinfachte Abbildung der realen Welt im rechnerinternen Modell. Über den mobilen Roboter Stanford CART sagte Moravec [4], dass dieses System zwar verlässlich liefere, aber sehr langsam sei, denn alle zehn bis fünfzehn Minuten bewege es sich plötzlich einen Meter vorwärts. Diese Art von Robotersystemen war mehr oder weniger dazu bestimmt in einfachen, künstlichen Welten zu leben.

Brooks entwarf eine Architektur, die sich mehr an den Fähigkeiten des Robotersystems orientierte und unter dem Namen subsumierende Architektur bekannt ist [6]. Sie besteht aus verteilten, parallel interagierenden Komponenten, die verschiedene funktionelle Verhaltensweisen repräsentieren, beispielsweise die Motorik oder die Perzeption. Dieses Prinzip wurde von vielen Forschungsgruppen angewandt, beispielsweise auch von [7–9]. Noch nicht gelöste Probleme dieser Architektur sind die Repräsentation von Zielen und die Koordination der Komponenten.

Die 3-Ebenen-Architektur entstand aus der Verschmelzung von SPA- und subsumierender Architektur [10–12]. In dieser Architektur beinhaltet nur die unterste Ebene die Verhaltensfunktionen, wohingegen die mittlere Ebene die Komponenten der untersten Ebene koordiniert. Auf der obersten Ebene befindet sich schließlich eine Planungskomponente. Hackel [13] hat dieses Konzept auch noch weiter ausgearbeitet. Die 3-Ebenen-Architektur deckt sich gleichzeitig auch mit dem Modell des menschlichen Operators aus den Geisteswissenschaften. So teilt Rasmussen [14] die Informationsverarbeitung eines Menschen in eine fähigkeitsbasierte, eine regelbasierte und eine planbasierte Ebene ein. In den letzten Jahren kam noch die Bestrebung auf, Emotionen als Motivationssystem in die Architektur zu integrieren [15–17].

Der vorliegende Artikel untersucht zunächst die Anforderungen an ein humanoides Robotersystem (Abschnitt 2). Anschließend werden verschiedene kognitive Architekturen, die aus der kognitiven Psychologie stammen (Abschnitt 3) oder die in intelligenten mobilen Systemen (Abschnitt 4) bzw. speziell in humanoiden Robotersystemen (Abschnitt 5) zu finden sind, vorgestellt. Zum Abschluss werden die verschiedenen Architekturen miteinander verglichen (Abschnitt 6).

2 Allgemeine Anforderungen für humanoide Roboter

Mit dem Begriff *humanoider Roboter* werden im allgemeinen die Fähigkeiten eines Roboters zur menschenähnlichen Bewegung, Kommunikation und Intelligenz in Verbindung gebracht. Ein mit dem Roboter interagierender Mensch kann dessen Bewegungen intuitiv besser abschätzen und wird seltener durch unvorhergesehene Aktionen des Roboters

überrascht. Es gibt schon mechatronisch sehr ausgefeilte humanoide Roboter, beispielsweise Qrio (Sony), Asimo (Honda), blasende Trompeter (Toyota) oder Wendy (Waseda University). Sie können tanzen, hüpfen, Treppen steigen oder auch in der Küche rohe Eier zum Kochen aufschlagen, verfügen aber alle über keine höheren kognitiven Funktionen. Die Bewegung des Roboters per se und die Koordination seiner Vielzahl von notwendigen Aktoren soll nicht weiter Gegenstand dieses Artikels sein. Der Fokus liegt hier auf einer Architektur, die dem Roboter eine angemessene, vom Menschen intendierte Kognition und damit auch Interaktion mit der Umwelt ermöglicht.

Die kognitive Architektur eines intelligenten Robotersystems muss einerseits eine schnelle Wahrnehmung, Regelung und Aufgabenausführung ermöglichen. Andererseits muss sie ebenso höhere kognitive Funktionen unterstützen, wie beispielsweise das Erkennen und die Interpretation komplexer Zusammenhänge, das Planen von Aufgaben und das Lernen neuer Verhaltensweisen. Kognitive Funktionen der höheren Ebenen korrelieren dabei mit einer höheren Komplexität des Robotersystems und mit der Fähigkeit des Verstehens komplexer Zusammenhänge.

Ein solches Robotersystem muss die Sicherheit des Menschen garantieren, ansprechend aussehen und auch für den unbedarften Anwender einfach zu handhaben sein. Die eigentliche Herausforderung an die kognitiven Fähigkeiten des Roboters beginnt erst mit einem höheren erwarteten Grad an Intelligenz und Autonomie [18]. Das setzt die Beherrschung technischer Anforderungen wie die Objekterkennung unter verschiedenen Lichtverhältnissen, schnell wechselnden Szenen etc. voraus. Ebenso ist eine robuste Lokalisierung, Navigation, Bahnplanung, Sensorfusion sowie die Manipulation von deformierbaren Objekten (z.B. mit Flüssigkeiten) und in schwierigen geometrischen Umgebungen notwendig. Der Roboter muss seine eigenen Ressourcenbeschränkungen beachten und mehrere Handlungen parallel ausführen können.

Ein intelligentes Robotersystem sollte in der Lage sein, in natürlicher Sprache mit Laien zu reden, wobei der Wortschatz noch auf eine Domäne beschränkt sein kann. Dazu benötigt der Roboter eine aktive Auflösung von Mehrdeutigkeiten und Klärungsdialoge, um Missverständnisse, fehlende Worte, neue Objekte oder neue semantische Konstruktionen zu erfassen. Neben der sprachlichen Modalität spielt die nonverbale Kommunikation, die sich durch Gesten, Mimik und Körperhaltung ausdrückt, bei der zwischenmenschlichen Interaktion eine wichtige Rolle. Bei der Analyse und Interpretation von Situationen und der Planung von Handlungen muss der Roboter notwendige Informationen gezielt einholen und den Kontext berücksichtigen. Der Kontext umfasst u.a. interagierende Personen, gegenwärtige und erwartete zukünftige Situationen sowie Auswirkungen für den Nutzer. Das Ziel besteht im Herbeiführen positiver Situationen, wobei das Feedback des Menschen zu beobachten oder gezielt einzuholen ist.

In diesem Zusammenhang ist das Lernen neuer Aufgaben in einer unstrukturierten oder unbekanntem Umgebung relevant: Der Roboter sollte dies aus Vorführungen des Menschen lernen und unter Berücksichtigung seiner roboterspezifischen Eigenschaften umsetzen. Dies gilt nicht nur für Bewegungen, sondern auch für Problemlösungsstrategien, die mithilfe eines Tutors und aus eigenen Erfahrungen mit Optimierungsverfahren erworben werden.

Eine weitere wichtige Anforderung, die sowohl mit dem Erkennen von Situationen als auch dem Ausführen von geplanten Handlungen einhergeht, ist die gezielte Anpassung des Aufmerksamkeitsfokus für jede Sinnesart (z.B. optisch, akustisch). Während der Ro-

boter ein Objekt manipuliert, sollte er trotzdem wahrnehmen können, was der Mensch zu ihm sagt. Zur Steuerung des Aufmerksamkeitsfokus sind aber auch klare Regeln zur Bewertung einer Situation erforderlich, die festlegen, wann der Aufmerksamkeitsfokus verändert werden darf.

Bei der Interaktion eines humanoiden Roboters in der menschlichen Umwelt wird von diesem erwartet, dass er soziale Regeln kennt, erkennt und einhält. Dazu gehört neben der Proxemik auch das Erkennen nonverbaler Signale des Menschen, was wichtig für die Anpassung des Roboterverhaltens an die Situation ist, und das Geben nonverbaler Signale im Situationskontext. Die sozialen Regeln können auch noch weitergesponnen werden zu Fragen, die sich im Hinblick auf ein am öffentlichen Leben teilnehmendes Robotersystem in Zukunft stellen [19]: Wie steht es beispielsweise mit der Wahrung der Privatsphäre, der moralischen Verantwortung, intrinsischen moralischen Werten oder Gegenseitigkeit?

Ein Beispiel für Anforderungen ist die Anwendungsdomäne des humanoiden Roboters, der im Rahmen des Sonderforschungsbereichs 588 *Humanoider Roboter – Lernende und kooperierende multimodale Roboter* an der Universität Karlsruhe gebaut wird [20]. Die Schlüsselfragen, die in diesem Projekt bearbeitet werden, sind:

- die Kommunikation über multimodale Kanäle, um eine intuitive Interaktion des Benutzers mit dem Robotersystem zu ermöglichen,
- die Kooperation zwischen Mensch und Roboter in unterschiedlichem Kontext,
- die Fähigkeit des Roboters, ihm unbekannte Aufgaben, Bewegungen, Konzepte und Objekte zu erlernen, und
- die Fähigkeit des Roboters sich ähnlich einem Menschen zu bewegen.



Bild 1: Das im SFB 588 gebaute Robotersystem ARMAR III

Als Experimentierumgebung dient eine handelsübliche Einbauküche, in der auch der Roboter Schranktüren und Schubladen öffnen sowie die vorhandenen Geräte bedienen kann.

Der Roboter besteht aus einer mobilen Plattform mit einem beweglichen Oberkörper, zwei Armen mit jeweils einer 5-Finger-Hand, einem Kopf mit visuellen und akustischen Sensoren und ist mit einer künstlichen sensiblen Haut ausgestattet (Bild 1, siehe auch [21]).

Auch scheinbar einfache Kommunikationsexperimente in diesem SFB zeigen bereits die notwendige kognitive Komplexität des Roboters. Bei einem solchen Experiment bekam ein Proband die Aufgabe, ein Paket abzugeben, auf dem ein Name stand. Die weiteren notwendigen Informationen erhielt der Proband durch Konversation mit einem Roboter an einer Rezeption ([22], Szenario siehe Bild 2). Der Versuch wurde mit 16 Probanden an je drei aufeinanderfolgenden Tagen wiederholt, wobei die Probanden beim ersten Versuch den Roboter nicht kannten. Der verwendete Roboter bestand aus einem auf einem Rahmen montierten Halsgelenk mit Stereokamerakopf und verfügte über die gleichen Methoden und Komponenten zur Spracherkennung, zum Dialogmanagement und zur Gesichtserkennung wie ARMAR III.



Bild 2: Ein Proband versucht von einem unbekanntem Roboter Informationen zu erlangen

Dabei zeigte sich, dass sich die Art und Weise der Kommunikation zwischen Proband und Roboter kaum vorhersagen lässt. Der Proband wird schnell frustriert und kann sein Ziel nur umständlich oder gar nicht erreichen. Er projiziert viele Fähigkeiten in das Robotersystem hinein, die dieses gar nicht besitzt.

Als Fazit des Versuches ist zu fordern, dass ein mit einer Person interagierendes Robotersystem immer eine Anschlussfähigkeit gewährleistet. Die Person ist so zu lenken, dass immer eine möglichst eindeutige Aktion möglich ist. Dazu muss der Roboter über einen hinreichend großen Sprachschatz verfügen und verschiedene Strategien zum Informationserwerb sowie zur Informationsübermittlung unterstützen. Der Roboter soll sich mit seinen Aktionen an sein Gegenüber anpassen, um Schleifen, Brüche, Strategiewechsel, Frustrationen oder Ärger zu bemerken, damit er sein Verhalten daran anpassen kann.

3 Architekturen aus der kognitiven Psychologie

Die kognitive Psychologie hat eine Reihe von kognitiven Architekturen hervorgebracht, die einerseits dazu dienen, Verhaltensweisen des Menschen zu modellieren, andererseits um psychologische Experimente und Theorien zu evaluieren und auch am Computer zu simulieren. Die bekanntesten dieser Architekturen werden im Folgenden vorgestellt.

Die **SOAR-Architektur** (State Operator Apply Result, [23]) erfasst die aktuelle Situation über hierarchisch gegliederte Problemräume mit aktuellen Zuständen und je einem aktuellen Ziel. Das Langzeitgedächtnis enthält symbolische Regeln, aber auch Relationen zwischen Objekten und aufgezeichnete Episoden. Das Kurzzeitgedächtnis lädt relevante Regeln aus dem Langzeitgedächtnis und umfasst alle Wahrnehmungen in Form von Variablen. Bei einer Entscheidung werden solange alle zutreffenden Regeln aus dem Langzeitgedächtnis geladen und abgearbeitet, bis keine neue Regel mehr feuert (Elaboration Phase). Wenn dann eine eindeutige Entscheidung über eine Operation möglich ist, wird diese ausgeführt. Anderenfalls (keine oder widersprüchliche Operation) wird ein neuer untergeordneter Problemraum mit einem unterlagerten Zwischenziel angelegt. Der führt eine eindeutige Entscheidung herbei, um danach im nächsthöheren Problemraum fortsetzen zu können. Hierzu sind spezielle Lernmechanismen (Chunking) implementiert, die sich auf aufgezeichnete Episoden oder zufällige Operationen stützen. Der Abarbeitungsmechanismus ist domänenunabhängig, das enthaltene Wissen ist domänenabhängig. Ein Anwendungsbeispiel ist die Modellierung von menschlichen Piloten in Flugsimulatoren.

Die von J. R. Anderson und C. Lebiere 1998 entwickelte **ACT-R Theorie** (Adaptive Control of Thought - Rational) [24] fußt auf zwei Grundannahmen: Die eine Theorie bescheinigt der neuronalen Struktur des Gehirns einen entscheidenden Einfluss auf die Verarbeitung von Informationen. Die andere nimmt an, dass die Vorgehensweise des Menschen beim Lösen von Problemen rational ist. Diese zwei Theorien spiegeln sich im Aufbau von ACT-R wider, das zum Einen das Zusammenwirken aller Komponenten und die Wissensverarbeitung, zum Anderen die wahrnehmenden, handelnden und ausführenden Mechanismen beschreibt. Laut ACT-R spielt sich die menschliche Wahrnehmung auf einer symbolischen und einer subsymbolischen Ebene ab. Auf der symbolischen Ebene gibt es Wissen in Form von Faktenwissen zu Sachverhalten und in Form von Produktionswissen zur Beschreibung von Abläufen. Auf der subsymbolischen Ebene entwickelt sich das Wissen erst dadurch, dass es sich an die statistischen Eigenschaften der Umwelt anpasst, indem beispielsweise durch erfolgreiches Probieren gelernt wird. ACT-R unterscheidet zwischen deklarativem und prozeduralem Wissen [25, 26]. Das deklarative Wissen beschreibt Fakten, beispielsweise die Eigenschaften eines Objekts, zu denen das Gedächtnis einen bewussten Zugang hat. ACT-R stellt deklaratives Wissen in Form von *Chunks* dar. Prozedurales Wissen bildet hingegen Vorgänge ab, die auf Sachverhalte wirken und dessen sich der Mensch nicht bewusst ist. Typische Beispiele hierfür sind das Greifen eines Objektes oder auch das Lenken der Aufmerksamkeit auf ein Objekt. ACT-R verwendet *Produktionen*, um prozedurales Wissen darzustellen. Eine Produktion besteht aus einem Bedingungsteil und einem Aktionsteil. Der Bedingungsteil enthält und überprüft die Zustände, in denen sich das System vor der Anwendung der Produktion befinden muss. Der Aktionsteil beschreibt die Aktion, die bei Aufruf der Produktion ausgeführt wird.

Die Erfinder des **LICAI-Modells** (LInked model of Comprehension-based Action planning and Instruction taking) [27] interessieren sich dafür, wie ein Benutzer eines Computers die verschiedenen Anwendungen, die er gleichzeitig benutzt, verwaltet und neues Wissen über die zugrundeliegende Betriebssystemstruktur erfährt. Das Modell beschreibt einen erfahrenen Computerbenutzer, der Ketten von Aktionen durchführt, um durch Erkunden Aufgaben zu lösen (beispielsweise eine bestimmte Funktion in einem Texteditor anzuwenden). So gefundene, neue Aktionsketten werden gespeichert. Die Aufgaben des Benutzers werden von LICAI erfasst, die dazugehörigen Ziele und Teilziele erstellt und in einem Zielspeicher abgelegt. Die Ziele steuern die einzelnen Aktionen des Benutzers.

Dabei geht LICAI davon aus, dass ein Anwender seine Aufgabe umso schneller löst, je besser das gesuchte Objekt und das verfolgte Ziel zusammen passen.

Eine Weiterentwicklung des LICAI-Modells ist **CoLiDeS** (Comprehension-based Linked model of Deliberate Search) [29], das dazu verwendet wird, die Suche eines Benutzers nach Informationen im Internet zu modellieren. Hierbei war die Frage, warum die Suche so oft in einer Sackgasse endet. Das Modell analysiert Webseiten in vier Schritten: Im ersten Schritt wird eine neue Seite aufgerufen und dann im zweiten Schritt auf funktionale Elemente wie Navigationsbereiche oder Suchmasken hin analysiert. Dann wird eines der funktionalen Elemente näher analysiert, mit all seinen Alternativen, die es als weitere Verknüpfungen anbietet. Schließlich wird diejenige Verknüpfung aus der Menge der Alternativen ausgesucht, deren Namen mit dem Ziel der Suche die größte Ähnlichkeit aufweist, und die dazugehörige Webseite aufgerufen. Ist die gewünschte Information gefunden, so besteht eine Aktionskette aus einer Abfolge von Webseiten. Mithilfe der Modellierung ist es möglich, Erfolgsfaktoren und Gründe für das Fehlschlagen einer Suche zu finden.

Da die Evaluation von Mensch-Maschine-Schnittstellen mit menschlichen Probanden viel Zeit in Anspruch nimmt, wurde die kognitive Architektur **EPIC** (Executive Process Interactive Control) entworfen [30, 31], um das Verhalten der Testpersonen zu simulieren und so die Benutzerschnittstellen auf ihre Tauglichkeit zu testen. Die Stärke von EPIC ist das Modellieren der menschlichen Leistungsfähigkeit in einer Vielzahl verschiedener Szenarien mit Mensch-Maschine-Interaktion. Es können sowohl einfache Interaktionen (z.B. Bedienfeldeingaben) als auch komplexe Steuerungsaufgaben, die verschiedene Modalitäten einschließen (z.B. parallele Bedienung mehrerer Eingabegeräte), modelliert werden. EPIC besitzt Modelle zur Wahrnehmung, Wissensverarbeitung, Motorik und zur Umweltmodellierung. Letzteres bietet die Möglichkeit, ein logisches und physikalisches Modell des Zielsystems, beispielsweise ein Flugzeug und die Bedienschnittstelle mit all ihren Ein- und Ausgabemöglichkeiten, zu erstellen. Eine Lernkomponente ist nicht Bestandteil von EPIC. Die menschliche Wahrnehmung und die Motorik können sehr detailliert abgebildet werden: Ein Ohrenmodell simuliert die menschliche Wahrnehmung von Geräuschen, ein Augenmodell die visuelle Wahrnehmung und die Steuerung des visuellen Aufmerksamkeitsfokus, unterstützt vom Augenmotorikmodell. Weitere Modelle simulieren die Spracherzeugung und die Handbewegungen. Jedes Modellmodul legt seine Zustände und Sensordaten im Arbeitsspeicher von EPIC ab, so dass diese der wissensverarbeitenden Komponente als Faktenwissen zur Verfügung stehen. Eine Besonderheit der EPIC-Architektur besteht darin, dass keine direkte Modellierung der tatsächlichen Wahrnehmungs- und Ausführungsvorgänge stattfindet, vielmehr werden stattdessen die entsprechenden zeitlichen Informationsflüsse für jedes einzelne Modul vorhergesagt. Diese Vorgehensweise kann in unterschiedlich langen Wegen zwischen den einzelnen Modulen und dem Arbeitsspeicher für verschiedene Informationsarten resultieren.

Eine allgemeine kognitive Lernstruktur für organismische Lernprozesse beim Menschen beschreibt **Klix** [33] aus evolutionspsychologischer Sicht. In [34] wird aufgezeigt, dass sich Selektionsdruck in der Evolution auf die Wechselwirkung von Verhaltensentscheidungen mit der homöostatischen Regulation auswirkt. Damit werden Wahrnehmungen und daraus resultierende Verhaltensantworten in den Dienst der Reduktion von Bedürfniszuständen gestellt, wodurch eine phylogenetische Basis für die Entstehung von Verhaltensmotivationen resultiert. Aktionen können so durch deren Rückwirkung auf die Veränderung der homöostatischen Lage bewertet werden. Durch die anpassungsfähige Gedächtnisstruktur des Menschen können zuvor positiv bewertete Verhaltenskorrekturen re-

aktiviert werden. Dabei werden gegenwärtige situationsabhängige Wahrnehmungsmerkmale in der Gedächtnisstruktur gesucht. Hingegen kann auch das Ungleichgewicht der homöostatischen Lage einen Suchprozess auslösen, welcher für Unruhe im Organismus sorgt, bis gegenwärtig angezeigte Umgebungsmerkmale mit einer als bedürfnisreduzierend eingestuften Assoziation von Merkmalen und Aktionen übereinstimmen. Aus der funktionalen Betrachtung des Zusammenspiels der genannten Komponenten entstand eine allgemeine Lernstruktur [33]. Diese kognitive Lernstruktur wird mit chronologisch aufeinander aufbauenden Lerntypen instantiiert. Neben mathematischen Definitionen der Lerntypen belegen biologisch motivierte Beispiele die Funktionalität der Lernstrukturen für Lernen durch Abgewöhnen (Auflösung einer Verhaltensbindung an Merkmale), durch bedingte Reaktionen (Entstehung einer Verhaltensbindung), durch instrumentales Bedingen (Abbildung von realisierbaren Zusammenhängen) und durch Einsicht (Konstruktion von möglichen Verkettungen). Diese Lernprozesse bilden die Grundlage für komplexere kognitive Lernformen, die [33] in Kapiteln zu Klassifizierungsprozessen und Begriffsbildungen, Problemlösungsprozessen und produktivem Denken diskutiert.

Die **PSI-Architektur** [35] basiert auf einer psychologischen Theorie zur Erklärung von menschlichen Handlungen in komplexen Situationen unter besonderer Berücksichtigung von Bedürfnissen, Absichten, Motiven und Emotionen. Das Konzept wurde anhand umfangreicher Computersimulationen für künstliche Agenten in einer künstlichen Umwelt erprobt. Die Modellierung des Agenten erfolgt vollständig über speziell strukturierte hierarchische und rekurrente neuronale Netze. Die Stärke von Bedürfnissen und die erwartete Kompetenz des Agenten für eine erfolgreiche Lösung bestimmen die Stärke eines Motivs. Das gegenwärtig stärkste Motiv veranlasst die Auswahl einer Handlung und hemmt konkurrierende Motive. Dieser Auswahlprozess wird über Emotionen moduliert, um u.a. situationsgerechte Reaktionszeiten, Energieaufwendungen und Planungstiefen zu wählen. Das integrierte Lernen erfolgt über ein modifiziertes Hebbsches Lernen, das u.a. auf einer Erfolgs- und Misserfolgsbewertung aufgrund befriedigter oder nicht befriedigter Bedürfnisse und auf zeitlichen Korrelationen basiert.

Das **CogAff-Schema** von Sloman ist ein Rahmenwerk zum Vergleich unterschiedlicher Architekturkonzepte [36, 37]. Es entspricht einer Grobgliederung der informationsverarbeitenden Komponenten, von denen je nach Entwicklungsgrad der zu vergleichenden Architekturkonzepte alle, oder aber auch nur Teile davon vorhanden sein können. Es erlaubt zudem die Beschreibung ihrer internen Zusammenhänge unter Verwendung eines einheitlichen Vokabulars. Durch die Abstraktion von Implementierungsdetails können somit unterschiedliche Architekturansätze durch die Beschreibung über das CogAff-Schema einfacher verglichen werden. Grundsätzlich geht das CogAff-Schema aus der Kombination des 3-Säulen-Modells mit dem 3-Ebenen-Modell hervor. Das 3-Säulen-Modell beschreibt den Informationsfluss innerhalb eines Systems von den Sensoren (Perzeption) über eine zentrale Verarbeitung bis zu den Affektoren (Aktion). Das Ebenen-Modell trägt den unterschiedlichen Arten der Informationsverarbeitung Rechnung und trennt zwischen reaktiver, deliberativer und Metaebene. Eine konkrete Instanz dieses Konzepts, die alle wesentlichen Komponenten implementiert, ist die H-CogAff-Architektur.

4 Architekturen aus der Service-Robotik und für autonome Systeme

Intelligente Serviceroboter, die nicht zu den humanoiden Robotern zählen, benötigen eine adäquate Architektur zur Umsetzung ihrer kognitiven Fähigkeiten. Einige Beispielsarchitekturen werden im Folgenden kurz dargestellt.

ADAPT (Adaptive Dynamics and Active Perception for Thought) [38] ist eine kognitive Architektur, die speziell für Roboter entwickelt wird. ADAPT vereint die verschiedenen kognitiven Architekturen SOAR, ACT-R und EPIC zu einer einzigen Theorie. Damit soll der Roboter verschiedene Fähigkeiten, wie zum Beispiel den Gebrauch natürlicher Sprache und das Problemlösen, miteinander verbinden. Von SOAR stammt die generelle Architektur, das Problemlösen durch Suche im Problemraum und durch Zwischenziele sowie der Lernmechanismus Chunking. Von ACT-R wurde das deklarative Langzeitgedächtnis und der Bayes-Mechanismus zur Aktionsauswahl übernommen. Die EPIC-Architektur liefert das Modell für die menschenähnliche Wahrnehmung, die das parallele Feuern der Perzeptionsmodule sowie den Flaschenhals für Architekturziele beinhaltet. Entwickelt wird ADAPT von Benjamin, Lyons und Lonsdale. Benjamin und Lyons beschäftigen sich mit Problemstellungen der Robotik und Künstlichen Intelligenz, während der Linguist Lonsdale vor allem an der Komponente NL-Soar für die Sprachverarbeitung arbeitet. Bis dato gibt es allerdings keine bekannte, vollständige Implementierung dieser Architektur auf einem Roboter; publiziert wurde bisher lediglich die erfolgreiche Integration der Komponenten für die Sprachverarbeitung und die visuelle Perzeption in ein Gesamtsystem.

Im Europäischen Project **PACO-PLUS** [39] wird eine 3-Ebenen-Architektur vorgeschlagen, die auf neurophysiologischen und psychophysikalischen Erkenntnissen basiert. Diese gehen von der Existenz zweier komplementärer Langzeitgedächtnis-Systeme beim Menschen aus [40]: Ein schnell lernendes episodisches Gedächtnis und ein langsamer lernendes semantisches Gedächtnis, welches kontextinvariantes Wissen aus dem episodischen Gedächtnis extrahiert. Die unterste Ebene umfasst die sensomotorische Steuerung, die mittlere das semantische und episodische Gedächtnis. Die oberste Ebene ist für Schlussfolgerungen und die Aktionsplanung verantwortlich. Zusätzlich sieht die Architektur ein adaptives Top-Down-System vor, welches für die Koordination kognitiver Prozesse während der Ausführung von Handlungen verantwortlich ist.

Die Forschungsgruppe des **Cogniron**-Projektes [41] ist der Ansicht, dass in einer kognitiven Struktur eines intelligenten Robotersystems, das mit Menschen interagiert, vor allem die folgenden Fähigkeiten berücksichtigt werden müssen:

- die Interpretation der Semantik der Umwelt und des Kontextes,
- die Entscheidungsfindung,
- die Kommunikation und die Interaktion mit dem Menschen und
- das Lernen.

Die vorgeschlagene Cogniron-Architektur implementiert die oben genannten Eigenschaften und integriert konkurrierende Prozesse und interagierende Subsysteme, die unterschiedlichen zeitlichen Ausführungsbedingungen genügen. Eine zentrale Eigenschaft des Roboters muss sein, kontinuierlich zu lernen, denn nur so wird die Kognition des Systems nicht durch vorgegebene Wissensrepräsentationen und einer Menge möglicher Operationen oder Handlungen eingeschränkt. Die Sensoren und die Komponenten der unteren Ebene des Roboters liefern fortlaufend Daten, die ein fortwährender Prozess in bekannte und neue Daten einteilt. Aus letzteren werden neue Abbildungen von Objekten, des Raums und von Situationen erzeugt. Gleichzeitig muss die große Menge eingehender Daten auch gefiltert werden, um nur relevante Daten weiter zu verarbeiten. Neues Wissen soll der Roboter nicht allein durch die Wahrnehmung erwerben, sondern auch, indem er internes Wissen neu strukturiert, ableitet und assoziiert. Aus diesem Grund ist das Gedächtnis

für Faktenwissen nicht nur als passiver Wissensspeicher, sondern als aktive Komponente anzusehen, die über Mechanismen verfügt, um Daten neu zu strukturieren. Auch die Fähigkeiten, Regeln und Skripte (Beschreibungen, wie ein Verfahren ausgeführt werden muss) werden kontinuierlich verändert und verbessert, indem die erreichten Ergebnisse mit den zu erreichenden Zielen verglichen werden. Das benötigte Wissen ist über die ganze Architektur verteilt, da spezifisches Wissen mit spezifischen, funktionalen Prozessen verknüpft ist. Dazu benötigen die einzelnen Komponenten jeweils eigene Darstellungen eines Sachverhaltes. Nur der gesamte Prozess der Situationsinterpretation liefert auch ein zentrales Abbild einer Situation.

Die Kognitive Architektur von **BIRON** basiert auch auf einer 3-Ebenen-Architektur [42] mit einem reaktiven, rückgekoppelten Regelungsmechanismus, einem reaktiven Mechanismus zur Planausführung und einem Mechanismus zur Problemlösung unter Einbeziehung zusätzlichen Wissens und zur Handlungsplanung. Diese drei Komponenten laufen als getrennte Prozesse ab. Die wichtigste Komponente der Architektur ist der sogenannte *Sequencer*, der einerseits für die reaktive Ausführung eines erstellten Planes verantwortlich ist und andererseits die Vorgänge des Moduls zur Planung und zum Problemlösen überwacht. Deshalb ist der Sequencer auf der mittleren Ebene der Architektur angesiedelt. Für das Personenaufmerksamkeitssystem wird im Gegensatz zum Sequencer eine reaktive Regelung benötigt, die auf der untersten reaktiven Ebene liegt. Analog dazu existiert ein Aufmerksamkeitssystem für Objekte, dem zur einfacheren Auswahl eines Objektes auch die Gestenerkennung vorgelagert ist. Ein Szenenmodell, das sich auf der mittleren Ebene befindet, speichert Informationen über die an einer Interaktion beteiligten Objekte.

Das Aufmerksamkeitssystem wählt daraufhin die Aktionen aus, die die Robotersteuerung zu spezifischen Zeitpunkten verwenden soll, um die vom Sequencer angestoßene Aufgabe zu erfüllen. Das Aufmerksamkeitssystem, das sich sowohl auf Personen als auch auf Objekte richten kann, sowie die anderen Systemkomponenten der beschriebenen Architektur befähigen BIRON zur sozialen Interaktion durch Kommunikation mit einem Menschen. Die Robotersteuerung selbst ist Teil der ISR-Verhaltensweisen [12], die wiederum direkten Zugriff auf die Hardware des Roboters haben. Verschiedene Verhaltensweisen des Roboters stehen für Bewegungen und Navigationsfunktionen, die auf der direkten Kopplung von eingehenden Sensordaten mit den Aktoren basieren. Das Dialogkontrollsystem ist auf der obersten Ebene platziert, da es sowohl für die Erzeugung von Dialogen als auch für den Entwurf von Plänen zur Interaktion mit dem Menschen verantwortlich ist. Dafür liefern der Spracherkenner und das Szenenmodell des Roboters die benötigten Informationen, denn die Entscheidungen, die das Dialogkontrollsystem trifft, stehen im engen Zusammenhang mit den Anweisungen, die ein menschlicher Interaktionspartner dem Robotersystem gegeben hat. Wenn Mehrdeutigkeiten auftreten, werden diese durch einen Dialog mit dem Interaktionspartner gelöst. Der Dialogagent ist aus deterministischen Automaten aufgebaut, die jeweils definierte Teildialoge abbilden.

Die Forschungsgruppe Nagel beschäftigt sich seit vielen Jahre mit der Entwicklung eines kognitiven Bildverarbeitungssystems (**Cognitive Vision System**, CVS) [43]. Nagel stellt hier ein kognitives Bildverarbeitungssystem vor, welches durch Videobildauswertung von Verkehrsszenen natürlichsprachliche Beschreibungen dieser Szene erstellt. Die Architektur des CVS besteht aus mehreren Schichten, die in verschiedene Blöcke bzw. Subsysteme zusammengefasst werden können. Auf der unteren Ebene gibt es ein mehrschichtiges Kern-Bildverarbeitungssystem (Core Vision System), in welchem die Extraktion und das Verfolgen von Fahrzeugen bewerkstelligt wird. Als Ergebnis der höchsten

Schicht werden in der so genannten Szenen-Domänen-Schicht modellbasiert verschiedenen Fahrzeugtypen verfolgt und deren Orte, Lage und Trajektorien bereitgestellt. Über dem CVS gibt es dann eine Schicht, die für die konzeptuelle Repräsentation zuständig ist (Conceptual Representation Subsystem, CRS). Dort werden mithilfe von Domänenwissen zunächst bestimmte Fahrzeugmanöver durch Analyse der Fahrzeugtrajektorien erkannt (Conceptual Primitives Level). Auf einer höheren Ebene des CRS (Behavioral Representation Level) werden dann die Abfolgen und das Zusammenspiel einzelner Fahrzeugmanöver analysiert und bestimmte Situationen bzw. Verhalten klassifiziert. Hierbei wird A-Priori-Wissen über typische Situationen und Manöverabfolgen in so genannten Situationsgraphen repräsentiert, welche Situationsknoten und mögliche Übergänge zwischen den Knoten enthalten. Die Situationsknoten bestehen hierbei aus metrisch-temporalen logischen Prädikaten und möglichen Aktionsschemata. Die in diesen Graphen abgelegten Prädikate werden durch die vom Bildverarbeitungssystem gelieferten Ergebnisse instantiiert und mithilfe eines Inferenzverfahrens ausgewertet. Die so gewonnenen Pfade durch den Situationsgraphen repräsentieren die Aktionsfolgen bzw. das Verhalten einzelner Fahrzeuge. Schließlich werden in einem dritten Subsystem natürlichsprachliche Beschreibungen der erkannten Aktionsabfolgen generiert.

Cerebus [44] implementiert Erweiterungen einer verhaltensbasierten Roboterarchitektur, um Kognition auf höheren Ebenen zu ermöglichen. Dabei wird bewusst auf einen traditionellen Planer verzichtet. Die Grundidee basiert auf der Kombination eines sensomotorischen Systems mit einem semantischen Netzwerk, einem einfachen Parser und einem Inferenznetzwerk, das Aufgaben ausführen und Fragen beantworten kann. In einem parallelen Netzwerk werden logische Verknüpfungen in Form von Hornklauseln ausgewertet und Zustandsübergänge in einem endlichen Automaten repräsentiert. Diese realisieren ein Modell von Verhaltensmustern, welche hierarchisch verschachtelt und abgearbeitet werden können. Hierdurch wird eine Verknüpfung von symbolischen und parallel reaktiven Systemen realisiert, welches durch Kompilieren symbolischer Regeln anhand der Generic Robot Language (GRL), einem LISP Dialekt, in ein Laufzeitsystem implementiert wird. Ein weiterer zentraler Aspekt von Cerebus ist das Konzept der Rollenzuweisung (role passing). Diese erlaubt eine Erweiterung der verhaltensbasierten Subsysteme um die Fähigkeit, mit Parametern und Prädikaten umzugehen und ermöglicht die Konstruktion höherer Verhaltensregeln. Eine typische Verarbeitungsschleife in Cerebus besteht in der Erteilung einer Aufgabe durch einfache Sprache, der Zuweisung von Verhaltensregeln zu Rollen und der Ausführung des Verhaltens bis zur Erreichung des Zieles.

5 Architekturen aus der humanoiden Robotik

Auch wenn es inzwischen sehr viele humanoide Roboter in der internationalen Forschung gibt, so sind die meisten von ihnen keine Systeme, die bisher einen sehr hohen Grad an Kognition für ihre Aufgaben verlangen. Vielmehr liegt hier der Schwerpunkt noch hauptsächlich in der Mechatronik, der Regelung und der Koordination der Teilkomponenten. Im Folgenden werden die kognitiven Architekturen einiger ausgewählter humanoider Roboter mit kognitiven Fähigkeiten kurz skizziert.

ISAC (Intelligent Soft Arm Control) [7] ist ein Roboter, der vom *Center for Intelligent Systems* (CIS) an der Vanderbilt Universität entwickelt wurde. Auf ihm wurde die *Intelligent Machine Architecture* (IMA) implementiert, eine Multi-Agenten-basierte kognitive Roboterarchitektur. Diese besteht aus einer Gedächtnis-Hierarchie, Aktuatoragenten, Perzeptionsagenten, einem "Self Agent" sowie einem "Human Agent". Die Gedächtnis-

Hierarchie umfasst dabei ein Kurzzeitgedächtnis, das die Daten der Perzeption in einem räumlichen, kugelförmigen Modell der Umwelt (*Sensory EgoSphere*) speichert. Des Weiteren gibt es ein Langzeitgedächtnis, um gelernte Bewegungsprimitive und episodische Erfahrungen abzuspeichern. Ein Arbeitsgedächtnis speichert die für die jeweilige momentane Handlung notwendigen Informationen aus Kurz- und Langzeitgedächtnis sowie neue Informationen, die während der Handlung gewonnen werden. Der "Self Agent" ist für ISACs kognitive Fähigkeiten verantwortlich, die von Sensorbeobachtung über Planung bis zur kognitiven Steuerung reichen. Innerhalb des "Self Agent" selektiert der Intentionsagent die momentane Roboterhandlung, abhängig von dessen eigenem Zustand sowie der erkannten Intention des menschlichen Gegenübers. Die zentrale Ausführungssteuerung wiederum wurde dafür konzipiert, mit dem "Human Agent", dem Kurz- und dem Langzeitgedächtnis zu interagieren, um Pläne zu erstellen und abzurufen, mit denen dann verschiedene Handlungen des Roboters ausgeführt werden können. Schließlich gibt es noch den "Human Agent", der den menschbezogenen Input der Perzeption bündelt und aufgrund dieser Beobachtungen auf verschiedenen Engagementstufen mit dem Menschen interagiert.

Sowohl **Cog** als auch **Kismet** [15] liegt die gleiche verhaltensbasierte kognitive Architektur zugrunde, die aus sechs Subsystemen besteht: der niederwertigen Wahrnehmung, der höherwertigen Wahrnehmung, dem Aufmerksamkeitssystem, dem Motivationssystem, dem Verhaltenssystem und dem Bewegungssystem. Das niederwertige Wahrnehmungssystem auf der einen Seite und das Bewegungssystem auf der anderen Seite bilden beide zusammen die Schnittstelle des Robotersystems zu seiner Umwelt. Aufgenommene Sensordaten werden im niederwertigen Perzeptionssystem vorverarbeitet und aufbereitet: Dies betrifft insbesondere die Bestimmung von Gesichtsmerkmalen wie Farbe, Kanten, Bewegungen und Ungleichheiten. Des Weiteren liegt in diesem Subsystem der Gleichgewichtssinn des Roboters, die taktile und kinästhetische Wahrnehmung sowie die Extraktion akustischer Merkmale (z.B. Tonhöhe und Energie, Ort). Die hier gewonnenen Ergebnisse fließen dann einerseits in die höherwertige Wahrnehmung zur weiteren Verarbeitung ein, andererseits steuern sie auch das Aufmerksamkeitssystem des Roboters und das Bewegungssystem.

Das höherwertige Wahrnehmungssystem führt sowohl die Ergebnisse aus der niederwertigen Wahrnehmung zusammen, führt aber auch Berechnungen zur komplexeren Wahrnehmung aus. Zur komplexen Wahrnehmung gehört hier aber auch die Selbstwahrnehmung des Roboters, die Aufmerksamkeitswahrnehmung und das Abstimmen des eigenen Verhaltens des Roboters mit den gemachten Beobachtungen. Im Aufmerksamkeitssystem selbst sind die Gewöhnungsmechanismen angesiedelt, die beispielsweise bewirken, dass ein vom Roboter wahrgenommenes, aber nicht für die Interaktion mit dem Menschen relevantes Geräusch, zu späteren Zeitpunkten nicht mehr in das Zentrum der Aufmerksamkeit des Robotersystems rutscht. Gesteuert wird die Aufmerksamkeit des Roboters einerseits von der Wahrnehmung, andererseits bestimmen bei Kismet die übergeordneten Bedürfnisse des Robotersystems, worauf es seine Aufmerksamkeit lenkt. Umgekehrt beeinflusst der Fokus der Aufmerksamkeit auch die im Motivationssystem Kismets verankerten Bedürfnisse. Die Aufgabe des Verhaltenssystems besteht in der Auswahl des geeigneten Verhaltens. Dies schließt sowohl das Abwägen zwischen verschiedenen konkurrierenden Verhaltensweisen des Roboters ein, als auch das Vermeiden eines Verhaltens oder die Ausrichtung der Aktionen des Roboters auf ein Ziel hin. Die ausgewählten Verhaltensweisen des Robotersystems werden zu deren Umsetzung an das Bewegungssystem

weitergeleitet, das die Körperbewegungen des Roboters regelt.

Der Grundgedanke der Architektur zu **Robovie** [45] besteht darin, eine große Anzahl an interaktiven Verhaltensmöglichkeiten zu implementieren. Diese werden in Abhängigkeit von einzelnen Interaktionen in Situationsmodulen in Form von Aktions-Reaktions-Paaren implementiert. Für jede Interaktion sind dabei auch nur spezifische Sensordaten und Kontexte zugelassen. Die Aktionen des Roboters beinhalten dabei Bewegungen, Sprachausgabe und nonverbale Kommunikationskanäle (Augenkontakt, Nicken, Entfernung). Die Reaktion des Menschen wird als erwartetes Verhalten kodiert. Pro Zeiteinheit führt der Roboter nur ein Situationsmodul aus. Um eine Interaktion mit verschiedenen Aktionen durchzuführen, wird eine Reihenfolge der benötigten Situationsmodule mithilfe von Regeln (Episodenregeln) festgelegt. Bevor ein Situationsmodul ausgeführt werden kann, werden die für dieses Modul geltenden Vorbedingungen von der Modulregelung überprüft. Dazu fließen Daten aus Sensormodulen von externen Sensoren, Daten aus dem internen Zustandsmodul des Roboters und die Episodenregeln ein. Da die Interaktion in Form von Aktions-Reaktions-Paaren modelliert ist und der Roboter hauptsächlich aktiv die Interaktion anstößt, folgen die Übergänge zwischen agierendem Roboter und reagierendem Mensch sequentiell aufeinander. In Ausnahmen kommt es aber auch zu rein reaktiven Übergängen, die dann in Form von Reaktionsmodulen festgelegt sind. Dort wird das Antwortverhalten des Roboters auf wahrgenommene Umweltreize kodiert.

Der humanoide Roboter **HERMES** [9, 46] der Bundeswehruniversität München verfügt über einfache Verhaltensweisen, um mit Menschen, die ihm begegnen, ein einfaches Gespräch zu führen, auf Stöße an seine mobile Plattform zu reagieren, oder den Weg zu einem bestimmten Zimmer zu lernen. Die Fähigkeiten des Robotersystems bilden die Schnittstelle zu allen anderen Hardwarekomponenten, zu den Sensoren, den Aktoren, der Wissensbasis und der menschlichen Umwelt. Die Fähigkeiten bekommen beispielsweise Sensordaten als Eingabewerte und verständigen daraufhin ein Situationsmodul. Dieses ist für die Kognition, die Systemverwaltung und die Planung adäquater Verhaltensweisen verantwortlich. Kognitive Funktionen werden dadurch realisiert, dass das Situationsmodul Daten und Informationen aller Systemkomponenten fusioniert, Situationen bewertet und Verhaltensweisen auswählt. In Abhängigkeit des aktuellen Kontextes können unterschiedliche Basisfähigkeiten aktiviert oder deaktiviert werden. Weitere Fähigkeiten kann das Robotersystem autonom oder mithilfe des Menschen erwerben.

6 Vergleich und Diskussion

Ein Vergleich der unterschiedlichen Architekturen ist aufgrund ihrer unterschiedlichen Herkunft und Zielstellungen sehr komplex, da sie dazu dienen können, menschliches Verhalten und menschliche Kognition zu modellieren, intelligentes Verhalten zu generieren (z.B. von humanoiden Robotern) oder spezifische Aufgabenstellungen einfacher zu lösen. Er erfordert auf jeden Fall eine Betrachtung unter unterschiedlichen Kriterien. Erste Ansätze dafür finden sich in [2] und werden im Folgenden erweitert.

Die Kriterien lassen sich für die Anwendung auf humanoide Roboter in drei Komplexe einteilen:

- Die Integration verschiedener Teilaspekte wie Ziele, detaillierter Modelle der Umgebung und des Menschen, eine Spezifikation der Sensomotorik des Roboters und seiner Kommunikationspartner für konkrete Handlungen, die Erkennung und Verarbeitung natürlicher gesprochener Sprache und von Lernverfahren bestimmt die erreich-

bare Funktionalität des Roboters. Wenn eine oder mehrere Komponenten fehlen, ist der Anwendungsbereich eingeschränkt und die entsprechenden Komponenten müssen nachträglich noch in die Architektur integriert werden.

- Die Verarbeitung teilt die interne Informationsverarbeitung in der Architektur nach verschiedenen Kategorien ein. Eine Beschränkung auf eine rein symbolische Verarbeitung erweist sich insbesondere bei der Anwendung in komplexen Umgebungen als problematisch. Vorteilhaft ist daher eine Integration subsymbolischer Ansätze, die wiederum aber komplizierte Repräsentationen und u.U. eingeschränkte Fähigkeiten zur Generalisierung aufweisen. Die zulässige Parallelität verschiedener Handlungen ist eine Voraussetzung für komplexe Verhaltensweisen des Roboters (z.B. gleichzeitiges Zuhören und Bewegen). Eine Hierarchisierung erleichtert die Verwaltung komplexer Handlungen und die mehrfache Wiederverwendung von Aktionen, die in verschiedenen Handlungen vorkommen. Echtzeitfähigkeit ist die Voraussetzung für den experimentellen Einsatz, anderenfalls sind bestenfalls Simulationen oder Experimente in einer künstlich eingeschränkten Umgebung möglich.
- Der Stand der Erprobung der verschiedenen Architekturen ist ebenfalls unterschiedlich. Einige Architekturen wurden bereits mit Roboter-Hardware gekoppelt und somit experimentell erprobt. Andere wurden in Simulationsumgebungen integriert, die eine detaillierte Analyse der Verhaltensweisen erlauben. Beides setzt eine existierende Software-Implementierung der Architektur voraus. Besonders wichtig ist dabei eine freie Verfügbarkeit der Software (F wie Freeware), um eine Erprobung und Weiterentwicklung außerhalb des unmittelbaren Entwicklerteams zu ermöglichen. Andere Plattformen stellen immerhin Demoversionen (D) zur Verfügung, die einen Eindruck über die Leistungsfähigkeit geben. Die erprobten Szenarien weisen zudem eine stark unterschiedliche Komplexität auf. Nur komplexe Szenarien sind dabei in der Lage, eine realistische Prognose über das Einsatzpotential zu ermöglichen.

Tabelle 1 unternimmt den Versuch einer Einteilung der genannten Architekturen anhand der verfügbaren Literaturstellen und der persönlichen Erfahrungen der Autoren.

Die Komponenten einer kognitiven Architektur und ihre Informationsflüsse können unter unterschiedlichen Sichtweisen betrachtet werden:

- Eine *ebenenzentrierte Sicht* ordnet die einzelnen Komponenten des Robotersystems in Relation zu ihrer erforderlichen Verarbeitungsgeschwindigkeit an. Beispielsweise laufen bei einer 3-Ebenen-Architektur schnelle reaktive Komponenten auf der untersten Ebene sowie Bausteine zur Erkennung und zur Aufgabenkoordinierung auf der mittleren Ebene in Echtzeit ab. Die oberste Ebene umfasst hingegen langsamere und komplexe Komponenten, die für die Fusion, Interpretation und Bewertung der Resultate aus den untergeordneten Komponenten verantwortlich sind. Zu dieser Ebene gehören auch Dialog-, Planungs- und Lernkomponenten [14]. Auffällig ist, dass sich viele der in dieser Arbeit genannten kognitiven Architekturen an einer solchen 3-Ebenen-Architektur orientieren.
- Eine *planungszentrierte Sicht* (wie z.B. bei SOAR) zeigt hauptsächlich das Zusammenspiel der kognitiven Komponenten bei der Wahrnehmung und Planung von Handlungen.
- Eine *kommunikationszentrierte Sicht* konzentriert sich auf die Dialogkomponenten des Roboters bei der Auswahl von Handlungen.
- Eine *lernzentrierte Sicht* verdeutlicht hauptsächlich das Zusammenspiel der einzelnen kognitiven Komponenten bei verschiedenen Arten von Lernvorgängen.

ARCHITEKTUR	Integration von					Verarbeitung				Erprobung			
	Zielen und Teilzielen	Modellen	Sensomotorik	natürlicher Sprache	Lernverfahren	auch subsymbolisch	Parallelität	Hierarchisierung	Echtzeitfähigkeit	große Komplexität	Simulation	Roboter-Hardware	Verfügbare Software
SOAR	✓	✓		✓	✓			✓	✓	✓	✓		F
ACT-R	✓		✓		✓	✓		✓			✓	✓	F
LICAI	?			✓		✓							
CoLiDes	?			✓		✓							
EPIC		✓	✓			✓	✓	✓	✓	✓	✓		(✓)
Klix	✓	✓	✓	T	✓	✓	T	T	?	✓			
PSI (Dörner)	✓	✓		✓	✓	✓	✓	✓	?	✓	✓		D
CogAff (Sloman)	T	T	T	T	T	✓	✓	✓	?				
ADAPT	✓	✓		✓	✓	✓	✓	✓		✓			
Cogniron	✓	✓	✓	✓	✓	✓	✓	✓	✓				
Paco-Plus	✓	✓	✓	✓	✓	✓	T	✓	T	✓		✓	
BIRON	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
CVS (Nagel)	T	✓	T	✓	✓	✓		✓	?				
Cerebus	✓		✓			✓	✓	✓	✓			✓	?
ISAC	✓	?	✓	✓	✓	✓	✓		✓	✓	✓	✓	
Cog/Kismet		?	✓		✓	✓	✓		✓	✓		✓	
Robovie	✓	?	✓	✓	?	✓		✓	✓	✓		✓	
Hermes		?	✓	✓	✓	✓	✓		✓			✓	

Tabelle 1: Vergleich kognitiver Architekturen (✓: vorhanden, ?: unbekannt, D: Demoversion, F: Freeware, T: theoretisch denkbar)

Durch die unterschiedlichen Sichtweisen können inhaltlich ähnliche Architekturen vollkommen unterschiedlich dargestellt werden. Außerdem werden in jeder Sicht bestimmte Informationsflüsse ausgeblendet. Der Grund ist die Vielfalt der Informationsflüsse, wodurch im Extremfall alle Komponenten mit allen anderen verbunden sind und kommunizieren. Das Verständnis der Abläufe wird je nach Aufgabe und Sicht also erleichtert oder erschwert. Die verschiedenen Sichtweisen behindern somit einen einfachen Vergleich und sind u.U. eine Quelle für Verwirrungen.

7 Zusammenfassung

Der vorliegende Beitrag gibt eine Übersicht über verschiedene kognitive Architekturen, die aus der kognitiven Psychologie, der Servicerobotik und der humanoiden Robotik stammen. Die genannten Architekturen gehen ursprünglich von verschiedenen Zielstellungen aus, z.B. die Beschreibung menschlichen Verhaltens oder die Erstellung einer Systemarchitektur für agierende Systeme in einer komplexen Umgebung, lassen sich aber prinzipiell alle auf humanoide Roboter anwenden. Ausgehend von einer Diskussion der Anforderungen an intelligente humanoide Roboter werden ausgewählte Architekturen kurz vorgestellt und miteinander nach verschiedenen Kriterien verglichen. Beim Vergleich wird insbesondere auf die Bedeutung verschiedener Sichtweisen auf kognitive Architekturen eingegangen.

Danksagung: Diese Arbeit wird von der Deutschen Forschungsgemeinschaft (DFG) im Rahmen des Sonderforschungsbereiches (SFB) 588 "Humanoide Roboter - Lernende und kooperierende multimodale Roboter" gefördert.

Literatur

- [1] Strube, G.: *Wörterbuch der Kognitionswissenschaft*. Klett-Cotta Verlag. 1996.
- [2] Byrne, M.: Cognitive Architecture. In: *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications* (Jacko, J. A.; Sears, A., Hg.), S. 97–117. Lawrence Erlbaum Associates. 2003.
- [3] Langley, P.; Laird, J. E.; Rogers, S.: *Cognitive Architectures: Research Issues and Challenges*. Techn. Ber., Stanford University, Computational Learning Laboratory (CSLI). 2006.
- [4] Moravec, H.: Stanford Cart and the CMU Rover. *Proc., IEEE* 71(7) (1983), S. 872–884.
- [5] Nilsson, N. J.: *Shakey the Robot*. Techn. Ber. 323, SRI International. 1984.
- [6] Brooks, R. A.: A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation* 2(1) (1986), S. 14 – 23.
- [7] Kawamura, K.; Peters II, R. A.; Bodenheimer, R. E.; Sarkar, N.; Park, J.; Clifton, C. A.; Spratley, A. W.; Hambuchen, K.: A Parallel Distributed Cognitive Control System for a Humanoid Robot. *International Journal of Humanoid Robotics* 1(1) (2004), S. 65–93.
- [8] Pfeifer, R.; Schreier, C.: *Understanding Intelligence*. MIT Press. 1999.
- [9] Bischoff, R.: Design Concept and Realization of the Humanoid Service Robot HERMES. *Field and Service Robotics* (1998), S. 485–492.
- [10] Firby, R. J.: Building Symbolic Primitives with Continuous Control Routines. In: *Proc., 1st International Conference on Artificial Intelligence Planning Systems*, S. 62–69. 1992.
- [11] Connell, J.: SSS: A Hybrid Architecture Applied to Robot Navigation. In: *Proc., IEEE Conference on Robotics and Automation (ICRA)*, S. 2719–2724. 1992.
- [12] Gat, E.: Three-Layer Architectures. In: *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems* (Kortenkamp, D., Hg.), S. 195–210. AAAI Press. 1998.
- [13] Hackel, M.; Schwöpe, S.; Fritsch, J.; Wrede, B.; Sagerer, G.: A Humanoid Robot Platform Suitable for Studying Embodied Interaction. In: *Proc., IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2005.
- [14] Rasmussen, J.: Skills, Rules, Knowledge: Signals, Signs and Symbols and Other Distinctions in Humans Performance Models. *IEEE Transactions on Systems, Man, and Cybernetics* 13 (1983), S. 257–267.
- [15] Breazeal, C.: Socially intelligent robots: research, development, and applications. *Systems, Man, and Cybernetics* 4 (2001), S. 2121–2126.
- [16] Arkin, R. C.; Fujita, M.; Takagi, T.; Hasegawa, R.: An Ethological and Emotional Basis for Human Robot Interaction. *Robotics and Autonomous Systems* 42(3-4) (2003), S. 191–201.
- [17] Dörner, D.: *Bauplan für eine Seele*. Rowohlt. 2002.
- [18] Burghart, C.; Mikut, R.; Holzapfel, H.: Cognition-Oriented Building Blocks of Future Benchmark Scenarios for Humanoid Home Robots. In: *Proc., AAAI Workshop Evaluating Architectures for Intelligence, Vancouver*. 2007.
- [19] Kahn, P.; Ishiguro, H.; Friedman, B.; Kanda, T.: What is a Human? Toward Psychological Benchmarks in the Field of Human–Robot Interaction. In: *Proc., IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 2006.
- [20] Dillmann, R.; Becher, R.; Steinhaus, P.: ARMAR II - A Learning and Cooperative Multimodal Humanoid Robot System. *International Journal of Humanoid Robotics* 1(1) (2004), S. 143–155.
- [21] Asfour, T.; Regenstein, K.; Azad, P.; Schröder, J.; Dillmann, R.: ARMAR-III: A Humanoid Platform for Perception-Action Integration. In: *Proc., International Workshop on Human-Centered Robotic Systems (HCRS), Munich*, S. 51–56. 2006.
- [22] Burghart, C.; Holzapfel, H.; Haeussling, R.; Breuer, S.: Coding Interaction Patterns Between Human and Receptionist Robot. In: *Proc., Humanoids, Pittsburgh*. Submitted paper. 2007.

- [23] Lehman, J.; Laird, J.; Rosenbloom, P.: A Gentle Introduction to Soar: 2006 Update. Techn. Ber., University of Michigan. 2006.
- [24] Anderson, J. R.; Lebiere, C.: *The Atomic Components of Thought*. Mahwah, NJ: Lawrence Erlbaum Associates. 1998.
- [25] Anderson, T.: Biomechanics and Running Economy. *Sports Med* 22 (1996) 2, S. 76–89.
- [26] Anderson, J. R.; Matessa, M.; Lebiere, C.: ACT-R: A Theory of Higher Level Cognition and its Relation to Visual Attention. *Human-Computer Interaction* 12 (1997), S. 439–462.
- [27] Kitajima, M.; Soto, R.; Polson, P.: LICAI+: A Comprehension-Based Model of the Recall of Action Sequences. In: *2nd European Conference on Cognitive Modelling (ECCM'98)*, S. 82–89. 1998.
- [28] Kitajima, M.: The LICAI Model. [http://staff.aist.go.jp/kitajima.muneo/English/Project\(E\)/CognitiveModeling\(E\)/LICAI\(E\)/LICAI\(E\).html](http://staff.aist.go.jp/kitajima.muneo/English/Project(E)/CognitiveModeling(E)/LICAI(E)/LICAI(E).html). 2001.
- [29] Kitajima, M.; Blackmon, M.; Polson, P.: A Comprehension-Based Model of Web Navigation and Its Application to Web Usability Analysis. In: *People and Computers XIV - Usability or Else!* (McDonald, S.; Waern, Y.; Cockton, G., Hg.), S. 357–373. Springer. 2000.
- [30] Kieras, D.; Meyer, D.: An Overview of the EPIC Architecture for Cognition and Performance with Application to Human-Computer Interaction. *Human-Computer Interaction* 12 (1997), S. 391–438.
- [31] Kieras, D. E.; Meyer, D. E.: EPIC: A Cognitive Architecture for Computational Modeling of Human Performance. <http://www.eecs.umich.edu/kieras/epic.html>. 2005.
- [32] Meyer, D. E.: The Brain, Cognition, and Action Laboratory: EPIC. <http://www.umich.edu/bca-lab/epic.html>. 2005.
- [33] Klix, F.: *Information und Verhalten - Kybernetische Aspekte der organismischen Informationsverarbeitung - Einführung in naturwissenschaftliche Grundlagen der allgemeinen Psychologie*. Hans Huber. 1976.
- [34] Klix, F.: *Erwachendes Denken, Eine Entwicklungsgeschichte der menschlichen Intelligenz*. Berlin: VEB Deutscher Verlag der Wissenschaften. 1980.
- [35] Dörner, D.; Levi, P.; Detje, F.; Becht, M.; Lippold, D.: Der agentenorientierte, sozionische Ansatz mit PSI. *Sozionik aktuell* 2 (2001).
- [36] Sloman, A.; Logan, B.: *Evolvable Architectures for Human-Like Minds*. Nagoya: Elsevier. 1999.
- [37] Sloman, A.: *Architectural Requirements for Human-Like Agents Both Natural and Artificial (What Sorts of Machines Can Love?)*. John Benjamins. 2000.
- [38] Benjamin, D. P.; Lyons, D.; Lonsdale, D.: ADAPT: A Cognitive Architecture for Robotics. In: *Proc., International Conference on Cognitive Modeling*. 2004.
- [39] Lacroix, J.; Hommel, B.: Internal Report. University of Leiden, The Netherlands. 2007.
- [40] O'Reilly, R.; Rudy, J.: Conjunctive Representations in Learning and Memory: Principles of Cortical and Hippocampal Function. *Psychological Review* 108 (2001), S. 311–345.
- [41] Alami, R.; Chatila, R.; Clodic, A.; Fleury, S.; Herrb, M.; Montreuil, V.; Sisbot, E.: Towards Human-Aware Cognitive Robots. In: *AAAI Workshop on Cognitive Robotics*. Boston. 2006.
- [42] Haasch, A.; Hohenner, S.; Hüwel, S.; Kleinhagenbrock, M.; Lang, S.; Topsis, I.; Fink, G. A.; Fritsch, J.; Wrede, B.; Sagerer, G.: BIRON – The Bielefeld Robot Companion. In: *Proc. Int. Workshop on Advances in Service Robotics*, S. 27–32. 2004.
- [43] Nagel, H.-H.: Steps Towards a Cognitive Vision System. *AI Magazine* 25(2) (2004), S. 31–50.
- [44] Horswill, I.; Zubek, R.; Khoo, A.; Le, C. D.; Nicholson, S.: The Cerebus Project. In: *Proc., AAAI Fall Symposium on Parallel Cognition*. 2000.
- [45] Kanda, T.; Hirano, T.; Eaton, D.; Ishiguro, H.: Interactive Robots as Social Partners and Peer Tutors for Children: A Field Trial. *Human-Computer Interaction* 19(1-2) (2004), S. 61–84.
- [46] Bischoff, R.; Graefe, V.: Integrating Vision, Touch and Natural Language in the Control of a Situation-Oriented Behavior-Based Humanoid Robot. In: *Proc., IEEE International Conference on Systems, Man, and Cybernetics*, S. 999–1004. 1999.

Ein Gedankenexperiment zum Entwurf einer integrierten kognitiven Architektur für humanoide Roboter

Ralf Mikut¹, Catherina Burghart², Alexej Swerdlow³

¹Forschungszentrum Karlsruhe GmbH, Institut für Angewandte Informatik,

²Universität Karlsruhe (TH), Institut für Prozessrechentechnik, Automation und Robotik

³Universität Karlsruhe (TH), Institut für Nachrichtentechnik

E-Mail: ralf.mikut@iai.fzk.de, burghart@ira.uka.de, swerdlow@int.uka.de

1 Einleitung

Die Entwicklung menschenähnlicher (humanoider) Roboter war stets von hohen Erwartungen der zukünftigen Nutzer an die Leistungsfähigkeit solcher Roboter begleitet. Der Roboter soll nützliche Tätigkeiten in einer für Menschen geschaffenen Umgebung (z.B. in einer Wohnung) ausüben, aber auch auf natürliche, angemessene, berechenbare und angenehme Weise mit dem Menschen kommunizieren können. Die damit verbundenen Anforderungen an die Sicherheit, Zuverlässigkeit und Leistungsfähigkeit setzen neben umfassenden sensomotorischen Fähigkeiten auch Fähigkeiten zum Erfassen, Interpretieren und Bewerten von komplexen Situationen voraus (siehe u.a. [1, 2]). Diese bilden die Voraussetzung zu einer situationsangepassten Entscheidungsfindung, um Roboterhandlungen auszuwählen und zu parametrieren.

Zur Erfüllung dieser Erwartungshaltung ist eine leistungsfähige kognitive Architektur erforderlich. Zwar gibt es seit Jahren eine Vielzahl relevanter Arbeiten (siehe z.B. [3–5] für entsprechende Übersichten), die aber die obengenannten Anforderungen noch nicht oder nur zu kleinen Teilen erfüllen können. Die meisten Architekturen konzentrieren sich auf bestimmte Aspekte. Viele der scheinbar gegensätzlichen Architekturen haben deshalb das Potenzial, sich gegenseitig zu ergänzen, um als Bausteine in eine Gesamtarchitektur integriert zu werden. Insofern ist keine komplett neue Architektur notwendig, sondern eher eine Integration guter Ideen aus den bereits vorgeschlagenen Architekturen sinnvoll.

Für die experimentelle Erprobung ist eine leistungsfähige Roboterhardware mit einer vollständig implementierten kognitiven Architektur erforderlich. Da hier noch viele Entwicklungsarbeiten laufen (viele Architekturen existieren lediglich als Konzeptstudien auf dem Papier), ist eine Kombination aus zwei Vorgehensweisen sinnvoll. Einerseits müssen in einem Bottom-Up-Ansatz vorhandene Teilkomponenten von Architekturen auf dem Roboter experimentell erprobt und stückweise abstrahiert werden [6]. Andererseits sind in einem Top-Down-Ansatz mit Gedankenexperimenten mögliche Konzepte für eine integrierte kognitive Architektur zu analysieren, um frühzeitig aussichtsreiche Wege zu identifizieren und mögliche Irrwege zu falsifizieren.

Der vorliegende Beitrag widmet sich dem Top-Down-Ansatz und versucht anhand eines Szenarios für einen humanoiden Roboter in einem Haushalt (Kapitel 2), potenzielle Komponenten und Informationsflüsse in einer kognitiven Architektur zu skizzieren (Kapitel 3). Bei der Planung des Szenarios (Kapitel 4) werden insbesondere die Rolle von Ausgangssituationen, Bewertungen und vorhandenen Ressourcen des Roboters für die Planung und Entscheidungsfindung in komplexen Situationen sowie die unterschiedlichen Zeitskalen bei der echtzeitfähigen Planung von Handlungen herausgearbeitet. Die nachfolgende Diskussion (Kapitel 5) erläutert offene Fragen und zukünftige Schritte.

2 Ein Szenario und seine Tücken

Das Betrachten eines konkreten Szenarios verdeutlicht das Ausmaß der erforderlichen kognitiven Fähigkeiten des Roboters und des notwendigen Zusammenspiels der verschiedenen Komponenten bei der Planung von Handlungen. Wenn ein Roboter beispielsweise von einer Person namens Berta den Befehl erhält, eine Tasse zu holen, muss er die Tasse in der Umgebung finden, hinfahren, sie mit seiner Hand greifen, zum Menschen zurückkehren und ihm die Tasse reichen, so dass der Mensch die Tasse abnehmen kann. Das so schlicht anmutende Szenario ist leider nicht trivial:

Der *Befehl* zum Holen der Tasse kann auf verschiedene Weise entstehen:

- Im einfachsten Fall sagt die Person: "Bring mir die rote Tasse!" oder sie sagt "Bring mir DIE Tasse!" und zeigt darauf. Wenn nur eine passende Tasse sichtbar ist und der Roboter nicht aufgrund einer zu stark vereinfachten Farberkennung einen anderen roten Gegenstand apportiert, ist alles eine Frage der Sensomotorik des Roboters, was bereits eine schwierige Aufgabe darstellt.
- Schlimmer wird es, wenn der Befehl abstrakteres Kontextwissen voraussetzt. Beispiele dafür sind "Bring mir MEINE Tasse!" (Der Roboter sollte wissen, dass die bei Berta rot ist.) oder "Bring mir einen Kaffee!" (Der Roboter sollte wissen, dass der in Tassen und nicht in Gläsern serviert wird.). Außerdem muss der Roboter wissen, wo die Tasse in der Regel zu finden ist (z.B. im dritten Schrank von rechts).
- Bertas Wunsch kann auch ohne expliziten Befehl aus dem Kontext erkannt werden. Ein Beispiel ist ein Dialog mit einer dritten Person: "Trinken wir einen Kaffee?" "Aber gern!". Jetzt geht es um zwei Tassen und die rote kommt ungefragt zu Berta.
- Wenn hingegen der Fernseher läuft und dort ein Schauspieler einen Kaffee will, sollte da keiner hingbracht werden.
- Auch Berechtigungen zu einem Befehl sind keinesfalls trivial. Soll der Roboter einem Kind oder einer Alkoholikerin namens Berta eine Tasse mit Grog bringen? Soll die gleichnamige Einbrecherin einen Roboter per Befehl aus dem Raum schicken können?

Der Mensch erwartet eine Rückmeldung, dass der Roboter den Befehl verstanden hat und wie er ihn auszuführen gedenkt. Das kann implizit geschehen (Roboter dreht seinen Kopf zur Tasse, nickt und bewegt sich in angemessener Geschwindigkeit dahin) oder durch eine Sprachausgabe erfolgen.

Die geeignete Handlung hängt von der Ausgangssituation ab. Wenn der Roboter z.B. gerade die Spülmaschine ausräumt, muss er diese Handlung sicher unterbrechen, ehe er sich Berta zuwendet. Ganz ähnliche Anforderungen gibt es dann zur Festlegung der Zielposition (Tisch neben Bertas Sessel, Übergabe an Berta am aktuellen Platz oder am Esstisch?) oder implizite Erwartungen über Wege (NICHT der Weg zwischen dem Fernseher und Bertas Mann während eines Elfmeterschießens usw.). Kurzum, für eine geeignete Handlungsplanung muss der Roboter einen Befehl also verstehen, verifizieren und spezifizieren. Danach wählt er eine adäquate Handlung aus und parametrisiert sie.

Während der Ausführung der Handlung muss der Roboter die Umgebung fortwährend erfassen, interpretieren und überprüfen. Bei geänderten Situationen muss der Plan u.U. strukturell oder parametrisch modifiziert werden. Beispiele hierfür sind:

- Der Roboter soll helfende Hinweise des Menschen (Bertas Ruf "Nein, die!" mit der ausgestreckten Hand zur Tasse) wahrnehmen und als Indikatoren für die Zielposition interpretieren.

- Wenn die Tasse zu Berta soll, muss der Roboter ständig Bertas Position verfolgen. Wenn sie dann woanders sitzt, ist die Zielposition zu modifizieren. Das Gleiche gilt dafür, dass eine andere Person die Tasse wegräumt, bevor der Roboter dort ist.
- Stellt sich die Tasse als nicht sauber oder kaputt heraus, ist sie entsprechend auch nicht zu befüllen. Berta ist eine blaue saubere Tasse sicher lieber als ihre rote benutzte.
- Wenn der Roboter hinter sich ein lautes und unbekanntes Geräusch im Raum wahrnimmt, soll er sich in der Regel umdrehen, um die Situation zu verstehen. Aber nicht gerade dann, wenn er eine Tasse mit heißer Flüssigkeit neben Berta abstellt. Das gleiche gilt für höfliche Begrüßungen, wenn eine weitere Person den Raum betritt.
- Wenn ein Roboter die Tasse nicht erreichen kann (z.B. weil ein schwerer Sessel im Weg steht), soll er das Berta mitteilen.

Die Komplexität solcher Szenarien übersteigt die Fähigkeiten bisheriger Roboter bei weitem. Die folgenden Abschnitte schildern erste Ansätze, die aus Sicht der Autoren mittelfristig zur Bewältigung beitragen können.

3 Planungszentrierte Sicht auf das Szenario

3.1 Übersicht

Auf die Komponenten und Informationsflüsse einer kognitiven Architektur sind verschiedene Sichtweisen möglich [3], die jeweils andere Schwerpunkte betonen. Eine *ebenenzentrierte Sicht* ordnet die einzelnen Komponenten des Robotersystems nach ihrer erforderlichen Verarbeitungsgeschwindigkeit an (z.B. 3-Ebenen-Architektur [7]). Eine *kommunikationszentrierte Sicht* konzentriert sich auf Dialoge mit dem Menschen, eine *lernzentrierte Sicht* auf den Erwerb von Wissen und Fähigkeiten. Der folgende Abschnitt basiert im Wesentlichen auf einer *planungszentrierten Sicht* zur Interpretation der aktuellen Situation und der Planung von Handlungen. Diese Sichtweisen sind insbesondere beim Vergleich von Architekturen zu beachten, weil im Kern ähnliche Architekturen durch verschiedene Sichten scheinbar sehr unterschiedlich dargestellt werden können.

Beim Aufbau des Handlungswissens und der Planung in einer kognitiven Architektur sind zwei Extremstrategien denkbar:

- **Pragmatische Strategie:** Eine bestimmte wiedererkannte Situation wird fest mit einer auszuführenden Aktionsfolge assoziiert. Beispielsweise wird ein vorgeteates Programm abgespielt, wenn ein bestimmter Befehl erkannt wird. Eventuell bestehen fest definierte Freiheiten in der Feinplanung (Lokalisierung der roten Tasse nach dem Befehl "Hole die rote Tasse!"). Die Ressourcenverwaltung ist ebenfalls fixiert. Diese Strategie ist besonders bei Servicerobotern verbreitet. Mit ihr sind eindrucksvolle Choreographien und Demonstrationen möglich, solange sich der Roboter in strukturell bekannten Szenarien (bekannte Situationen, Personen, Befehle; robotergerechte Umgebung und Objekte) bewegt. Für bestimmte Aufgaben wie kollisionsfreies Ausweichen im Messegetümmel reicht diese Strategie u.U. bereits aus.
- **Lerndominierte Strategie:** Der Roboter versucht, alles aus dem Nichts zu lernen. Er kann und muss somit sich selbst (z.B. mögliche Bewegungen) und seine komplette Umgebung explorieren (z.B. neue Objekte, Räume, Personen), menschliche Handlungen beobachten und imitieren [8], Assoziationen (z.B. "Tassen kann ich greifen.")

selbst oder in Rücksprache mit den Benutzern erfassen und durch eigenständige Überlegung (engl. deliberation) entsprechende Pläne aufstellen (z.B. Routenplanung, Griffe) [9]. Diese Strategie ist langfristig vielversprechend, aber derzeit in ihrer Komplexität beschränkt und meist auf spezielle Teilaufgaben und bzw. oder ungefährliche Spielwelten reduziert.

Zwischen diesen Extremen gibt es verschiedene Mischformen, bei denen z.B. einzelne eher lerndominierte Komponenten in eine pragmatisch motivierte Architektur eingebunden werden oder umgekehrt. Somit wird versucht, die Praxisrelevanz der pragmatischen Strategie mit der Flexibilität der lerndominierten Strategie zu verbinden. Ein Vorschlag für eine solche Mischform wird im Folgenden erläutert. Bild 1 zeigt die planungszentrierte Sicht auf diese Mischform, die zudem einige Informationsflüsse aus einer lernzentrierten Sicht enthält.

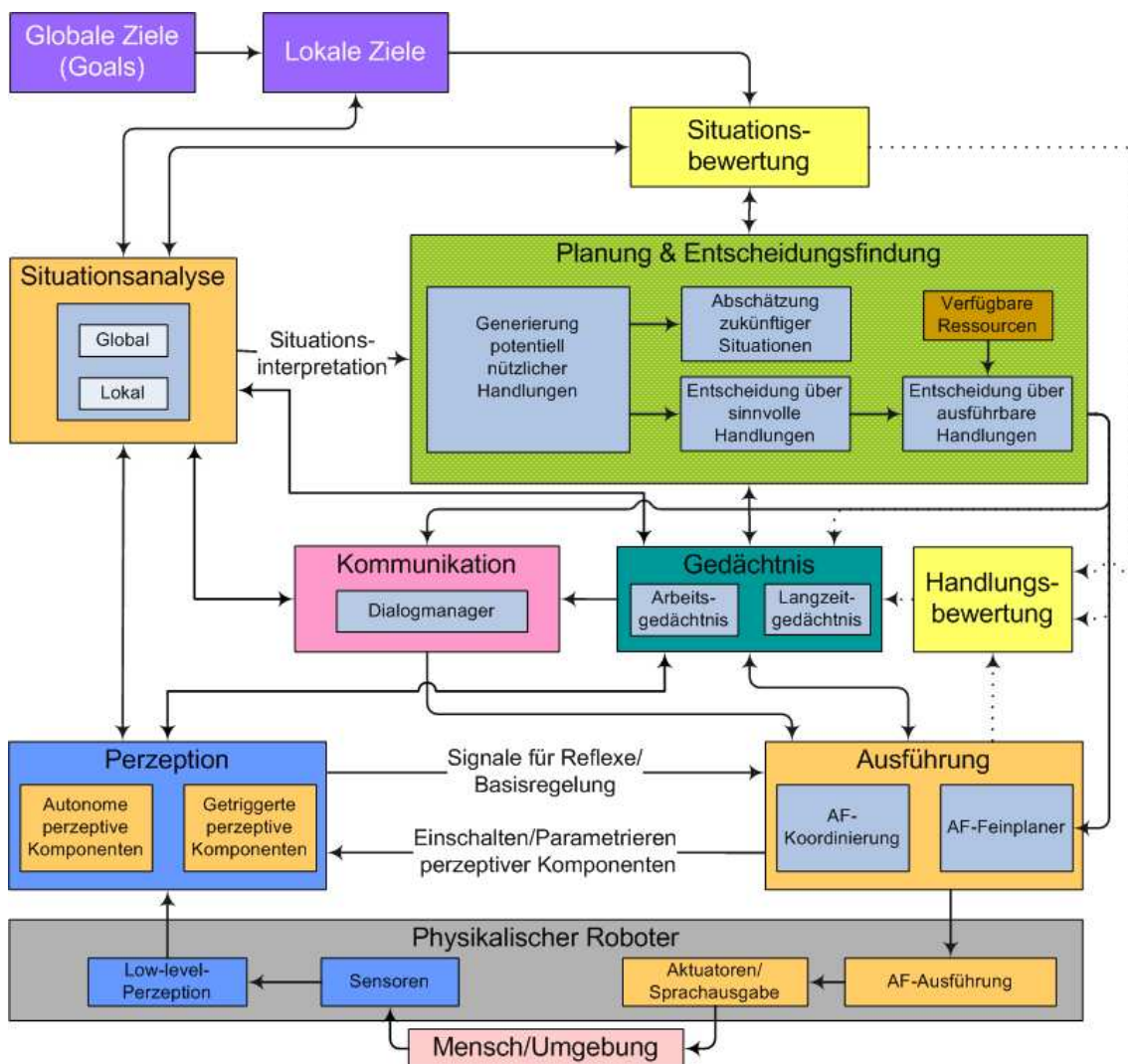


Bild 1: Planungszentrierte Sicht der kognitiven Komponenten des Roboters (ausgewählte Informationsflüsse einer lernzentrierten Sicht in gepunkteten Linien)

Aus den aktivierten Komponenten der **Perzeption** wird über mehrere Hierarchieebenen hinweg die vom Roboter wahrgenommene Szene aktuell gehalten (siehe z.B. [10]). Einige perzeptive Komponenten sind autonom und ständig aktiv, andere werden erst durch

Aktionsfolgen getriggert ein- oder ausgeschaltet. Dabei fließen mit jeder höheren Ebene in stärkerem Umfang Inhalte aus dem **Gedächtnis** (Lang- und Arbeitsgedächtnis LZG bzw. AG) mit ein. Dabei ist es zur Effizienzsteigerung ratsam, nur die gerade relevanten Informationen (z.B. alles über Berta) in das Arbeitsgedächtnis zu laden.

Das Ergebnis der **Situationsanalyse** ist eine interpretierte Situation, die auch Vermutungen des Roboters über die Semantik bestimmter Aussagen umfasst (Berta → meine Tasse → rote Tasse). Wichtig ist hierbei eine Einschätzung des Roboters über seinen eigenen Zustand und den seiner Umgebung. Dabei sind möglichst detaillierte Modelle aller mit ihm kommunizierenden Benutzer von zentraler Bedeutung. Der für die Leistungsfähigkeit der kognitiven Architektur extrem wichtige Block der Situationsanalyse erfordert zwar große Anstrengungen, steht aber nicht im Fokus dieses Beitrags.

Die aktuelle Situation wird unter Bezug auf die **globalen und lokalen Ziele** des Roboters (Abschnitt 3.2) einer **Situationsbewertung** unterzogen (Abschnitt 3.3). Daraus entstehen Aussagen über vorliegende Bedürfnisse und Absichten des Roboters zur Änderung der aktuellen Situation.

Die **Planung und Entscheidungsfindung** erfolgt zunächst auf einer relativ hohen Abstraktionsebene in Form von auszuwählenden Handlungen (Abschnitt 3.4). Die eigentliche koordinierte **Ausführung** der Aktionsfolgen (Abschnitt 3.5) erfolgt weitgehend entkoppelt von der Planung- und Entscheidungsfindung. Hier werden vorgeplante Ereignisse zur Koordinierung der Subsysteme verwendet sowie Algorithmen zur Trajektoriengenerierung und Regler situationsabhängig ausgewählt und konfiguriert [11, 12]. Wichtig ist hier eine relativ große Anzahl von Reflexen. Diese sind mit bestimmten Perzeptionsergebnissen auf verschiedenen Ebenen direkt verschaltet und werden über Aktionsfolgen aktiviert oder deaktiviert. Die Ausführung der Reflexe erfolgt ohne eine explizite und u.U. zeitraubende Bewertung und sorgt somit implizit für eine Einhaltung der globalen und lokalen Ziele. Nach der Ausführung erfolgt eine **Handlungsbewertung** (Abschnitt 3.6), deren Ergebnisse in das Langzeitgedächtnis eingetragen werden.

Eine Implementierungsvariante für das Handlungswissen sind Petri-Netze [11, 13], deren Plätze u.a. Aktionsfolgen und deren Bestandteile (z.B. Aktionen) oder allgemeine Operatoren beschreiben können. Parallele (Teil-) Handlungen sowie die Ressourcenverwaltung werden miteinander durch Ereignisse synchronisiert, die als Transitionen kodiert werden. Somit ergeben sich hierarchische, steuerungstechnisch interpretierte Petri-Netze.

Der **Dialogmanager** (Abschnitt 3.7) ist die derzeit am weitesten untersuchte und implementierte Komponente [14]. Er spielt eine Sonderrolle, weil er in besonders starkem Maße parallel zu anderen Komponenten arbeitet und mit diesen kommuniziert.

3.2 Interne globale und lokale Ziele des Roboters

Der Roboter benötigt zunächst einmal globale Ziele, um die Grundregeln des Verhaltens des Roboters zu etablieren. Dazu gehören allgemeine Gesetze wie z.B. die Asimovschen Robotergesetze [15, 16]. Diese fordern mit abnehmender Priorität, dass der Roboter keinen Menschen verletzt oder durch Untätigkeit zu Schaden kommen lässt, menschlichen Befehlen gehorcht und seine eigene Existenz schützt. Daraus sollen erwünschte Präferenzen wie die Vermeidung von Kollisionen (insbesondere mit Menschen), das Warnen vor erkannten Gefahren, das Ausführen von Befehlen und Wünschen, das Herbeiführen der Zufriedenheit des Menschen usw. abgeleitet werden.

Aus diesen globalen Zielen ergeben sich untergeordnete lokale Teilziele, die nach Ansicht des Roboters zur Erreichung der globalen Ziele beitragen. Dazu gehören Ziele wie das erfolgreiche, schnelle und effiziente Ausführen geplanter Handlungen, der Erhalt der eigenen Leistungsfähigkeit (z.B. ausreichende Aufladung des Energiespeichers) und die Fähigkeit zur zutreffenden Prognose zukünftiger Situationen, die ein Maß für die Vorhersagefähigkeit des Roboters ist. Hinzu kommt das Ziel, einen aktuellen Überblick über seine Umgebung zu haben.

Im Szenario setzt der Roboter somit das Erfüllen von Bertas Wunsch auf seine aktuelle Liste der lokalen Ziele. Dieses Ziel konkurriert aber unter Umständen mit anderen genannten Zielen.

3.3 Situationsbewertung

Die obengenannten Ziele werden in erster Linie für die Handlungsplanung benötigt, damit der Roboter aktuelle und zukünftige erwünschte Situationen von unerwünschten unterscheiden kann. Dazu ist eine geeignete interne Repräsentation der Ziele notwendig. Dörner [17] schlägt beispielsweise vor, Ziele in Form von Motiven zur Erfüllung von Bedürfnissen (soziale Bindung, Energiehaushalt als Analogie von Hunger/Durst, Schmerzvermeidung, Kompetenz als Fähigkeit zur Beeinflussung und Vorhersage zukünftiger Situationen) zu kodieren. Die detaillierte Einschätzung wird dann in eine (quantitative) Situationsbewertung überführt.

Außerdem ist darauf hinzuweisen, dass der Roboter die Situation immer aus seiner Sicht bewertet. Die Ursache besteht darin, dass er über keine direkten Informationen zu den Wünschen des Benutzers verfügt, sondern hier auf seine Modellannahmen (z.B. Bertas Bedürfnis nach einem Kaffee) angewiesen ist. Allerdings erhält der Roboter vom Menschen nur im bestimmten Umfang ein Feedback über die Erreichung dieser Ziele. Deswegen ist es sinnvoll, auch die Unsicherheit über die eigene Einschätzung in der Situationsanalyse mitzuführen.

3.4 Planung- und Entscheidungsfindung

Auf dieser Ebene wird die Handlung zunächst in Form (allgemeiner) Operatoren repräsentiert und noch nicht konkretisiert. Die Operatoren stehen in einer (zunächst feststehenden) Bibliothek (Beispiele: "Mache Nichts", "Hole PERSON ETWAS", "Schau zum Geräusch", "Rede mit PERSON"), die entsprechend zu parametrieren sind (Bild 2).

Die Entscheidung über aussichtsreiche Handlungen erfolgt über eine stark vereinfachte Abschätzung der zukünftigen Situation, wenn die Handlung ausgeführt wird (z.B. "Berta nicht mehr durstig", "Berta lobt mich", "Berta durstig", "Berta unzufrieden", "Berta ärgerlich", "Roboter-Energie reduziert", "Geräuschquelle identifiziert"). Hierbei müssen Unsicherheiten und zeitliche Aspekte (z.B. Erfolgsaussichten und Dringlichkeit von Handlungen) einbezogen werden.

Auch solche zukünftigen Situationen werden bewertet. Das entspricht einem Probedandeln in Gedanken und erfordert ein hochwertiges Handlungswissen im Langzeitgedächtnis, das solche Abschätzungen in vorprogrammierter oder sukzessive erlernter Form enthält. Ein Beispiel sind erwartete Reaktionszeiten für die Perzeption und nachfolgende Aktionen des mit dem Roboter kommunizierenden Menschen [18].

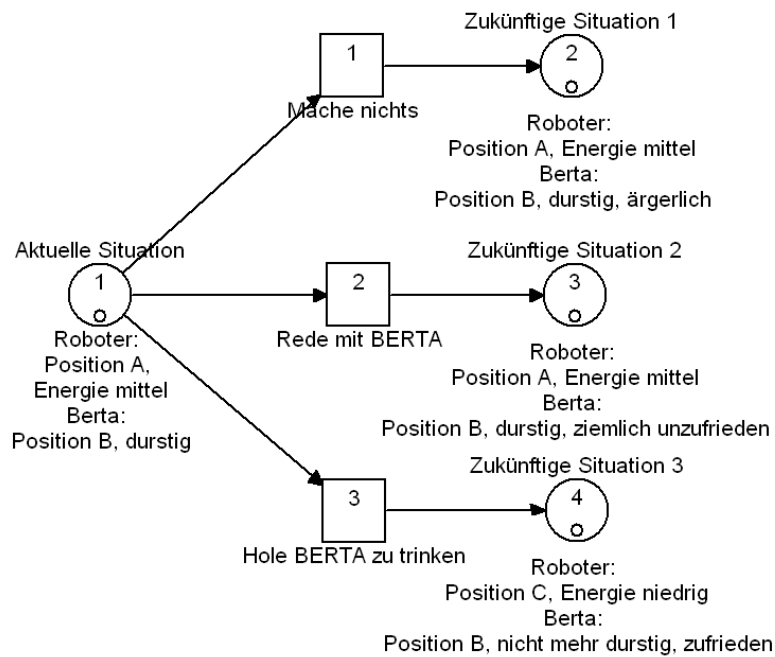


Bild 2: Mögliche Petri-Netz-Kodierung der Auswahl von allgemeinen Operatoren für Handlungen

Mit einem allgemeinen Operator für eine Handlung sind eine oder mehrere Aktionsfolgen (z.B. "Bringe tragbares Objekt O von einem Ort A zu einem anderen Ort B.") assoziiert, von denen eine zur möglichen Ausführung ausgewählt wird. Eine Aktionsfolge umfasst in der Regel mehrere parallele und sequentielle Aktionen (siehe Abschnitt 3.5, z.B. "Fahre von A nach B.", "Greife Objekt"), die auch Informationen zur Koordinierung der Aktionen enthalten (z.B. wahrgenommene Ereignisse wie "Position erreicht", "Objekt lokalisiert", "Objekt gegriffen"). Diese Aktionsfolgen sind zunächst mehr oder weniger manuell zu entwickeln und im Handlungswissen abzuspeichern.

Daraus ergibt sich eine nach Prioritäten sortierte Liste mit einer oder mehreren sinnvollen Handlungen, die bezüglich der zukünftigen Situationen eine gute Bewertung aufweisen. Die Entscheidung über ausführbare Handlungen berücksichtigt noch die verfügbaren Ressourcen. Die benötigten Ressourcen (z.B. Kopf, Sprachausgabe) für Operatoren und Aktionsfolgen müssen folglich im Handlungswissen hinterlegt sein. Die Herausforderung besteht darin, eine gegenseitige Blockade von zu startenden, derzeit ausgeführten und unterbrochenen (wartenden) Aktionsfolgen (Verklemmungen) zu vermeiden. Ein Ansatz ist ein zulässiges "Ausborgen" bestimmter Ressourcen von unterbrochenen Aktionsfolgen mit einem späteren Abbruch, falls die Ressource nicht zurückgegeben werden kann.

3.5 Ausführung

Die ausgewählte Aktionsfolge ist in der Feinplanung geeignet zu parametrieren (Objekt = Tasse, Ort A = Tisch neben Kaffeemaschine, Ort B = Tisch neben Berta) und zu starten. Zur Beherrschbarkeit der hohen Systemkomplexität bestehen Aktionsfolgen aus einer oder mehreren sequentiellen oder parallelen Aktionen oder Aktionsfolgen (siehe z.B. [13]). Eine Aktion setzt sich wiederum aus einer oder mehreren sequentiellen oder (koordinierten) parallelen Elementaraktionen zusammen. Eine Elementaraktion umfasst

eine oder mehrere parallele Elementaroperationen, die je eine primitive sensomotorische Operation für einen einzelnen Freiheitsgrad (z.B. einen Regelkreis) beschreiben.

Aktionen können (und müssen) aber auch zur Laufzeit fest programmierte Strategien zur schritthaltenden Planung enthalten, ohne dass der Prozess der Planung- und Entscheidungsfindung erneut explizit eingreift. Beispiele hierfür sind fertig konfektionierte Aktionsfolgen, welche die Zielposition der Tasse dynamisch nachführen und in der Bibliothek bereits abgespeicherte Aktionen zu deren Erreichen enthalten (z.B. Tasse in Griffweite: Armbewegungen, Tasse nicht in Griffweite: erst hinfahren usw.). Ab hier entspricht der Ablauf im wesentlichen der pragmatischen Strategie.

Aus der ebenenzentrierten Sicht einer 3-Ebenen-Architektur wie in [11] erstreckt sich die Ausführung einer Handlung über die mittlere und untere Ebene.

3.6 Handlungsbewertung

Für die Qualität der Situationsanalyse sowie der Planung- und Entscheidungsfindung ist ein leistungsfähiges Gedächtnis des Roboters notwendig. Neben einer hier nicht betrachteten Offline-Programmierung (z.B. Programmieren durch Vormachen [19]) sind mehrere Online-Informationsquellen denkbar, um die Gedächtnisinhalte Stück für Stück zu verbessern und verfeinern.

Einerseits gibt es natürlich "bewusste" Lernvorgänge, die vom Roboter gezielt gestartet werden. Ein Beispiel hierfür ist ein Dialog, bei dem der Roboter ein Objekt präsentiert bekommt, sich einen Merkmalsvektor für eine spätere Wiedererkennung sowie damit assoziierte Begriffe ("Müsli-Packung") abspeichert [20].

Andererseits entstehen auch viele wertvolle Informationen nebenbei aus den ausgeführten Handlungen. So ist es sinnvoll, für allgemeine Operatoren, Aktionsfolgen, Aktionen und Elementaraktionen bestimmte Informationen zu archivieren. Dabei werden immer bestehende Werte aktualisiert und zeitlich aggregiert (z.B. Summe, Extrema und Filter mit exponentiellem Vergessen):

- Erfolgs- und Misserfolgshäufigkeiten (z.B. über Abbruch ja/nein, vordefinierte Erfolgs- oder Misserfolgszustände),
- Häufigkeiten bestimmter Ereignisse (z.B. in einer Repräsentation als Hidden-Markov-Modell) in bestimmten Zuständen,
- Ausführungszeit,
- Assoziationen mit bestimmten Situationen und Handlungen, z.B. in Form von Wahrscheinlichkeiten für die Übergänge "Handlung → Handlung" und "Situation → Handlung", sowie
- ermittelte Parametrierungen aus dem Feinplaner (wichtig, um später ähnliche Feinplanungen abkürzen zu können).

Diese Werte können allgemein, aber auch in Abhängigkeit von wenigen ausgewählten Parametern erfasst werden. Sie werden zunächst initialisiert und später durch den Roboter nach seinen Erfahrungen adaptiert.

Das Ziel besteht darin, spätere Planungsvorgänge abzukürzen und in ihren Erfolgsaussichten zu verbessern. Vorteilhaft ist hier die Tatsache, dass das Lernen in einer dem Roboter bekannten Repräsentation erfolgt (engl. "grounding").

3.7 Dialogmanager

Der Dialogmanager verarbeitet Spracherkennungshypothesen aus dem kontinuierlichen aktiven Spracherkennungssystem, das eine Komponente der Perzeption darstellt. Diese Hypothesen werden durch die Situationsanalyse weitergeleitet. Der Dialogmanager besitzt eine ereignisbasierte Architektur und stößt Aktionen durch erkannte Ereignisse (z.B. dem Eingehen einer Hypothese) an.

Diese Spracherkennungshypothese wird zunächst bezüglich ihres semantischen Inhalts interpretiert, dann in den vorliegenden Kontext eingeordnet und in Relation zu diesem erneut interpretiert. Zusätzlich werden verschiedene Eingangsmodalitäten fusioniert (z.B. gesprochene Sprache und zeitgleiche begleitende Gesten des Sprechers). Alle Daten des Dialogs werden in einem sogenannten Diskurs gesammelt. Der Diskurs ist eine Datenstruktur, die alle zu dem gerade aktiven Dialog gehörenden Informationen repräsentiert. Dazu gehören die erkannten Worte und Gesten des Benutzers, die Antworten des Roboters usw. Mittelfristig kommen hier weitere ausgeführte Handlungen des Roboters dazu, da jene den Kontext ebenso beeinflussen. Der Diskurs ist somit eine komponentenspezifische Form der Situationsinterpretation innerhalb des Dialogmanagers, da hier die Situation in Form eines abstrakten Dialogzustandes modelliert wird.

Die Funktionsweise des Dialogmanagers beruht auf der Annahme, dass jeder Dialog zu einem (u.U. zunächst dem Roboter unbekanntem) Ziel führen muss, beispielsweise dem Auffinden und Bezeichnen des richtigen Objektes, das ein Benutzer haben möchte. Auf dieser Grundlage sowie basierend auf vorgegebenen Regeln fällt der Dialogmanager eine Entscheidung über die nächste, auszuführende Aktion. Dabei betrachtet der Dialogmanager verschiedene Wissensquellen wie das Ziel des Dialogs, die schon existierenden Informationen zum Erreichen des Ziels ebenso wie die noch fehlenden Informationen.

Im oben beschriebenen Szenario löst jedes Sprachsignal eine Überprüfung durch den Dialogmanager aus. Dieser entscheidet, ob das Signal zu ignorieren ist (z.B. Fernseher), ob es sich um eine zur gegenwärtigen ausgeführten Handlung zugehörige Information handelt oder ob eine wesentliche Änderung der Situation aufgetreten ist. Wichtig ist hierbei, dass alle anderen gestarteten Aktionsfolgen zunächst parallel weiterlaufen. Bei Bedarf stößt der Dialogmanager aber als Ergebnis der Entscheidungsfindung eine zusätzliche Aktion an oder generiert ein Ereignis, das die aktuelle Aktionsfolge unterbricht oder abbricht. Beispiele hierfür sind Rückfragen an Berta oder die Aufforderung nach einer auszuführenden Zeigegeste ("Zeig mir bitte die Tasse!"). Ein Beispiel für einen Abbruch sind Ausrufe wie "Nein, nicht die!".

4 Planung- und Entscheidungsfindung für das Szenario

Bei der Auswahl einer Handlung müssen nun alle Informationen zusammengefasst werden. Dabei geht es immer um eine beabsichtigte *Änderung* einer Situation, ausgehend vom aktuellen Zustand des Roboters (z.B. derzeit verfügbare on-board-Ressourcen wie das Kamerasystem) und der Umgebung. Mit Bezug auf die Situationsbewertung ist der Roboter in der Lage, Absichten zum Herbeiführen gewünschter Situationen zu generieren. Das erfolgt über eine Kommunikation zwischen den Komponenten Situationsanalyse, -bewertung sowie Planung- und Entscheidungsfindung. Die Planung hängt folglich stark von der aktuellen Situation ab, wie das Szenario-Beispiel zeigt:

- Fall 1: Der Roboter redet bereits mit Berta und schaut sie an. Somit sind die Ressourcen Kopf+Hals und Sprachausgabe belegt.
- Fall 2: Der Roboter wartet im Raum ("IDLE-Modus").
- Fall 3: Der Roboter räumt gerade die Spülmaschine aus und stellt eine Tasse in den Schrank. Für die gegenwärtig aktivierte Handlung "Spülmaschine ausräumen" führt er derzeit eine detailliert vorgeplante Aktionsfolge (AF Spülmaschine) aus. Ein zugehöriges Petri-Netz als eine mögliche Implementierung zeigt Bild 3. Sein Aufmerksamkeitsfokus (engl. Focus of Attention FoA) wechselt zwischen Spülmaschine (A_1, A_4, A_6) und Schrank (A_5). Somit sind die Ressourcen Kopf+Hals, rechte Hand, rechter Arm, Torso und Plattform belegt. In der Aktionsfolge ist vorgeschrieben, dass nur die Ressource Kopf+Hals bei einer Unterbrechung verliehen werden darf.

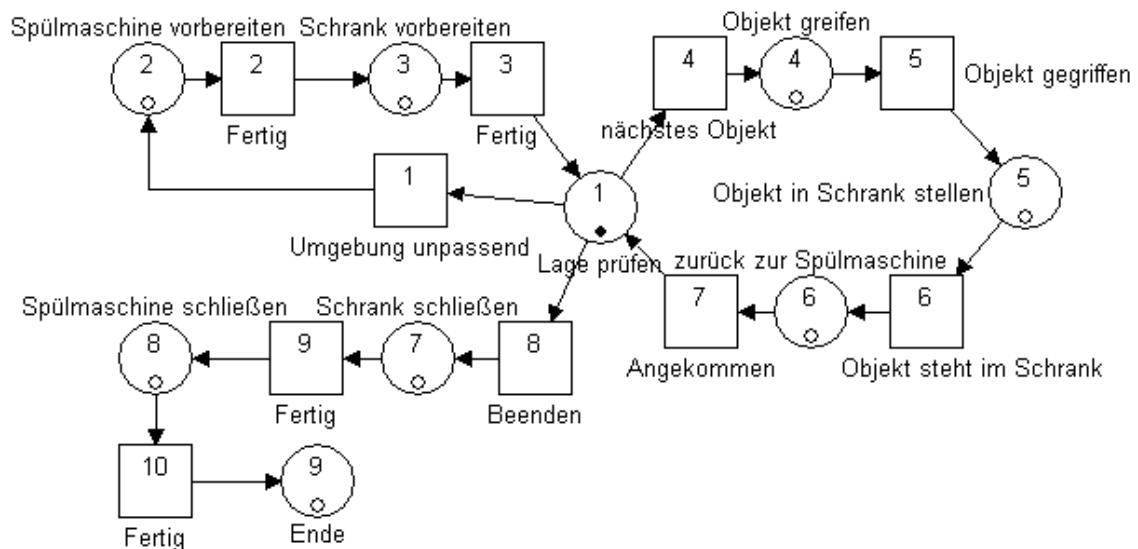


Bild 3: Aktionsfolge Spülmaschine mit Plätzen A_i (rund) und Transitionen T_i (quadratisch), Bild mit Netlab [21] gezeichnet

In allen drei Fällen kommt zum Zeitpunkt $k = 5$ ein Geräusch aus Richtung Berta, das durch die Perzeption erkannt und interpretiert wird. Das Geräusch wird als Sprache identifiziert und zum Dialogmanager weitergeleitet.

Während es sich bei Fall 1 um die Fortsetzung eines bereits begonnenen Dialogs handelt, ist die Situation im Fall 2 ein wenig komplizierter. Am Einfachsten ist hier eine feste Aktionsfolge, die u.a. das geräuschbasierte Initiieren von Handlungen vorsieht, z.B.

- Wenn der Roboter nichts hört, macht er nichts.
- Wenn er ein "interessantes Geräusch" bemerkt (was das ist, muss die Situationsanalyse entscheiden), richtet er seine visuelle Aufmerksamkeit durch Drehen seines Kopfes auf das Geräusch, sonst ignoriert er es. Hierzu wird die Ressource Kopf+Hals sowie u.U. Plattform und Torso (für eine Drehung) benötigt.
- Handelt es sich um Sprache und ist der Sprecher zu weit weg, gibt der Roboter eine kurze Bestätigung ("Moment"), fährt zur Geräuschquelle und bleibt in einem psychologisch angenehmen Abstand stehen. Dazu braucht der Roboter die Ressourcen Kopf+Hals, Plattform, Torso und Sprachausgabe.
- Nun wird ein Dialog eröffnet und u.U. nachgefragt, wenn der Roboter etwas nicht verstanden hat usw. Der Sprecher wird angeschaut, die Ressourcen bleiben reserviert.

Durch Anwenden dieser Aktionsfolge dreht sich der Roboter im Fall 2 zu Berta um, fährt u.U. ein Stück zu ihr und stellt somit die gleiche Situation wie in Fall 1 her.

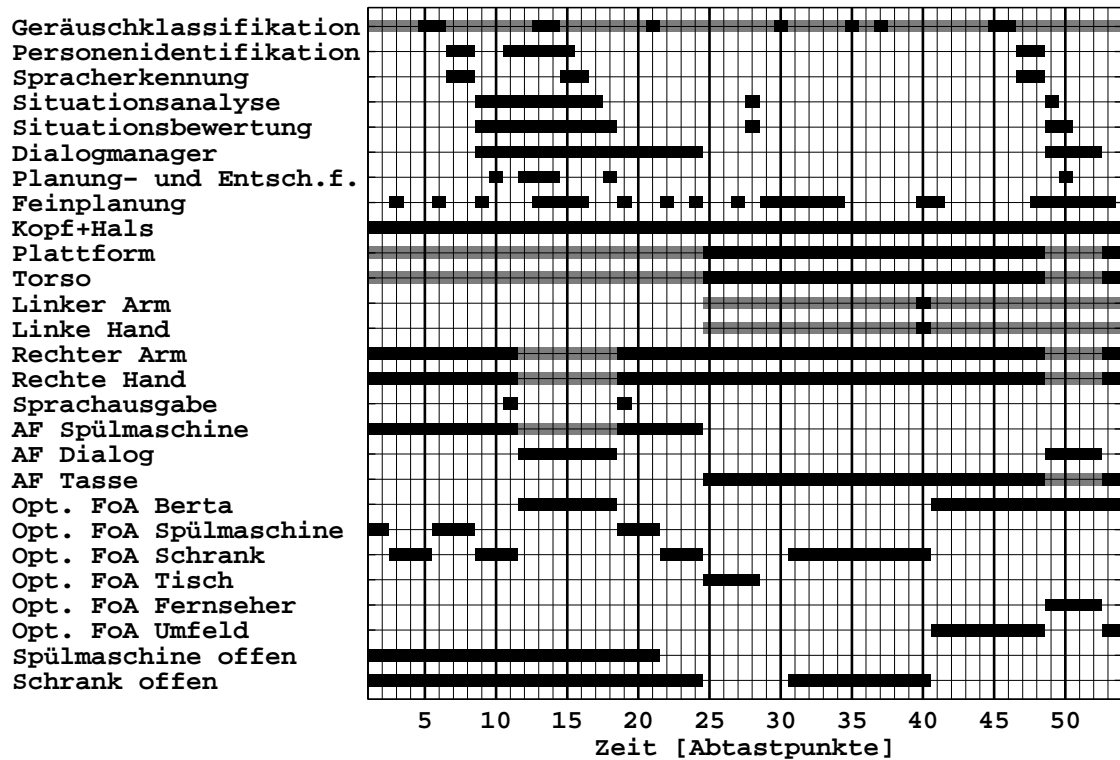


Bild 4: Zeitverlauf für aktive Komponenten, Ressourcen, Aktionsfolgen, Aufmerksamkeitsfokus und ausgewählte Umweltzustände für Fall 3 (schwarz: aktiv, grau: Stand-by, weiß: inaktiv)

Noch deutlich komplexer ist die Situation für Fall 3. Bild 4 zeigt einen manuell generierten Zeitverlauf für aktive Komponenten, Ressourcen, Aktionsfolgen, den Aufmerksamkeitsfokus sowie für ausgewählte Umweltzustände, um den prinzipiellen Ablauf zu verdeutlichen. Zukünftig ist eine schrittweise simulative und experimentelle Evaluierung der Abläufe geplant.

Der Roboter hat bei $k = 5$ eine Tasse in der Hand (A_4 der AF Spülmaschine) und soll natürlich nicht zum Geräusch schauen, während er sie in den Schrank einräumt. Natürlich kann der Roboter aber zuhören und registrieren, dass Berta mit ihm reden möchte. Das erfasst er zum Zeitpunkt, zu dem er erkennt, dass Berta ihn mit seinem Namen angesprochen hat ($k = 9$). Wenn die Situationsbewertung nun Berta hinreichend Priorität zuweist, bietet sich folgende sinnvolle Reaktion an:

- Berta eine kurze sprachliche Bestätigung geben (das "Moment" aus Fall 2, $k = 11$).
- Der Aktionsfolge Spülmaschine den Wunsch nach einer sicheren zeitweiligen Unterbrechung schicken. Der kann z.B. so realisiert werden, dass die Aktionen "Objekt in Schrank stellen" A_5 und "zurück zur Spülmaschine" A_6 auch bei einer Unterbrechung noch fortgesetzt werden, um einen sicheren Zustand zu erzielen [13].
- Die eigentliche Unterbrechung erfolgt erst, wenn die nächste Tasse im Schrank ist und die Hand wieder über der Spülmaschine A_1 ($k = 11$). Das Wissen über eine sichere Unterbrechung gehört hier also lokal zur Aktionsfolge!
- Nun kann er sich zu Berta umdrehen (FoA Berta) und sie ansprechen, um sich die Aufgabe erklären zu lassen ($k = 12$). Berta sagt, dass sie einen Kaffee möchte ($k = 13$),

was wiederum eine Auswertung durch Geräuschklassifikation und Spracherkennung nach sich zieht. Der Roboter bestätigt, dass er den Wunsch verstanden hat ($k = 19$).

Ab hier ist der Roboter jetzt bei allen drei Fällen (fast) in der gleichen Aktionsfolge, eine Besonderheit für Fall 3 folgt aber noch später.

Nun sind die Situationserkennung und die Planung und Entscheidungsfindung gefordert, um zu verstehen, dass es Kaffee aus Tassen gibt, Berta die rote bevorzugt und diese geholt werden soll. Der Roboter verfügt über eine Handlung "Objekt holen" mit einer assoziierten Aktionsfolge, die jetzt spezifiziert werden ("Hole Berta ihre rote Tasse mit Kaffee", "AF Tasse"). Die Aktionsfolge benötigt Ressourcen, die derzeit durch die AF Spülmaschine blockiert und nicht ausleihbar sind. Die AF Spülmaschine ist folglich zu beenden.

Allerdings sind in Fall 3 Schrank und Spülmaschine noch offen und somit in einem unsicheren Zustand. Wenn das Ereignis "Beenden" (T_8) geschickt wird, muss der Roboter ab $k = 19$ noch Schrank und Spülmaschine schließen, um einen sicheren Zustand zu erreichen. Erst dann gibt die AF die Ressourcen Kopf+Hals, Plattform, Arm und Hand frei ($k = 24$, A_9 der AF Spülmaschine). Außerdem ist sinnvoll, die halb ausgeräumte Spülmaschine noch in der Situationsanalyse als unfertig zu markieren, um später fortzusetzen.

Nun wird die AF Tasse gestartet. Um in einer ersten vorgeplanten Aktion herauszufinden, wo sich die rote Tasse befindet, muss der Roboter den Kopf bewegen und gegebenenfalls in der Küche umher fahren. Besser ist es natürlich, wenn der Standort der Tasse im aktuellen Szenengraphen im Arbeitsgedächtnis oder mit ihrem üblichen Platz im Langzeitgedächtnis steht: normalerweise entweder sauber in einem anderen Schrank oder benutzt auf dem Tisch oder in der Spülmaschine. In der Aktionsfolge ist vorgesehen, zunächst nacheinander mögliche Positionen des Objektes zu prüfen und evtl. ein anderes Objekt zu verwenden, wenn das ursprünglich geplante Objekt nicht gefunden wird oder ungeeignet ist. Das erforderliche Abstraktionsniveau für solche Aktionen ist natürlich relativ hoch und erfordert eine geschickte Parametrierung aus dem Gedächtnis.

Der Roboter dreht den Kopf zum Tisch ($k = 25$), um den 1. Ort zu prüfen, und findet die Tasse. Ansonsten würde er als nächstes zum Schrank (2. Ort) fahren und den Standort der Tasse prüfen. Die Situationsanalyse schlussfolgert, dass die rote Tasse auf dem Tisch vermutlich benutzt und somit für die Aufgabe uninteressant ist. Die Feinplanung beschließt, als Ersatzobjekt eine andere frische Tasse zu verwenden, die nun wiederum zu lokalisieren ist. Der 1. Ort ist der Schrank, der dazu geöffnet wird ($k = 31$).

Der Roboter sucht eine geeignete Tasse im Schrank und ruft einen Greifweg, d.h. einen Plan zum richtigen Greifen des Objektes, aus dem Langzeitgedächtnis ab und instantiiert ihn. Die Komponenten der taktilen Wahrnehmung an den Roboterfingern sowie die Auswertung des Kraft-Momenten-Sensors im Handgelenk und der Positionssensoren in den Fingern geben dem Roboter Rückmeldung darüber, ob sich die Tasse bereits in seiner Hand befindet. Ist die Tasse gegriffen, schließt der Roboter mit der freien linken Hand die Tür ($k = 40$) und ändert wieder seinen optischen Aufmerksamkeitsfokus.

Nachdem der Roboter sich dann auf den Weg zur Kaffeemaschine begibt ($k = 41$), nimmt er ein weiteres Geräusch wahr, das sich als Sprachsignal erweist ($k = 45$). Im Gegensatz zu vorherigen als irrelevant eingestuften Geräuschen (z.B. $k = 21, 30, 35$) fordert die Situationsanalyse eine sichere Unterbrechung der AF Tasse an, um die Signalquelle optisch aufzuklären. Auch hier wird die Ressource Kopf+Hals ausgeborgt und der Aufmerksamkeitsfokus auf das Geräusch gerichtet ($k = 49$). Da sich das Signal als irrelevantes Signal aus Richtung Fernsehen erweist, wird die ursprüngliche Handlung fortgesetzt ($k = 53$).

An dieser Stelle soll auf eine detaillierte Erläuterung des weiteren Vorgehens verzichtet werden. Auch im weiteren Verlauf bleiben aber viele kognitive Herausforderungen:

- Gibt Berta beispielsweise den zusätzlichen Befehl, noch einen Teller mitzubringen, so muss nach der Situationsanalyse zwischen Planung und Situationsbewertung abgestimmt werden, ob es möglich ist, beide Aufgaben parallel auszuführen. Der Roboter verfügt noch über eine zweite Hand als Ressource, so dass die parallele Ausführung zweier Aktionsfolgen theoretisch möglich ist.
- Falls der Roboter etwas nicht findet, kann er evtl. Berta aktiv fragen. Das kann entweder fest in der Aktionsfolge programmiert sein oder neu geplant werden. Kann der neue Dialog den Aufenthaltsort des Objektes klären, steigt die Erfolgsquote für die fortzusetzende Handlung im Vergleich zu einem ziellosen Suchprozess.
- Wenn Absichten des Menschen unklar bleiben, können proaktive Komponenten einbezogen werden [22].
- Die Zielposition der Tasse kann nur während der Ausführung der gestellten Aufgabe festgelegt werden. Dies stellt wiederum die globale Situationsanalyse fest, die lokale Situationserkennung richtet ihre Aufmerksamkeit auf die Roboterhand, die Zielposition und die Objektübergabe bzw. auf das Abstellen des Objektes.

5 Diskussion

Die bisherigen Ausführungen zeigen, dass der Weg zu einer kognitiven Architektur für humanoide Roboter noch lang ist. Die hier präferierte Variante bevorzugt bei der Koordinierung und Ausführung einen eher vorgeplanten Weg, der sich an der pragmatischen Strategie aus Abschnitt 3.1 orientiert und lediglich einige strategische Entscheidungen auf der obersten Ebene durch Bewertungen sowie Assoziationen zwischen Situationen und Handlungen löst.

Eine mögliche Methode für die lerndominierte Strategie sind rekurrente neuronale Netze, die einen universellen Ansatz zur Verbindung der verschiedenen Informationsflüsse (z.B. Ziele, Perzeption, Aktionen usw.) bieten und somit den erforderlichen Programmieraufwand bei Implementierung des Handlungswissens drastisch reduzieren können [9, 17, 23]. Leider konnte diese Strategie bisher nur in vereinfachten Umgebungen erfolgreich realisiert werden. Das Hauptproblem bei komplexen Aufgaben besteht darin, dass auch die Komplexität der Lernvorgänge zu groß wird und Fehlentscheidungen bei der Ausführung von Handlungen zu Sicherheitsproblemen führen können. Aus der Robotik ist bekannt, dass Ergebnisse aus robotergerechten vereinfachten Umgebungen nur schwer auf reale Umgebungen übertragbar sind. Eine zu starke Orientierung an der lerndominierten Strategie wird deshalb von den Autoren derzeit eher skeptisch gesehen.

Deswegen ist es wichtig, sich auch bei frühen Implementierungen an konkreten Anwendungsdomänen und -szenarien zu orientieren. Das wird bei der vorgeschlagenen Lösung durch die intensiv getesteten pragmatischen Teillösungen erreicht, die dann in Aktionsfolgen mit freien Parametern abgelegt werden. Somit können viele sinnvolle Aktionen des Roboters bereits fest programmiert werden. Wenn beispielsweise der Roboter den Menschen beim Sprechen und Zuhören anschaut, empfindet der Mensch das Kommunikationsverhalten als angenehmer. Eine feste Programmierung dieses angestrebten Verhaltens reduziert den gesamten Suchraum der Lernvorgänge und somit deren Komplexität.

Andererseits ist es sinnvoll, bestimmte eher "kleinteilige" Lernvorgänge gezielt einzubetten, wie z.B. für Assoziationen zwischen Situationen und Handlungen oder für Erfolgs-

aussichten für Handlungen. Somit können die Informationen in einem bereits vorstrukturierten Langzeitgedächtnis schrittweise adaptiert werden. Außerdem sind bestimmte Lernvorgänge (z.B. für einzelne Wörter oder Objekte) gezielt zu starten, wenn dem Roboter wichtige Informationen fehlen.

Der Detailliertheitsgrad der Pläne kann sich von Aktionsfolge zu Aktionsfolge unterscheiden und u.U. auch je nach Situation variieren. So können für bekannte Situationen fein ausgearbeitete und weit in die Zukunft reichende Pläne existieren, in unbekanntem und schwer prognostizierbaren Situationen ist hingegen eine feingliedrigere Planung sinnvoll.

Allerdings verbleiben viele offene Fragen, z.B.

- Wie gelingt es, eine ausreichend große Menge an Operatoren und Aktionsfolgen aufzubauen, freie Parameter vorzusehen und diese in der Feinplanung zu spezifizieren?
- Wie muss ein geeignetes quantitatives Bewertungssystem als Voraussetzung für eine Priorisierung von Handlungen aussehen?
- Welche Mechanismen zum Ausborgen von Ressourcen sind sinnvoll?
- Wie sehen geeignete Gedächtnisstrukturen für einen aktiven Zugriff auf das Langzeit- und Arbeitsgedächtnis aus?
- Welchen Grad an nicht geplanten Situationen verarbeitet der Roboter mit dieser Strategie bei einer guten Qualität der gefundenen Pläne?

6 Zusammenfassung

Der Beitrag stellt neue konzeptionelle Überlegungen für eine integrierte und in absehbarer Zeit implementierbare kognitive Architektur für humanoide Roboter vor, diskutiert notwendige Komponenten und Informationsflüsse sowie ein Gedankenexperiment für die Planung komplexer Handlungen anhand eines konkreten Anwendungsszenarios in einer häuslichen Umgebung. Die Grundidee der vorgeschlagenen Strategie besteht darin, möglichst große Teile des Handlungswissens modular aufzubauen und nur durch die obere Ebene einer kognitiven Architektur auswählen und parametrieren zu lassen. Damit verbindet sich die Erwartung, die Planungskomplexität auf ein implementierbares Maß zu reduzieren. Die zugehörigen Komponenten werden erläutert und ihre wesentlichen Informationsflüsse anhand des konkreten Szenarios skizziert. Zudem werden Strategieelemente für die Handlungsbewertung als Basis für Lernverfahren vorgeschlagen und offene Fragen diskutiert. Die Beschränkung auf ein Gedankenexperiment resultiert daraus, dass wesentliche Teile der Strategie bislang noch nicht implementiert sind. Das Experiment soll dazu dienen, eine Diskussionsbasis für einen nachfolgenden strukturierten Entwurf der Architektur zu schaffen und frühzeitig Hinweise zu generieren.

Danksagung: Diese Arbeit wird von der Deutschen Forschungsgemeinschaft (DFG) im Rahmen des Sonderforschungsbereiches (SFB) 588 "Humanoide Roboter - Lernende und kooperierende multimodale Roboter" gefördert. Die Autoren danken allen Mitgliedern der AG "Kognitive Architektur" dieses SFBs, insbesondere Tamim Asfour, Rüdiger Dillmann, Hartwig Holzapfel, Arne Lehmann, Peter Steinhaus und Rainer Stiefelhagen, für viele konstruktive und kontroverse Diskussionen, die dieses Konzept maßgeblich beeinflusst haben. Ein weiterer Dank gilt der AG "Demonstrator" des gleichen SFBs, deren Diskussionen immer wieder die praktischen Schwierigkeiten und die große Komplexität von realen Roboteraufgaben vor Augen führten. Außerdem waren die Bücher von Dietrich Dörner eine wichtige Inspirationsquelle.

Literatur

- [1] Burghart, C.; Mikut, R.; Holzapfel, H.: Cognition-Oriented Building Blocks of Future Benchmark Scenarios for Humanoid Home Robots. In: *Proc., AAAI Workshop Evaluating Architectures for Intelligence, Vancouver*. 2007.
- [2] Sloman, A.; Wyatt, J.; Hawes, N.; Chappell, J.; Kruijff, G.: Long Term Requirements for Cognitive Robotics. In: *Proc., AAAI Workshop on Cognitive Robotics*, S. 143–150. Boston. 2006.
- [3] Burghart, C.; Mikut, R.; Asfour, T.; Schmid, A.; Kraft, F.; Schrempf, O.; Holzapfel, H.; Stiefelhagen, R.; Swerdlow, A.; Bretthauer, G.; Dillmann, R.: Kognitive Architekturen für humanoide Roboter: Anforderungen, Überblick und Vergleich. In: *Proc., 17. Workshop Computational Intelligence*. Universitätsverlag Karlsruhe. Eingereichter Beitrag. 2007.
- [4] Byrne, M.: Cognitive Architecture. In: *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications* (Jacko, J. A.; Sears, A., Hg.), S. 97–117. Lawrence Erlbaum Associates. 2003.
- [5] Langley, P.; Laird, J. E.; Rogers, S.: Cognitive Architectures: Research Issues and Challenges. Techn. Ber., Stanford University, Computational Learning Laboratory (CSLI). 2006.
- [6] Asfour, T.; Regenstein, K.; Azad, P.; Schröder, J.; Dillmann, R.: ARMAR-III: A Humanoid Platform for Perception-Action Integration. In: *Proc., HCRS, Munich*, S. 51–56. 2006.
- [7] Connell, J.: SSS: A Hybrid Architecture Applied to Robot Navigation. In: *Proc., IEEE Conference on Robotics and Automation (ICRA)*, S. 2719–2724. 1992.
- [8] Schaal, S.: Is Imitation Learning the Route to Humanoid Robots? *Trends in Cognitive Science* 3(6) (1999), S. 233–242.
- [9] Ritter, H.; Steil, J.; Nölker, C.; Röthling, F.; McGuire, P.: Neural Architectures for Robot Intelligence. *Reviews in Neuroscience* 14 (1-2) (2003), S. 121–143.
- [10] Stiefelhagen, R.; Ekenel; Fügen, C.; Giesemann, P.; Holzapfel, H.; Kraft, F.; Nickel, K.; Voit, M.; Waibel, A.: Enabling Multimodal Human-Robot Interaction For the Karlsruhe Humanoid Robot. *IEEE Transactions on Robotics* 23(5) (2007).
- [11] Asfour, T.; Ly, D.; Regenstein, K.; Dillmann, R.: Coordinated Task Execution for Humanoid Robots. In: *Proc., 9th International Symposium on Experimental Robotics 2004 (ISER 04)*. 2004.
- [12] Lehmann, A.; Mikut, R.; Asfour, T.: Petri Nets for Task Supervision in Humanoid Robots. In: *Proc., 37th International Symposium on Robotics (ISR 2006), München*, S. 71–73. 2006.
- [13] Lehmann, A.; Mikut, R.; Asfour, T.: Petri-Netze zur Aufgabenüberwachung in humanoiden Robotern. In: *Proc., 15. Workshop Computational Intelligence*, S. 157–171. Universitätsverlag Karlsruhe. 2005.
- [14] Holzapfel, H.; Nickel, K.; Stiefelhagen, R.: Implementation and Evaluation of a Constraint Based Multimodal Fusion System for Speech and 3D Pointing Gestures. In: *Proc., International Conference on Multimodal Interfaces*. 2004.
- [15] Asimov, I.: *Robots and Empire*. London: Grafton Books. 1985.
- [16] Clarke, R.: Asimov's Laws of Robotics Implications for Information Technology. *IEEE Computer* 26-27 (1993-4), S. 53–61 (Part 1), 57–66 (Part 2).
- [17] Dörner, D.: *Bauplan für eine Seele*. Rowohlt. 2002.
- [18] Kieras, D.; Meyer, D.: Predicting Performance in Dual-Task Tracking and Decision Making with EPIC Computational Models. In: *Proc., 1st Int. Symp. on Command and Control Research and Technology*, S. 314–325. 1995.
- [19] Dillmann, R.: Teaching and Learning of Robot Tasks Via Observation of Human Performance. *Robotics and Autonomous Systems* 47 (2-3) (2004), S. 109–116.
- [20] Azad, P.; Asfour, T.; Dillmann, R.: Combining Appearance-Based and Model-Based Methods for Real-Time Object Recognition and 6D Localization. In: *Proc., IROS, Peking*. 2006.
- [21] Orth, P.; Abel, D.: Rapid Control Prototyping petrinetzbasierter Steuerungen mit dem Tool NETLAB. *at - Automatisierungstechnik* 54 (2006), S. 222–227.
- [22] Schrempf, O.; Hanebeck, U.; Schmid, A.; Wörn, H.: A Novel Approach to Proactive Human-Robot Cooperation. *Proc., IEEE Int. Workshop on Robot and Human Interactive Communication* (2005), S. 555–560.
- [23] Porr, B.; Wörgötter, F.: Isotropic-Sequence-Order Learning in a Closed-Loop Behavioural System. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences* 361 (2003) 1811, S. 2225–2244.

Diskret-kontinuierliche Regelung und Überwachung von Robotern basierend auf Aktionsprimitiven und Petri-Netzen

Giulio Milighetti, Helge-Björn Kuntze

Fraunhofer-Institut IITB

Fraunhoferstraße 1, 76131 Karlsruhe

Tel.: +49-721-6091- $\{502, 310\}$

Fax: +49-721-6091-413

E-Mail: $\{milighetti, kuntze\}@iitb.fraunhofer.de$

1. Einleitung

Gegenwärtig werden im Bereich der Robotik starke Anstrengungen unternommen, um eine neue Generation von sog. humanoiden Robotern zu entwickeln, die in der Lage sind, anspruchsvolle Aufgabenstellungen im häuslichen Umfeld weitgehend autonom sowie auch interaktiv mit dem Menschen zu erfüllen. Um den höheren Anforderungen hinsichtlich Flexibilität und Autonomie in einer weitgehend unstrukturierten Umgebung mit ständig ändernden Umwelteinflüssen und Aufgabenstellungen gerecht werden zu können, benötigt der Roboter menschenähnliche Fähigkeiten. Diese werden einerseits durch entsprechende Sensoren erreicht, die das komplette Spektrum der menschlichen Wahrnehmungsmöglichkeiten abdecken, um Informationen über den internen Zustand und die Umgebung zu sammeln. Genauso wichtig ist aber eine Intelligenz, um die komplementären Messdaten zu kombinieren, auszuwerten und anhand der Ergebnisse zielführende Handlungen planen zu können. Schließlich ist noch eine geeignete Steuerungs- und Regelungsarchitektur erforderlich, die erheblich flexibler hinsichtlich wechselnder Aufgabenstellungen ist, als bei marktüblichen Industrierobotern.

Um der Komplexität solch anspruchsvoller Aufgaben begegnen zu können, muss dem Roboter ein situationsabhängiger Entscheidungsfindungsalgorithmus, d. h. eine ereignisdiskrete Regelung zur Verfügung stehen, die den komplexen Handlungsplan in eine dynamisch konfigurierbare Folge sog. Aktionsprimitive zerlegt, die dann unter Verwendung aktionsspezifischer kontinuierlicher Regelungsalgorithmen ausgeführt werden. Die Entwicklung und Optimierung einer solchen hybriden diskret-kontinuierlichen Regelung basiert auf geeigneten diskreten Modellen, die durch Petri-Netze oder Automaten beschrieben werden [1], [14]. Das klassischste und am weitesten verbreitete Anwendungsgebiet solcher Strukturen stellt die industrielle Produktionssteuerung dar. Verschiedene Anwendungen von Petri-Netzen bei der Regelung industrieller Prozesse kann zum Beispiel in [5], [6], [19] und [23] gefunden werden.

Allerdings werden Petri-Netze auch verstärkt im Bereich der Robotik, zur Modellierung und Planung komplexer Aufgaben [3], [4] als auch zu deren Überwachung und Regelung [14] eingesetzt. Viele Anwendungen finden sich zum Beispiel im Gebiet der unbemannten Fahrzeuge [18] und der humanoiden Roboter [13]. Überdies ist eine auf Petri-Netzen basierende Aufgabenstruktur auch kompatibel mit existierenden und noch laufenden Forschungsaktivitäten im Rahmen des gemeinschaftlichen Sonderforschungsbereiches SFB 588 „Humanoide Roboter“ [2].

Um autonome Roboter mit aufgabenspezifischer Intelligenz auszustatten, wurden in der Vergangenheit verschiedene Entscheidungsfindungsalgorithmen vorgeschlagen [8], [9], meist in der Form von reaktiver Planung in der mobilen Robotik. Die meisten von ihnen basieren auf einer dynamischen reaktiven Planung der Roboteraktionen. Hierfür finden überwiegend lernfähige Neuro-Fuzzy-Methoden oder probabilistische Bayes-Verfahren erfolgreich Anwendung [7], [11], [15], [19], [20].

Für das am Fraunhofer IITB entwickelte, auf Aktionsprimitiven basierende diskret-kontinuierliche Regelungskonzept [16], wird in diesem Beitrag ein neuer Fuzzy-basierter Entscheidungsfindungsalgorithmus vorgestellt, welcher die Eigenständigkeit und Flexibilität des Roboters beträchtlich verbessert. Im Falle von Störeinflüssen (z. B. Rauschen oder Fehlfunktionen von Sensoren) oder unvorhergesehenen Ereignissen (z. B. Hindernissen), ist der Roboter in der Lage dynamisch zu reagieren, indem er den Ablauf seiner elementaren Aktionen anpasst. Die Effizienz des vorgestellten Konzeptes wird in einem ersten Experiment demonstriert, einer Greifaufgabe, bei der verschiedene optische und kraftmessende Sensoren zur Anwendung kommen.

2. Das Regelkonzept

Roboterplattform



IITB-Plattform



SFB-Plattform



Abb. 1: Multisensorielle Versuchsplattform

Für die Entwicklung und Untersuchung des diskret-kontinuierlichen Überwachungs- und Regelungskonzeptes steht am Fraunhofer IITB eine Versuchsplattform zur Verfügung, die aus zwei 7DoF AMTEC-Roboterarmen, einem 2DoF Pan-Tilt-Sensorkopf und einer Fluidhand besteht. Um eine Vielzahl an Grundfähigkeiten bewältigen zu können, ist der Roboter mit mehreren redundanten und sich ergänzenden Sensoren und Aktoren ausgestattet (Abb. 1).

Der Kopf verfügt über eine Stereokamera, welche vordefinierte Objekte verfolgen kann und einen akustischen Sensor (Mikrophon-Array), der in der Lage ist, Geräuschquellen zu lokalisieren. Des Weiteren ist eine miniaturisierte Kamera zur präzisen Lokalisierung von Objekten in die Handinnenfläche integriert. Für die taktile Tastprüfung sind zwei Kraft-Momenten-Sensoren an den Handgelenken angebracht, außerdem kann ein

Schlupfsensor in einen der Greifer eingebaut werden, um Relativbewegungen zwischen dem Endeffektor und Oberflächen zu detektieren. Die auf die Entwicklungsplattform implementierten und erprobten Ergebnisse können unter Zuhilfenahme von MCA-Modulen [24] auf den gemeinsamen Demonstrator ARMAR [25] des SFB 588 transferiert werden.

Reglerarchitektur

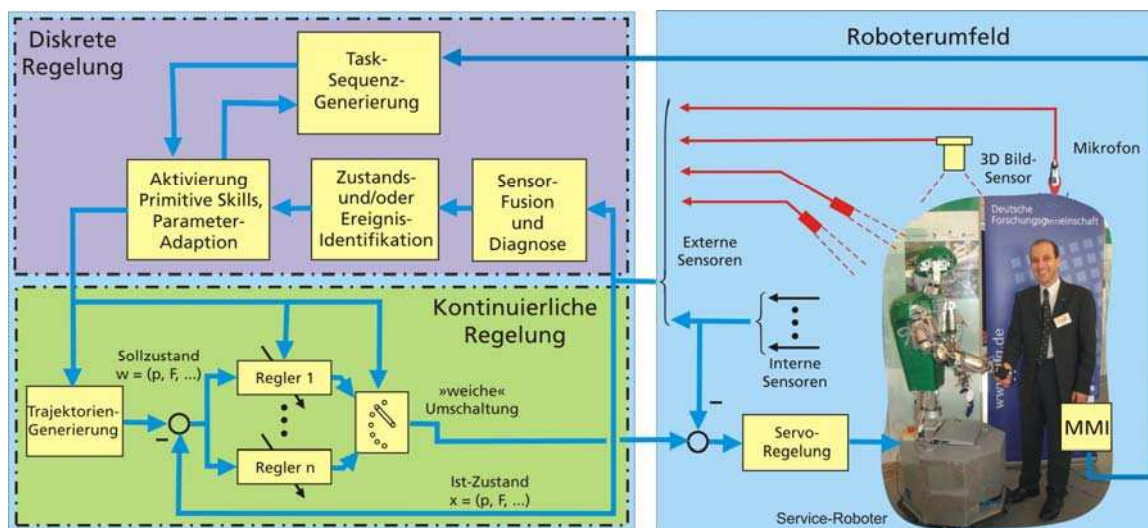


Abb. 2: Multisensorielle, diskret-kontinuierliche Reglerarchitektur

Die Struktur, welche dem Regelkonzept zugrunde liegt, das zur Überwachung des Roboters während der vollständigen Ausführung seiner Aufgabe entwickelt wurde, ist in Abb. 2 dargestellt. Der Zustand des Roboters, der während der Ausführung seiner Aufgabe mit einem menschlichen Operator und dem Zustand der Umgebung interagiert, wird mithilfe interner (z. B. Encoder) und externer Sensoren (z. B. Kamera, Mikrofon, Kraft-Momenten-Sensor) überwacht.

Auf der höheren Ebene der Reglerhierarchie verarbeitet eine diskrete Regelung die Sensormessungen und trägt diese zusammen, um Diagnosesignale zu erstellen, welche quantitative Informationen über den kontinuierlichen Zustand des Systems enthalten (z. B. Positionen von Objekten, Geräusche, Kräfte). In einem zweiten Schritt wird diese Information zur Erkennung des diskreten Zustands oder Ereignisses benutzt, wodurch qualitative Informationen über die gegenwärtige Situation erstellt werden (z. B. ein bestimmtes Ereignis ist aufgetreten). Aufgrund von Interpretationen der gesammelten Erkenntnisse über den kontinuierlichen und diskreten Zustand des Systems überwacht eine Entscheidungseinheit die ablaufende Handlungssequenz und, wenn dies nötig ist um eine unvorhergesehene Situation zu bewältigen und das ursprünglich angestrebte Ziel zu erreichen, gestaltet sie dynamisch eine neue Aktionsfolge oder passt die bisherige an.

Mit dem Ziel, eine flexible Generierung von Handlungssequenzen durchzuführen und gleichzeitig eine schnelle Online-Modifikation der Sequenz zu ermöglichen, wurde eine diskrete Aufgabenstruktur entwickelt, die ein transparentes und effizientes Bearbeiten von Online-Entscheidungen gewährleistet und auf die außerdem unkompliziert durch die Entscheidungseinheit zugegriffen werden kann. Eine Möglichkeit, solch eine Aufgabenstruktur zu realisieren besteht darin, ihre Architektur auf einer Sequenz von Modulen aufzubauen. Jede Handlung kann dann als das Resultat der Ausführung einer Kette von elementaren Aktionen angesehen werden, von denen jede ein eigenes

Unterziel besitzt [21]. Solche Aktionen werden Aktionsprimitive (engl. Primitive Skills, PS) genannt.

Hat die Entscheidungseinheit erst die auszuführende PS-Sequenz bestimmt und den geeignetsten Regler für das gerade auszuführende PS gefunden, dann stellt die kontinuierliche Regelung in der unteren Ebene der Hierarchie sicher, dass das gerade aktive PS von diesem optimalen und spezifischen Regler gesteuert wird.

Aktionsprimitive und Petri-Netze

Ein Aktionsprimitiv wird in [21] definiert als das Tupel

$$PS = \{HM, \tau, \lambda\} \quad (1)$$

Die Variable *HM* (Hybrid Motion) enthält alle Informationen, welche für die Trajektoriengenerierung und die Regelung der Bewegung (z. B. Sollwerte, aktive Regler und verwendete Parameter, usw.) zum Erreichen des PS-Zieles benötigt werden. Die an der Ausführung des PS beteiligten Ressourcen (z. B. Sensoren, Werkzeuge, Objekte, etc.) und die zugehörigen Befehle (z. B. Activate, Open, etc.) sind in der Variablen τ gespeichert. Die letzte Variable λ repräsentiert den Endzustand des PS, der erreicht werden muss, um das PS als erfüllt anzusehen und das nächste auszuführen („Position erreicht“, „Fehlfunktion“, usw.). Nähere theoretische Einzelheiten über diese drei Variablen finden sich in [21].

Das Konzept wurde in [16] um zwei Variablen erweitert und führt zu

$$PS = \{HM, \tau, \lambda, E, a\} \quad (2)$$

Die Affinität *a* bewertet die A-priori-Eignung des PS, ein gegebenes Ziel zu erreichen. Werden zwei unterschiedliche PS verglichen, die beide in der Lage wären dieselbe Aktion durchzuführen, so beschreibt der Affinitätswert, wie optimal das eine im Vergleich zum anderen dazu geeignet ist (z.B. schneller, genauer). So lässt sich zum Beispiel eine Lokalisierung mit einem optischen Sensor schneller durchführen als mit einem taktilen. Die Affinität ist zeitunabhängig, sie beruht lediglich auf A-priori-Kenntnissen der Leistungsmerkmale jeder PS, ohne die tatsächlich vorherrschende Situation zu berücksichtigen.

Die Effizienz *E* misst die momentane Leistungsfähigkeit des PS, indem sie analysiert, wie gut die zur Ausführung benötigten Ressourcen im Bezug auf den aktuellen Zustand des Systems funktionieren. Sie kann wie folgt in ihre Komponenten zerlegt werden:

$$E(t) = \prod_{i=1}^n (av_i(t) \cdot q_i(t) \cdot r_i(t)) + \sum_{j=1}^m (av_j(t) \cdot q_j(t) \cdot r_j(t)) / m \quad (3)$$

$$av, q, r, E(t) \in [0,1]$$

Dabei stellt av_i die Verfügbarkeit der i-ten Ressource dar (1→verfügbar, 0→nicht verfügbar), q_i seine Qualität (1→funktioniert gut, 0→funktioniert nicht) und r_i die Zuverlässigkeit dieser beiden Bewertungen (1→zuverlässig, 0→nicht zuverlässig). Um die gesamte Effizienz des PS zu erhalten, werden die Effizienzen der *n* daran beteiligten und zusammenwirkenden Ressource multipliziert, die der ergänzenden Ressource dazuaddiert.

Jedem PS kann ein diskreter Zustand des Systems zugewiesen werden, wodurch wiederum jedes PS durch eine Stelle in einem die Handlungsaufgabe beschreibenden Petri-Netz ausgedrückt werden kann. Um einen Konflikt in den Netzen zu lösen (wenn mehr als eine Transition zum gleichen Zeitpunkt feuerbereit ist), muss eine Entscheidung über die am ehesten zu verfolgende Strategie getroffen werden.

Wie eine solche PS-Struktur genutzt werden kann, um einen flexiblen, entscheidungsfindenden Algorithmus zu implementieren, soll im folgenden Abschnitt gezeigt werden.

3. Entscheidungsfindungsalgorithmus

Der Entscheidungsfindungsalgorithmus kann als diskrete Regelvorschrift des Robotersystems verstanden werden. Tatsächlich wird bei jedem Zeitschritt der aktuelle diskrete Zustand mit dem Sollzustand verglichen und korrigiert, falls er für die aktuelle Situation nicht optimal ist.

Der diskrete Sollwert ist jener Zustand, der die beste Leistung gewährleistet, nämlich gerade das PS mit der größten Effizienz (das PS, dessen Ressource momentan am besten funktioniert) und mit der höchsten Affinität (die A-priori beste Wahl). Die Entscheidung, die den lokalen Konflikt im Netz löst, fällt somit auf das PS mit dem höchsten Wert der Nutzenfunktion, die gegeben ist durch das Produkt $E \cdot a$.

$$\max(E) \wedge \max(a) \Rightarrow \max(E \cdot a) \quad (4)$$

Zu jedem Zeitschritt können parallel die Effizienzen in Abhängigkeit der aktuellen Informationen über die Ressourcen aller PS aktualisiert und daraufhin genutzt werden, um die optimale Entscheidung zu treffen.

Mit diesem Ansatz wird eine dezentrale Entscheidungsfindungsstruktur erreicht, die auf lokalen Größen beruht, welche den PS, unabhängig von seiner Position im Netz, zugewiesen sind, und die somit nicht von speziellen Netzkonfigurationen oder -konflikten beeinflusst wird (Abb. 3). Auf diese Weise können die Folgen einer erhöhten Komplexität des Entscheidungsfindungsalgorithmus bei einer höheren Anzahl möglicher Entscheidungen vernachlässigt werden. Die Tatsache, dass der größte Teil der für die Entscheidung benötigten Intelligenz lokal in den einzelnen PS gespeichert ist, erlaubt es dem Algorithmus außerdem, auch im Falle von selbstgenerierten Aktionsfolgen automatisch abzulaufen.

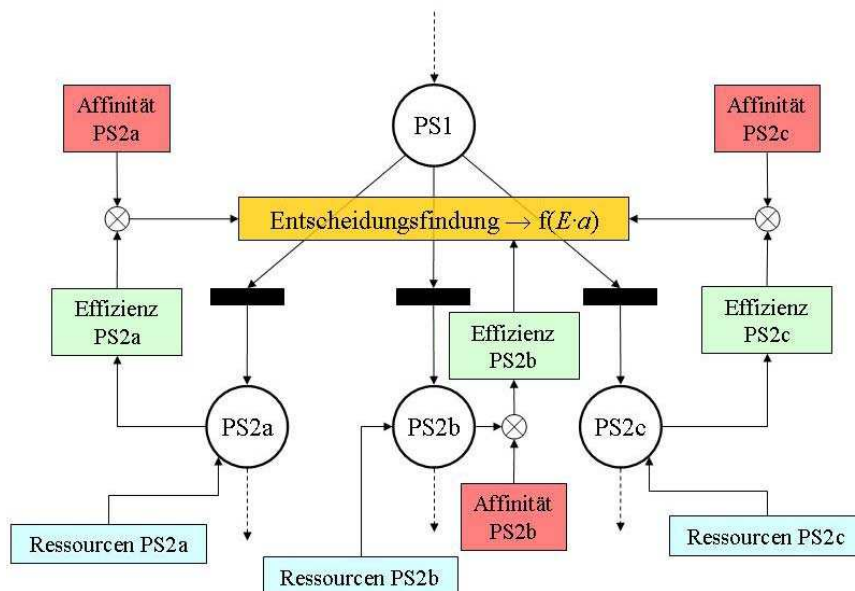


Abb. 3: Dezentralisierte PS-basierte Entscheidungsfindungsstruktur

Da nach diesem Ansatz jeweils nur der nächste Schritt zur Ausführung des Plans betrachtet wird, wird hier nur ein lokales Optimierungsproblem gelöst, was dann dazu

führt, die optimale Aktion für den jeweils nächsten Schritt in der diskreten PS-Sequenz zu bestimmen. Jedoch kann der Zeithorizont für die Entscheidungsfindung auch erweitert werden, und zwar im Hinblick auf das globale Netz der PS.

Wenn man zu jeder Kante das Komplement der zugehörigen Effizienz- und Affinitätswerte ($I - E \cdot a$) zuordnet, kann man mit einem Baumsuche-Algorithmus (z. B. Dijkstra-Algorithmus) den optimalen Pfad vom momentan aktiven PS zum Ziel ermitteln und daraus die nächste auszuführende PS extrahieren.

Im vorangegangenen Kapitel wurde gezeigt, dass der Wert jeder einzelnen Effizienz durch drei verschiedene Parameter gegeben ist:

- die Verfügbarkeit der Ressourcen
- ihre Qualität und
- die Zuverlässigkeit dieser Abschätzungen

Das Abschätzen, ob eine Ressource verfügbar ist oder nicht, erfordert keinen großen Aufwand und eine Abwägung der Zuverlässigkeit (durch Bewertung der vorherigen Ausführungen) ist zunächst nicht zwingend nötig.

Als Methode für die Abschätzung der Qualität wurde ein Fuzzy-Ansatz gewählt. Die vom Roboter gesammelten Informationen über die Ressourcen, die an den einzelnen PS beteiligt sind, werden in linguistische Variablen überführt und daraufhin nach Vorschriften ausgewertet, die menschliches Denken und heuristische Erfahrungen reproduzieren.

Im Bestreben, das Problem der Entscheidungsfindung zu lösen, wurden sowohl Bayes- als auch Fuzzy-Methoden sorgfältig untersucht [7]. Dabei stellte sich heraus, dass die Fuzzy-Methode geeigneter ist, da sie eine transparentere Struktur bietet und die getroffene Entscheidung für einen menschlichen Operator besser nachvollziehbar ist.

Das Prinzip der Qualitätsabschätzung kann anhand des exemplarischen Beispiel-PS „Lokalisieren eines Objektes mittels einer Stereokamera“ verdeutlicht werden. In diesem Fall sind die beiden hauptsächlich verwendeten Ressourcen das Objekt und die Kamera. Um das Beispiel zu vereinfachen, wird davon ausgegangen, dass das Objekt stets verfügbar und seine Effizienz somit konstant gleich eins ist. Dadurch kann die Effizienz des PS ausschließlich auf die der Kamera reduziert werden.

Die drei Hauptfaktoren, welche die Effizienz einer Kamera beeinflussen, sind die Umgebungshelligkeit, das Signalrauschen und der Arbeitsbereich des Sensors. Beispielhafte Zugehörigkeitsfunktionen, die diesen drei Variablen zugeordnet werden könnten, sind in Abb. 4 dargestellt.

Nach der Fuzzifizierung werden die Variablen nach recht intuitiven Vorschriften ausgewertet, wie zum Beispiel:

*If (Noise is High) or (Range is too far) or
(Illumination is Low) then (Quality is Low)*

*If (Noise is Low) and (Range is Good) and
(Illumination is High) then (Quality is
High)*

(5)

Nach dem Defuzzifizierungsprozess liegt ein Wert zwischen 1 und 0 vor, der durch die zugehörige Affinität gewichtet wird. Schließlich löst ein einfacher Vergleich zwischen den resultierenden Werten der verschiedenen PS den Konflikt.

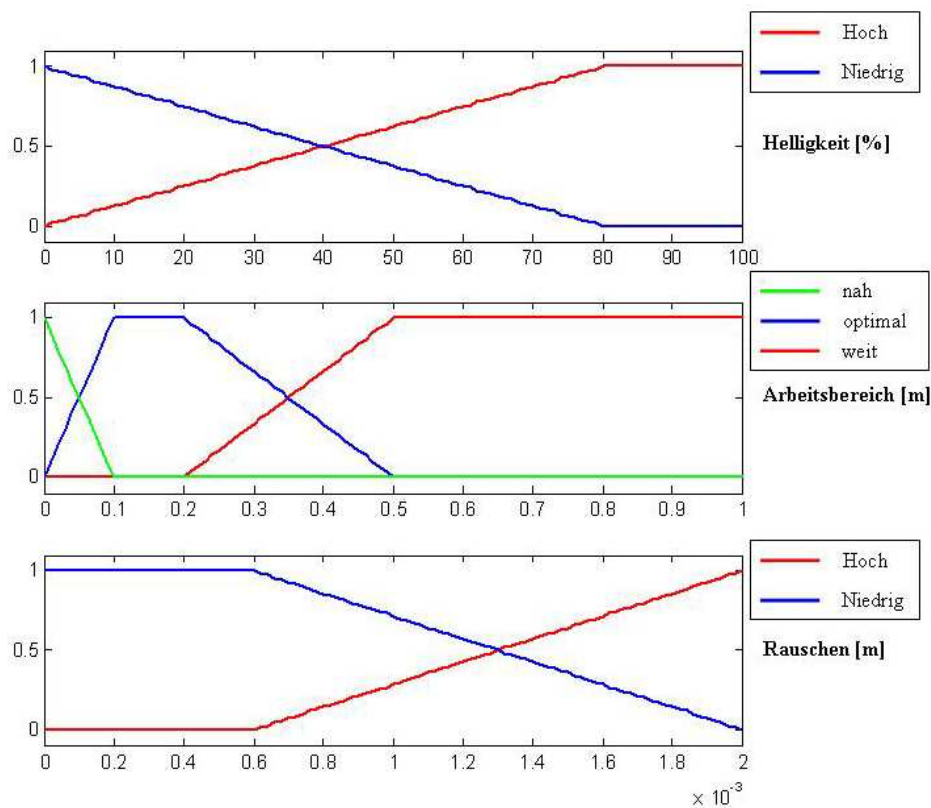


Abb. 4: Membership-Funktionen der Variablen Beleuchtung (Illumination), Entfernung zum Objekt (Range) und Rauschen (Noise)

4. Experimentelle Ergebnisse

Der vorgestellte, Fuzzy-basierte Entscheidungsfindungsalgorithmus wurde in die diskret-kontinuierliche Regelung integriert und experimentell für verschiedene Szenarien untersucht.

Die in diesem Artikel dargestellten Ergebnisse beziehen sich auf eine Greifaufgabe, deren Ziel das Greifen eines Bechers aus einer A-priori unbekannten Position ist. Um das zu erreichen, hat der Roboter drei verschiedene Handlungsweisen zur Verfügung, die mit drei verschiedenen PS einhergehen:

- Lokalisieren und Verfolgen des Bechers mithilfe der Stereokamera im Sensorkopf und gerichtete Annäherungsbewegung ausführen – PS „Bewegen per Stereokamera“;
- Lokalisieren und Verfolgen des Bechers mithilfe der Handkamera und gerichtete Annäherungsbewegung ausführen – PS „Bewegen per Handkamera“;
- Annähern an den Tisch und Lokalisieren des Bechers durch eine tastende Untersuchung der Oberfläche – PS „Bewegen per taktilem Sensor“.

Während der Durchführung der Aufgabe ist es dem Roboter möglich, abhängig von der aktuellen Situation zwischen den drei Möglichkeiten umzuschalten.

Das resultierende Netz kann erweitert werden, wenn unerwartete Ereignisse auftreten wie zum Beispiel das Wahrnehmen eines Hindernisses in der Annäherungsphase durch die Handkamera (Abb. 5a) oder eine Behinderung der Sicht der Stereokamera durch die Bewegung des Arms (Abb. 5b). Für diese Fälle wurden zwei unterschiedliche Strategien eingebunden, um das Problem zu beheben (eine vertikale Abweichbewegung für den ersten Fall und eine laterale Abweichbewegung für den zweiten). Das in Abb. 6 dargestellte Petri-Netz stellt die beschriebene Handlungsstruktur dar.

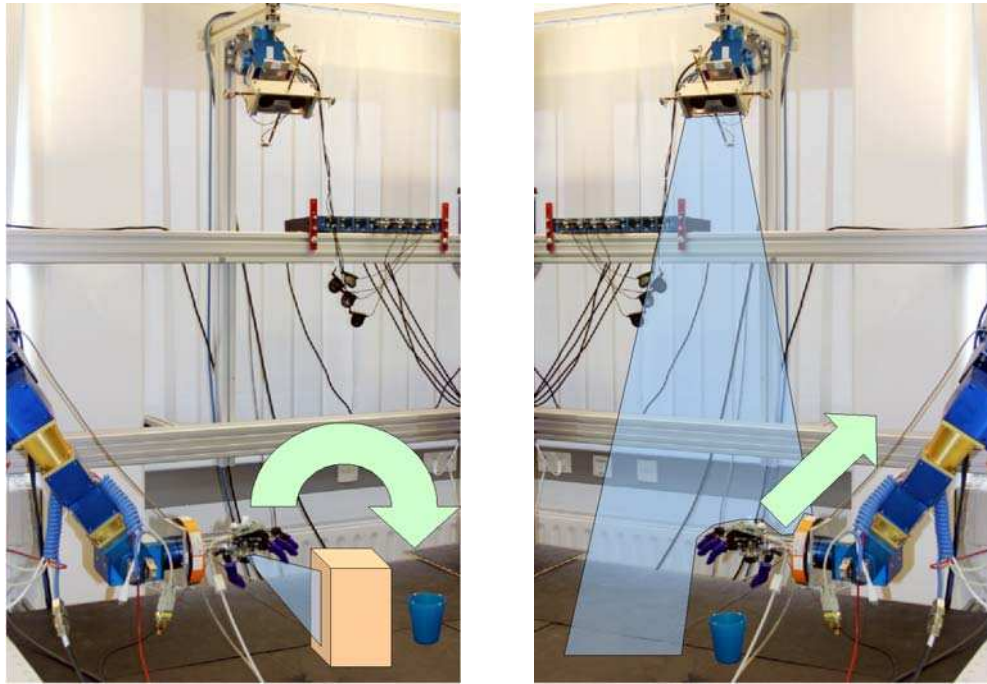


Abb. 5: Unerwartete Situationen während der Ausführung der Aufgabe:
 Abschattung a) der Handkamera b) der Stereokamera

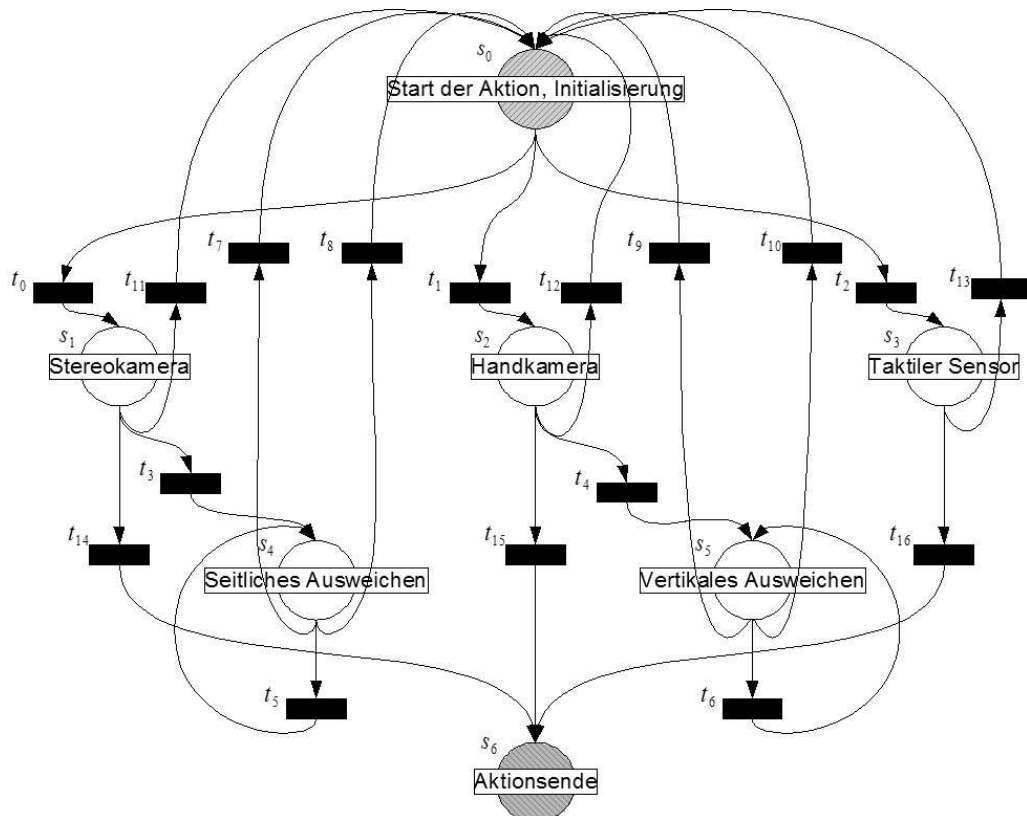


Abb. 6: Petri-Netz-Modellierung der Greifaufgabe

Die Affinitäten dieser drei wesentlichen PS sind wie folgt definiert:

- $a_{HandKamera} = 1$ (am genauesten)
- $a_{StereoKamera} = 0.8$
- $a_{TaktileSensoren} = 0.6$ (am langsamsten)

Während der normalen Ausführung einer Aufgabe, sofern alle Sensoren korrekt arbeiten, ist die Effizienz des PS „Bewegen per Handkamera“ für die erste Bewegungsphase größer als die beiden anderen (Abb. 7). Je weiter sich der Roboter auf das Objekt zubewegt, umso näher kommt die Handkamera dem Objekt, wodurch sich ihr Arbeitsbereich verkleinert. Ab einem gewissen Punkt befindet sich die Handkamera zu nah am Objekt, und das PS „Bewegen per Stereokamera“ wird zur optimalen Vorgehensweise zum Greifen des Objektes.

Die zwei Spitzen bei ca. 7 und 14 Sekunden sind auf ausgelassene Messzyklen der Handkamera zurückzuführen, was zu einer sofortigen Verminderung ihrer Zuverlässigkeit führt. Die Nummernbezeichnung der Stellen in Abb. 7 stimmt mit der in Abb. 6 überein. In den Diagrammen werden nur die Variablen der beiden optischen Sensoren dargestellt, da die taktilen Sensoren in den gezeigten Ergebnissen nicht zum Einsatz kamen.

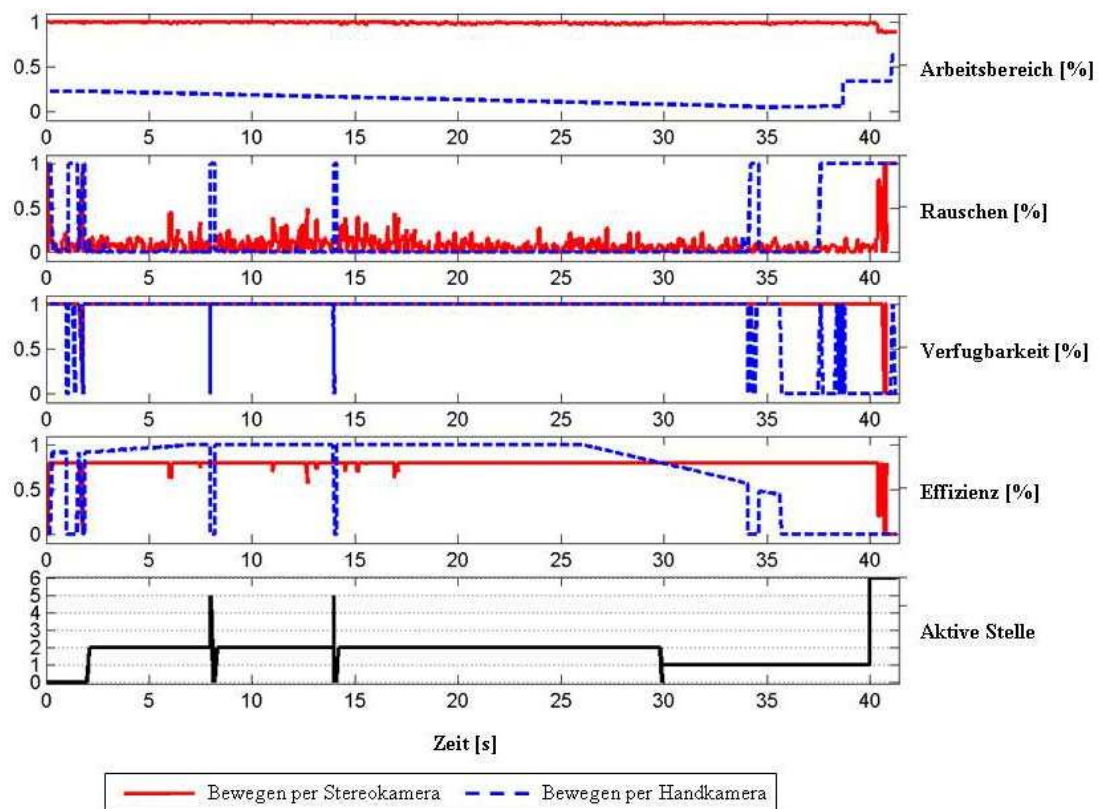


Abb. 7: Variablen für die Entscheidung während der Ausführung einer normalen Greifaufgabe

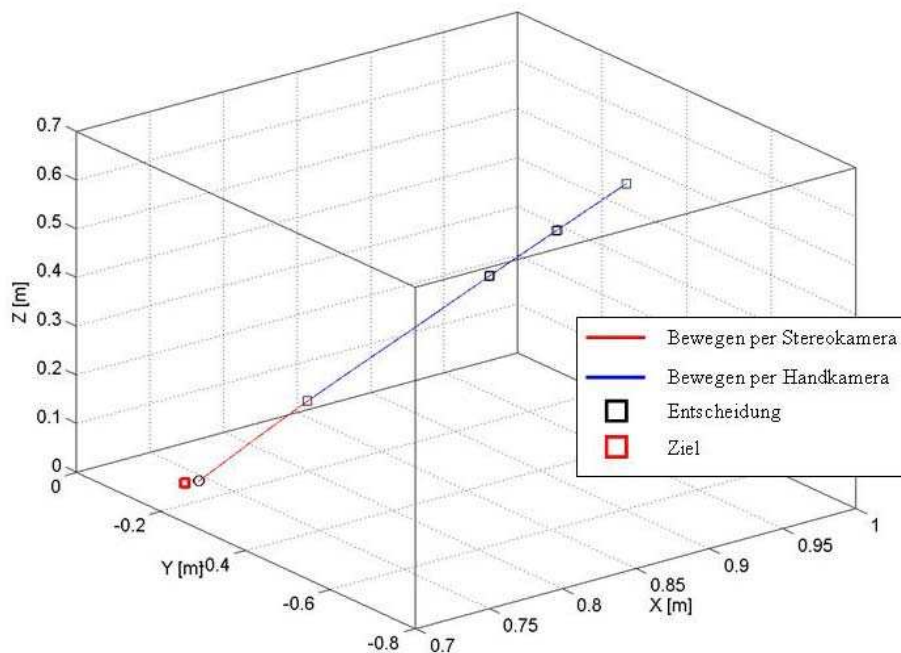


Abb. 8: Kartesische Trajektorie während der Ausführung einer normalen Greifaufgabe

In Abb. 8 ist die kartesische Trajektorie der Roboterbewegung dargestellt, wobei das jeweils aktive PS der Kurvenfärbung zu entnehmen ist.

Zur Überprüfung des entwickelten diskret-kontinuierlichen Regelungskonzeptes für eine Aufgabe mit unerwarteten Ereignissen wurde den beiden Kameras manuell die Sicht verdeckt, um Situationen wie in Abb. 5 zu erzeugen.

Die Ergebnisse sind in Abb. 9 und Abb. 10 dargestellt.

Zuerst wurden vier Abschattungen der Handkamera simuliert. Der Zusammenhang zwischen dem Absinken der Verfügbarkeit (während diesem Zeitintervall gehen keine Messungen ein) und dem Wert der Zuverlässigkeit ist klar erkennbar. Der Roboter reagiert mit vertikalen Bewegungen (Stelle 5), bis eine neue Messung vorliegt (zurück nach Stelle 2).

Das Ausschalten der Stereokamera (bei ca. 45 Sekunden) brachte den Roboter dazu, sich für ein Fortfahren mithilfe der Stereokamera zu entscheiden (Stelle 1), obwohl ihr optimaler Arbeitsbereich noch nicht erreicht war.

Schließlich wurde die Stereokamera dreimal kurz verdeckt, worauf der Roboter eine laterale Bewegung ausführte (Stelle 4), bis er wieder ein freies Sichtfeld hatte (zurück nach Stelle 1). Die beiden Kurven der Zuverlässigkeitswerte weisen, wie beim vorhergegangenen Versuch, einige vereinzelte Spitzen auf.

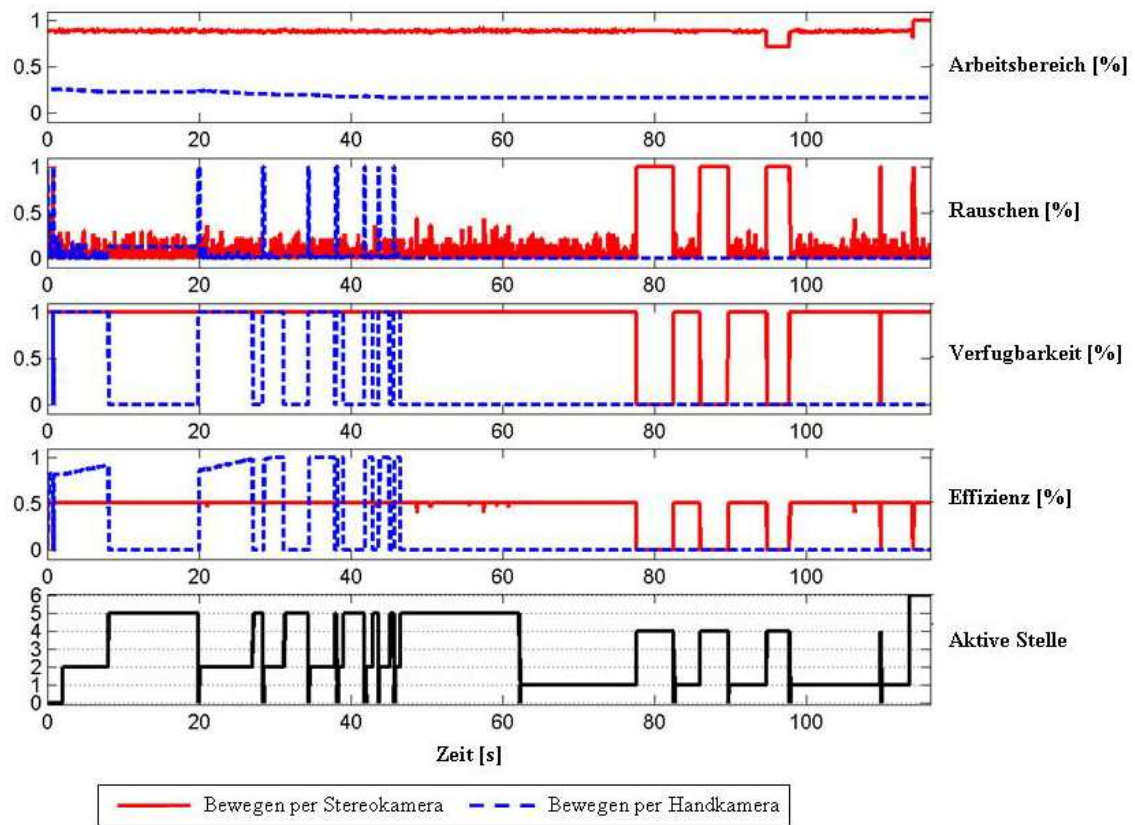


Abb. 9: Variablen für die Entscheidung während der Ausführung einer Greifaufgabe mit Abschattungen der Kameras

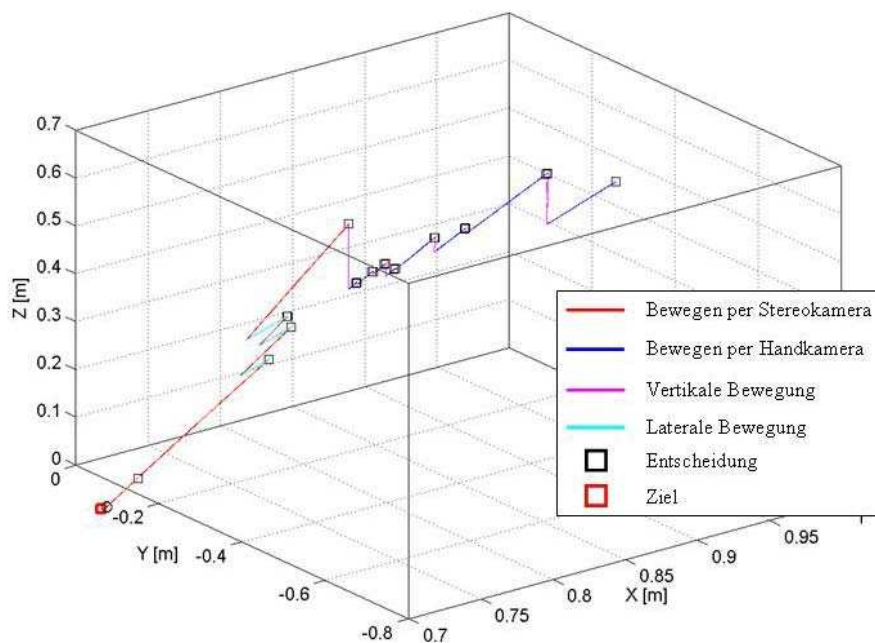


Abb. 10: Kartesische Trajektorie während der Ausführung einer Greifaufgabe mit Abschattungen der Kameras

5. Zusammenfassung

Im vorliegenden Beitrag wurde ein neues multisensorielles diskret-kontinuierliches Regelungskonzept vorgestellt, das in der Lage ist, komplexe Roboteraufgaben in einer zeitvarianten Umgebung flexibel zu bewältigen. Hierzu wurde eine flexible und transparente Aufgabenstruktur unter der Verwendung von Aktionsprimitiven (Primitive Skills, PS) entwickelt. Jedes PS wird in einem mitlaufenden Petri-Netz abgebildet, in dem die diskrete Aufgabenstellung modelliert ist.

Durch seine perzeptorische Wahrnehmung ist der Roboter fähig, Fehlfunktionen und unerwartete Ereignisse oder Sachverhalte während der Ausführung der Aufgabe zu identifizieren. Ein Fuzzy-Ansatz, der die unscharfen kognitiven Eigenschaften des Menschen modelliert, wurde untersucht, um den Roboter mit der Intelligenz vorzusehen, die er bei der flexiblen Bewältigung unerwarteter Situationen benötigt.

Die Tauglichkeit des vorgeschlagenen Konzepts wurde durch erste Versuche mit einem Greifvorgang mithilfe unterschiedlicher Sensoren demonstriert. Die erzielten Resultate zeigen, dass die vorgestellte Methode auch für kompliziertere Aufgaben geeignet ist.

Danksagung

Mit großem Dank möchten die Autoren die Unterstützung des in diesem Artikel vorgestellten Forschungsprojektes durch die Deutsche Forschungsgemeinschaft (DFG) erwähnen.

Literatur

- [1] Antsaklis, P.J.; Koutsoukos, X.D.; Zaytoon, J., "On Hybrid Control of Complex Systems: A Survey", 3rd International Conference ADMP'98, Automation of Mixed Processes: Dynamic Hybrid Systems, März 1998.
- [2] Asfour, T.; Ly, D.N.; Regenstein, K.; Dillmann, R., "Coordinated Task Execution for Humanoid Robots", 9th International Symposium on Experimental Robotics, Marina Mandarin, Singapore, Juni 2004, pp. 1-10.
- [3] Cao, T.; Sanderson, A.C., "Modeling of Sensor-Based Robotic Task Plans Using Fuzzy Petri Nets", Proceedings of the Fourth International Conference on Computer Integrated Manufacturing and Automation Technology, 10.-12. Oktober 1994.
- [4] Chang, Z.-H.; Bian, X.-Q.; Shi, X.-C., "Autonomous Underwater Vehicle: Petri Net Based Hybrid Control of Mission and Motion", Proceedings of the Third International Conference on Machine Learning and Cybernetics, 26.-29. August 2004.
- [5] Chew S. F. ; Lawley M. A., "Robust Supervisory Control for Production Systems With Multiple Resource Failures", IEEE Transactions on Automation Science and Engineering, Juli 2006.
- [6] Chung, S.-Y.; Lee, D.Y., "Discrete Event Systems Approach to Fixtureless Peg-in-Hole Assembly", Proceedings of the American Control Conference, 25.-27. Juni 2001.
- [7] D'Angelo, L.T., "Ereignis-diskrete Regelung für humanoide Roboter basierend auf dem Aktionsprimitivenkonzept", Diplomarbeit, Fraunhofer IITB, Universität Karlsruhe (TH), 2006.

- [8] Estlin, T.; Volpe, Rich; Nesnas, I.; Mutz, D.; Fisher, F.; Engelhardt, B.; Chien, S., „*Decision-Making in a Robotic Architecture for Autonomy*“, International Symposium on Artificial Intelligence, Robotics and Automation for Space (i-SAIRAS 2001), Montreal, CA. Juni 2001.
- [9] Finzi, A.; Ingrand, F.; Muscettola, N., „*Model-based Executive Control through Reactive Planning for Autonomous Rovers*“, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 28. September – 2. Oktober, Sendai, Japan, 2004.
- [10] Gadeyne, K.; Lefebvre, T.; Bruyninckx, H., "*Bayesian Hybrid Model-State Estimation Applied to Simultaneous Contact Formation Recognition and Geometrical Parameter Estimation*", The International Journal of Robotics Research, 2005.
- [11] Gao, M.; Zhou, M., "*Control Strategy Selection for Autonomous Vehicles in a Dynamic Environment*", IEEE International Conference on Systems, Man and Cybernetics, 10.-12. Oktober 2005.
- [12] Jantzen, J., "*Fuzzy Supervisory Control*", Tech. report n. 98-H-875, Technical University of Denmark, Department of Automation, November 1998.
- [13] Kobayashi, K.; Nakatani, A.; Takahashi, H.; Ushio, T., "*Motion planning for humanoid robots using timed Petri net and modular state net*", IEEE International Conference on Systems, Man and Cybernetics, 6.-9. Oktober 2002.
- [14] Lehmann, A.; Mikut, R.; Asfour, T., "*Petri Nets for Task Supervision in Humanoid Robots*", In Proceedings: 37th International Symposium on Robotics (ISR 2006), München, Mai, 2006.
- [15] Lundell, M.; Jinpeng Tang; Nygard, K., "*Fuzzy Petri Net for UAV Decision Making*", Proceedings on the 2005 International Symposium on Collaborative Technologies and Systems, 15.-20. Mai 2005.
- [16] Milighetti, G.; Kuntze, H.-B., „*On the Discrete-Continuous Control of Basic Skills for Humanoid Robots*“, International Conference on Intelligent Robots and Systems IROS 2006, Beijing, China, 9.-15. Oktober 2006.
- [17] Nenninger, G.; Krebs, V., "*Modeling and Analysis of Hybrid Systems: A New Approach Integrating Petri Nets and Differential Equations*", Proceedings of the Joint Workshop on Parallel and Distributed Real-Time Systems, 1997.
- [18] Palomeras, N.; Carreras, M.; Ridao, P.; Emili, H., "*Mission control system for dam inspection with an AUV*", Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, 9.-15. Oktober 2006.
- [19] Schrempf, O.C.; Hanebeck, U.D., "*A generic model for estimating user intentions in human-robot cooperation*", Proceedings of the Second International Conference on Informatics in Control, Automation and Robotics (ICINCO), 14.-17. September 2005.
- [20] Serrano, N., "*A Bayesian Framework for Landing Site Selection during Autonomous Spacecraft Descent*", Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9.-15. Oktober 2006.
- [21] Thomas, U.; Finkemeyer, B.; Kröger, T.; Wahl, F.M., "*Error-Tolerant Execution of Complex Robot Tasks Based on Skill Primitives*", IEEE International Conference on Robotics and Automation, Taipei, Taiwan, Mai 2003.
- [22] Yazdi, H., "*Control and Supervision of Event-Driven Systems*", Ph.D. thesis, FLS Automation A/S, Denmark, Technical University of Denmark, Februar 1997.

- [23] Zhou, M.-C.; Jeng, M.D., "*Modeling, Analysis, Simulation, Scheduling, and Control of Semiconductor Manufacturing Systems: A Petri Net Approach*", IEEE Transactions on Semiconductor Manufacturing, August 1998.
- [24] Modular Controller Architecture MCA2 <http://www.mca2.org/>
- [25] Asfour, T., Regenstein, K., Azad, P., Schröder, J., Dillmann, R. : "*ARMAR-III: A Humanoid Platform for Perception-Action Integration*", International Workshop on Human-Centered Robotic Systems (HCRS), München, 6. – 7. Oktober, 2006.

Kontrolle dynamischer Eigenschaften des Online-Lernens in Neuro-Fuzzy-Systemen mit dem SILKE-Ansatz

Nils Rosemann, Werner Brockmann

Institut für Informatik,
Universität Osnabrück
Albrechtstraße 28
49076 Osnabrück
Tel. (0541) 969-2598
Fax (0541) 969-2799

E-Mail: nils@informatik.uni-osnabrueck.de;
brockmann@informatik.uni-osnabrueck.de

Kurzfassung

Viele technische Systeme wie Kraftfahrzeuge oder mobile Roboter haben heutzutage eine solche Komplexität erreicht, dass ihr Verhalten nicht mehr vollständig zur Entwurfszeit festgelegt werden kann. Dies liegt an der inneren Komplexität der Systeme, aber auch daran, dass sie auf komplexe Weise mit ihrer Umwelt interagieren. Ein Ansatz zur Reduktion des Entwurfsaufwands und der Beherrschung der Komplexität ist, dass diese Systeme teilautonom ihr Verhalten zur Laufzeit einstellen und/oder an sich verändernde Betriebsbedingungen anpassen. Diese Selbstoptimierung lässt sich durch inkrementelles Online-Lernen erreichen. Um die durch den geschlossenen Wirkungskreis entstehenden Probleme wie schlechte Konvergenz und potentiell chaotisches Systemverhalten zu beherrschen, verwenden wir lernfähige Fuzzy-Systeme mit entsprechenden Erweiterungen. Zur Kontrolle des dynamischen Lernverhaltens ist der SILKE-Ansatz eine solche Möglichkeit. Dabei wird die Umsetzung von Lernimpulsen durch lokale Template-Funktionen gesteuert, um gezielt Meta-Eigenschaften des Lernvorgangs zu steuern.

Der SILKE-Ansatz greift vergleichbar zur Regularisierung bei anderen Lernparadigmen direkt in den Lernvorgang ein. Seine Wirkung kann global über die Anpassungsrate eingestellt werden. In diesem Beitrag soll gezeigt werden, dass der SILKE-Ansatz auch in der Praxis an einem nicht-trivialen technischen Prozess vorteilhaft eingesetzt werden kann. Neben der Untersuchung zur optimierten Einstellung der Anpassungsrate wird die Erweiterung um eine dynamische Reduktion der Lernrate vorgeschlagen, um einen guten Kompromiss aus Konvergenzgeschwindigkeit und -güte zu erreichen.

1 Einleitung

1.1 Problemstellung

Die Kontrolle komplexer technischer Systeme erfordert in der Praxis vielfach einen Kompromiss aus Entwurfsaufwand und erzielten Systemeigenschaften, weil z. B. die mögliche Entwurfszeit begrenzt oder das System schlecht zu modellieren ist. Nichtsdestotrotz möchte man das Systemverhalten im Großen, d. h. als Ganzes, und im

Kleinen, also in den einzelnen Details, beherrschen. Lerntechniken können hier helfen, indem der Entwurfsaufwand auf eine Selbstoptimierung des Systems bzw. die Selbsteinstellung der Parameter im laufenden Betrieb verlagert wird. Dadurch müssen viele Details des zu kontrollierenden Systems zur Entwurfszeit nicht genau modelliert werden. Das vereinfacht den Entwurfsvorgang deutlich, sofern der Entwurfsaufwand, um das Lernen zu integrieren, im Rahmen bleibt. In ähnlicher Weise kann zeitvarianten Phänomenen durch Selbstadaption begegnet werden, ohne sie explizit modellieren zu müssen.

Das Lernen erfolgt online. D. h. die Lerndaten werden inkrementell generiert und müssen unter harten Echtzeitbedingungen verarbeitet werden. Das Hauptproblem ist, dass so durch lokale Lerndaten die zu lernende Funktion nur punktuell beschrieben wird. Allerdings sind die Lerndaten meist sehr ungleichmäßig im Eingangsraum verteilt. In der Regel sind sie auch noch verrauscht, was zu (kleinen) lokalen Widersprüchen führt. Ein weiteres Problem ergibt sich aus dem Plastizitäts-Stabilitäts-Dilemma: Auf der einen Seite soll das Lernen möglichst schnell global konvergieren (Konvergenzgeschwindigkeit), was eine große Flexibilität des lernenden Systems erfordert. Und auf der anderen Seite darf die gelernte Funktion durch das ständig mitlaufende Lernen im weiteren Lernverlauf nicht wieder verschlechtert werden (Konvergenzgüte). Auch wenn kurzzeitige Störungen auftreten oder wenn nicht alle Effekte modelliert sind, sollten z. B. keine Oszillationen des gelernten Verhaltens auftreten. Denn das bedeutet, es würde zwar im Mittel ein gutes Verhalten erreicht, aber das gelernte Wissen würde ständig hin und her geändert.

Besonders kritisch ist das Konvergenzverhalten beim Closed-loop-Lernen. Denn hier wird das gelernte Wissen unmittelbar zur Steuerung des Prozesses angewendet und hat damit Auswirkungen auf das, was in Zukunft gelernt werden wird. Dadurch wird ein potentiell chaotisches System aufgebaut, bei dem kleine Änderungen der Anfangsbedingungen zu einem gänzlich anderen Lernverhalten führen können, also auch bis zur Instabilität. Der Lernvorgang sollte deshalb als solches auf einer Metaebene kontrolliert werden können.

Für das Lernen großer Datenmengen (batch-learning) stehen mächtige, theoretisch motivierte Methoden zur Verfügung, um den Lernvorgang zu steuern und eine möglichst große Generalisierungsfähigkeit des Systems zu erreichen. Die beiden wichtigsten sind Regularisierung [1,2] und vorzeitiges Abbremsen des Lernvorgangs [3] durch monotonen Absenken der Lernrate (early stopping). Diese Methoden lassen sich auf inkrementelle Lernalgorithmen übertragen [4,5], die eigentliche Motivation bleibt aber den Lernvorgang so zu steuern, dass das System optimal generalisiert. Eine direkte Einflussnahme auf den potentiell chaotischen Adaptionsprozess ist damit nicht möglich. Die Gefahr einer chaotischen Rückkopplung durch Lernen reduziert sich umso mehr, je weniger Einfluss ein einzelner Lernschritt auf das Gesamtsystem hat. Dies motiviert den Einsatz von lokalen Lernverfahren, bei denen jeder Lernschritt die gelernte Abbildung nur für einen begrenzten Bereich um den aktuellen Eingabevektor verändert [6-8]. Gesucht sind also Erweiterungen von lokalen, inkrementellen Lernalgorithmen, die eine Kontrolle des Lernvorgangs als solchem in Closed-loop-Anwendungen ermöglichen.

Ein Ansatz dafür ist Tagging in lernfähigen Fuzzy-Systemen, d. h. das Ausschließen von Regeln aus dem Lernprozess durch ein Tag-Bit. Dadurch kann Vorwissen fixiert werden [9]. So wird ein Sicherheitskorridor für das lernende System geschaffen, den es (bei richtiger Gestaltung des Vorwissens) auch beim Lernen nicht verlässt, weil es an dem Betreten kritischer Betriebsbereiche gehindert wird. Dieses Vorwissen ist vergleichsweise einfach anzugeben, weil es nicht weiter optimiert werden muss. Der

SILKE-Ansatz (System to Immunize Learning Knowledge-based Elements, [10-12]) geht noch einen Schritt weiter, indem ein lokal arbeitendes Template den Lernvorgang online gezielt so beeinflusst, dass das gelernte Wissen Meta-Eigenschaften genügt (insbesondere auch innerhalb des Sicherheitskorridors). Je nach verwendetem Template sind dies z. B. Glattheit, Stetigkeit oder Monotonie der gelernten Funktion.

So werden Möglichkeiten für einen guten Kompromiss geschaffen, um auf der einen Seite Systemeigenschaften wie Regelgenauigkeit, Konvergenzgeschwindigkeit, Stabilität des gelernten Wissens, Stellenergie, Stress der Aktuatoren bzw. des kontrollierten Systems und gleichzeitig auf der anderen Seite den Aufwand für das Engineering des Lerngesetzes sowie die Beherrschbarkeit und Kontrolle des Lernvorgangs zu optimieren.

1.2 Zielsetzung

Um die grundlegenden Eigenschaften des SILKE-Ansatzes zu untersuchen, wurden bisher Simulationen durchgeführt [10,11]. Darüber hinaus wurde mit der Anpassungsrate eine Möglichkeit eingeführt, den SILKE-Ansatz auf die Besonderheiten des jeweiligen Prozesses anzupassen [12]. In dieser Arbeit wird der SILKE-Ansatz daraufhin überprüft, ob er auf reale, nicht-triviale Prozesse übertragbar ist und ob er unter praxisrelevanten Betriebsbedingungen die gleiche Wirkung entfaltet. Der untersuchte reale Prozess ist ein Wagen mit aufmontiertem Pendel, der als „inverses Pendel“ betrieben wird. Im Vergleich zu anderen realen Aufbauten gleicher Art weist dieser Aufbau bewusst einige schwer zu modellierende Eigenschaften wie Reibung, Schlupf und Lose sowie starke Begrenzung der Stellenergie auf. Das macht den Entwurf eines Kontrollsystems auch beim konventionellen Vorgehen schwierig zu beherrschen.

Der Pendelwagen wird von einem lernfähigen Fuzzy-System als zentralem Element kontrolliert, das online das Regelverhalten des Winkelreglers ausgehend von grobem a priori Wissen optimiert. Die Wirkung des SILKE-Ansatzes wird anhand verschiedener Untersuchungen gezeigt und es wird untersucht, ob das Konvergenzverhalten weiter verbessert werden kann. Dazu wird die Lernrate beim laufenden Online-Lernen kontinuierlich reduziert. Untersuchungen zeigen, welchen Einfluss beide Methoden im Vergleich haben und ob sich durch ihre Kombination eine sinnvolle Verbesserung ergibt.

2 Anwendungsbeispiel

2.1 Versuchsaufbau

Um die Wirkung des SILKE-Ansatzes in der Praxis an einem realen Beispiel zu zeigen und hier auch den Effekt der Anpassungsrate zu untersuchen, wurde bewusst ein Beispielprozess gewählt, der viele praxisrelevante Effekte vereint, aber dennoch recht anschaulich und einfach zu betreiben ist. Dabei handelt es sich um einen Pendelwagen mit voll schwingfähig montiertem Pendel, wie er in Abb. 1 gezeigt ist. Dieser Aufbau wird als inverses Pendel betrieben, welches eine Standardanwendung für die Untersuchung lernfähiger Kontrollsysteme ist [13].

Allerdings werden die meisten solcher Untersuchungen in der Simulation durchgeführt, was die Aussagekraft für praktische Anwendungen einschränkt. Es gibt deutlich weniger Arbeiten an realen Pendelaufbauten, die dann zudem meist stark idealisiert

sind, z. B. [14]. Um gerade den praxisrelevanten Aspekt zu betonen, werden die nachfolgenden Untersuchungen an einem Pendelwagen ausgeführt, der im Vergleich zu anderen real angewendeten Aufbauten über einige ungünstige Eigenschaften verfügt, die aber häufig in der Praxis auftreten.



Abb. 1: Verwendeter Versuchsaufbau

Die Wagenmasse beträgt 2,34 kg (ohne Pendel), die Pendelmasse 0,1 kg bei einer Pendellänge von 15,7 cm (eff.). Die Höhe der Pendelachse über Grund ist 22 cm. Der Antrieb erfolgt mittels eines Standard-Modellbau-Gleichstrommotors mit ca. 60 W Nominalleistung über ein zweistufiges Getriebe und Moosgummi-Bereifung. Die Positions- und Winkelerfassung erfolgen über Inkrementalgeber (Ortsauflösung; 0,25 mm, Winkelauflösung: 0,11°).

Das Verhältnis aus Wagenmasse und Pendelmasse ist hier allerdings sehr ungünstig, weil eine große Masse beschleunigt werden muss, um das relativ leichte Pendel zu balancieren. Außerdem führt das Getriebeispiel zu einer großen Lose von ca. $\pm 0,5$ mm, die gerade im Arbeitspunkt wirkt, weil das Pendel durch Hin- und Herfahren des Wagens balanciert werden muss. Beim starken Beschleunigen kann auch Schlupf durch durchdrehende Räder auftreten. Die Abtastzeit beträgt 10 ms. Das wäre zwar ausreichend schnell bei der Trägheit des Wagens, ist aber grenzwertig für das Balancieren des Pendels. Dadurch weist dieser Beispielprozess eine ganze Reihe von praxisrelevanten Dreckeffekten und Nichtlinearitäten gleichzeitig auf, die auch einen konventionellen Regleransatz und -entwurf deutlich erschweren.

2.2 Reglerstruktur

Für die Regelung wird die in Abb. 2 gezeigte Kaskadenstruktur aus Positionsregler und lernfähigem Winkelregler verwendet. Der Positionsregler ist trotz der Nichtlinearitäten als PID-Regler ausgeführt, dem eine Logarithmierung des Positionsfehlers vorgeschaltet ist. Der Winkelregler wird durch ein (statisches) System für eine grobe Reibungskompensation unterstützt.

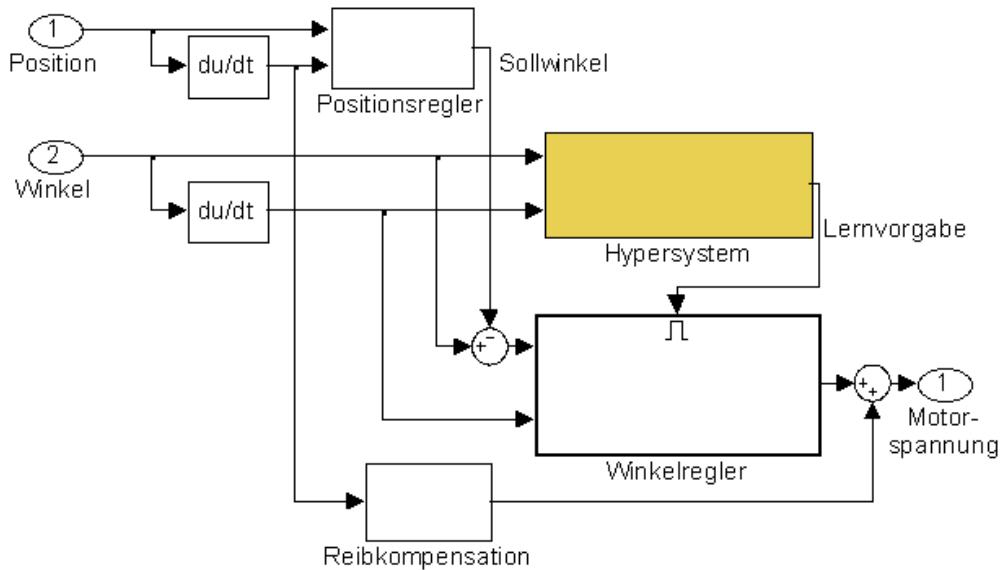


Abb. 2: Reglerstruktur

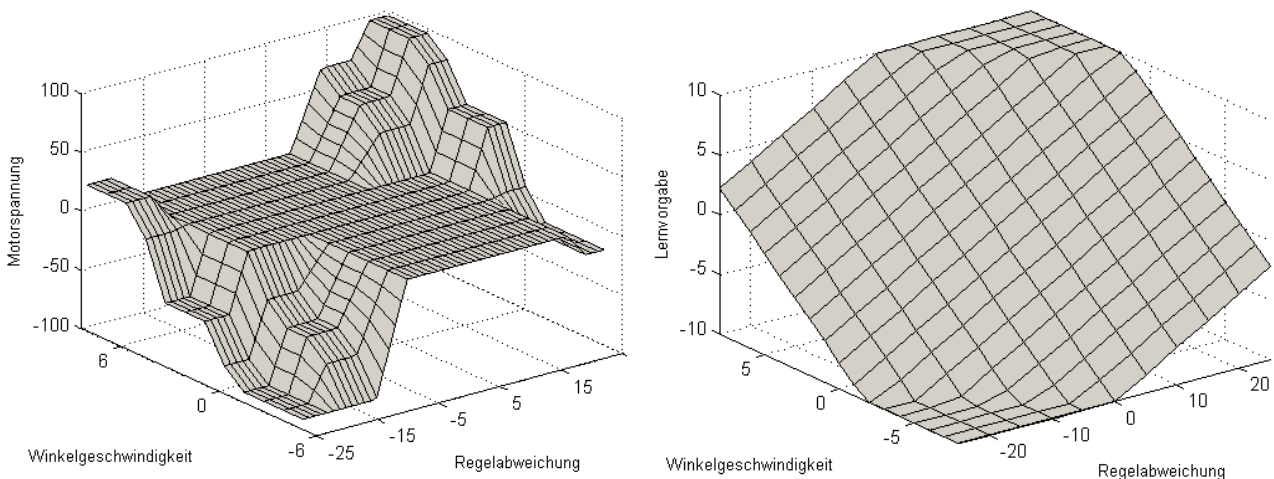


Abb. 3: Kennflächen des Winkelreglers mit Vorwissen vor dem Lernen (links) und des Hyperwissens (rechts)

In der Lernarchitektur stellt der Winkelregler das zentrale, lernende System dar, das closed-loop in den Regelkreis integriert ist. Es erhält seine Lernvorgaben durch ein Adaptionsgesetz, das im übergeordneten Hypersystem abgelegt ist. Die Lernvorgaben werden zu jeder Abtastung mit einer Lernrate von 1 gelernt und verändern das Regelverhalten lokal und inkrementell. Das verwendete Adaptionsgesetz ist sehr einfach gehalten. Es handelt sich um eine (auf ± 10 begrenzte) Ebene durch den Nullpunkt (Abb. 3 rechts), die linear abhängig von Winkel und Winkelgeschwindigkeit Korrekturvorgaben erzeugt. Dadurch wird keine Korrektur erzeugt, wenn der Winkelregler mit einer Winkelgeschwindigkeit von Null den Regelfehler exakt ausgleicht. Ist die Winkelgeschwindigkeit ungleich Null, aber so gestaltet, dass sie den Regelfehler ausgleichen wird, wird ebenfalls keine Änderung durchgeführt. In allen

anderen Fällen wird die Stellaktion durch die Lernvorgabe so beeinflusst, dass die vorgenannten Bedingungen iterativ erreicht werden.

Bei dem lernfähigen Winkelregler handelt es sich um ein Takagi-Sugeno-Fuzzy-System 0. Ordnung. Es enthält wenige, einfache (Sicherheits-)Regeln als Vorwissen, das das Pendel am Umkippen hindert. Die so definierte Reglerfunktion ist in Abb. 3 (links) dargestellt. Sie löst im eigentlichen Arbeitsbereich (um den Nullpunkt) keine Stellaktionen aus, enthält also in diesem Bereich praktisch kein Wissen.

Bei den folgenden Untersuchungen wurde auf das Tagging der Sicherheitsregeln verzichtet, also kein fester Sicherheitskorridor eingebaut, um die Wirkung der untersuchten Methoden auf die Dynamik des Lernvorgangs unverfälscht beurteilen zu können.

3 Lernen ohne den SILKE-Ansatz

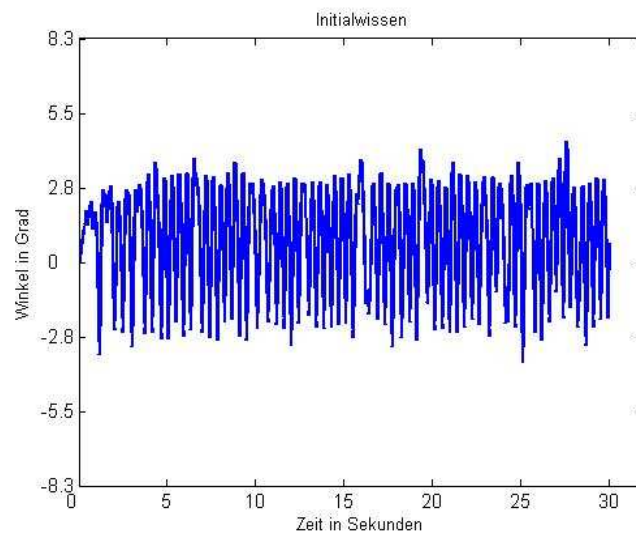


Abb. 4: Verhalten des Winkelreglers nur mit Vorwissen ohne Lernen

Abb. 4 zeigt den zeitlichen Verlauf des Winkels, wenn der Pendelwagen nur mit dem Vorwissen und ohne Lernen betrieben wird. Es ist durch eine heftige Regelbewegung des Pendels sowie eine Bewegung des Wagens mit einer sehr hohen Frequenz von ca. 2,1 Hz gekennzeichnet. Dadurch erfährt das gesamte System einen großen Stress (und Verschleiß). Es bewegt sich aber mit einer relativ geringen Amplitude von ca. 3° .

Zum Vergleich zeigt Abb. 5 das Verhalten mit Lernen. Der Einfluss des Hypersystems bewirkt zunächst größere Ausschläge des Pendels von über 8° . Es kommt also vorübergehend zu einer Verschlechterung des Regelverhaltens durch Einschwingvorgänge beim Lernen, die offensichtlich auch zu Änderungen am Vorwissen führen. Insgesamt führt das Lernen aber zu einem schnellen Stabilisieren des Pendels und einer Konvergenz innerhalb von ca. 5 sek. Die Regelbewegung erfolgt dann mit einer auf ca. $0,7^\circ$ reduzierten Amplitude, was bei der Prozesscharakteristik, insbesondere mit der Lose im Arbeitspunkt, nicht schlecht ist. Bemerkenswert ist auch, dass das Verhalten trotz des sehr einfachen Adaptiongesetzes nicht durch fortgesetztes Lernen schlechter wird, die Konvergenz also stabil ist.

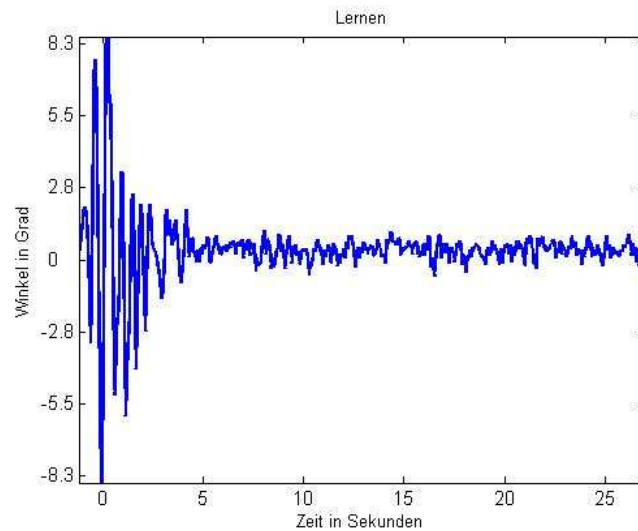


Abb. 5: Verhalten des Winkelreglers mit Lernen ohne den SILKE-Ansatz

4 Lernen mit dem SILKE-Ansatz

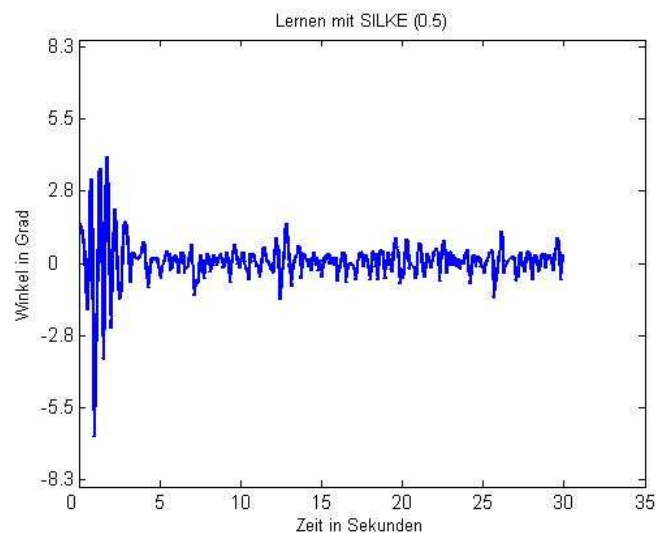


Abb. 6: Verhalten des Winkelreglers mit Lernen und dem SILKE-Ansatz bei einer Anpassungsrate von 0,5

Der SILKE-Ansatz greift beim Lernprozess im Winkelregler nach jedem Lernschritt ein. Das lokal mittelnde Template beeinflusst das Lernen in Richtung einer glatten Kennfläche. Bei der in Abb. 5 verwendeten Anpassungsrate von 0,5 führt das zu einer noch schnelleren Konvergenz, wie Abb. 6 zeigt. Die Wirkung des SILKE-Ansatzes führt auch zu einer deutlichen Reduktion der größeren Ausschläge in der Einschwingphase zu Beginn des Lernvorgangs und zu einer kleineren Regelbewegung danach (Amplitude ca. $0,5^\circ$). Auch hierbei ist das gelernte Verhalten stabil, weist aber ab und zu etwas größere Regelbewegungen auf, die durch das Zusammenspiel mit der Positionsregelung entstehen können. Das Regelverhalten sieht so auf den ersten Blick

praktisch wie vorher aus. Die Unterschiede werden aber deutlicher, wenn der dargestellte Zeitraum von 30 sek. mit integralen Maßen betrachtet wird.

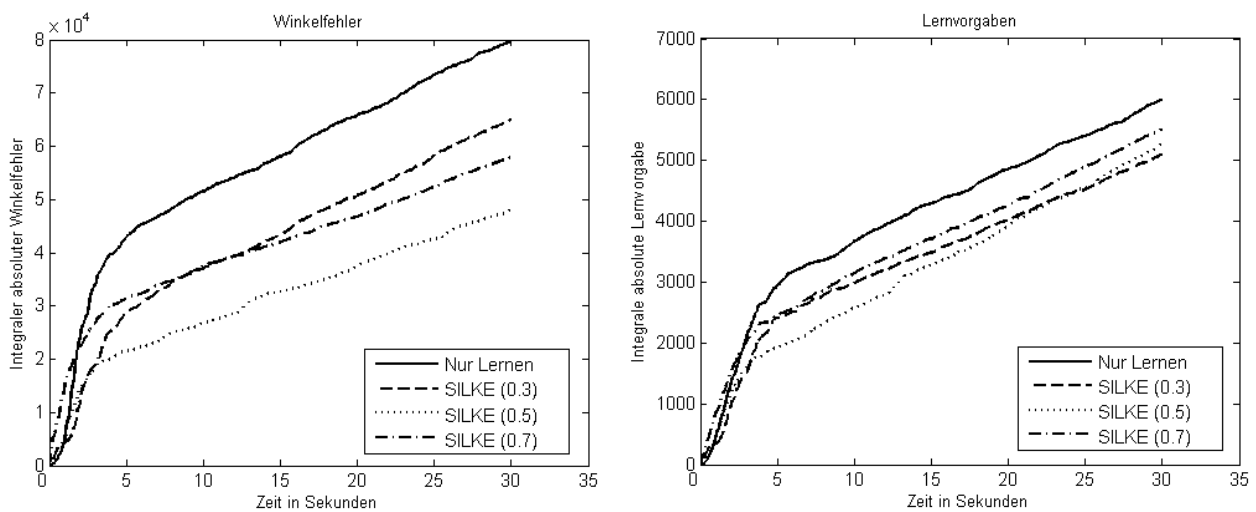


Abb. 7: Integraler absoluter Winkelfehler (links) und effektive Lernvorgabe (rechts) für verschiedene Anpassungsraten

Abb. 7 (links) zeigt das Integral des Betrags des Winkelfehlers über der Zeit exemplarisch für verschiedene Anpassungsraten. Das vermittelt einen Eindruck davon, wie der SILKE-Ansatz auf das Regelverhalten einwirkt. Ohne den SILKE-Ansatz (Anpassungsrate 0) entspricht das dem in Abb. 4 gezeigten Verhalten. Hier führen die Regelbewegungen zu einem raschen Anstieg des integralen Winkelfehlers und zu starken Abweichungen vom Idealverhalten. Sie bewirken auch starke Lerneingriffe des Hypersystems, die ebenfalls sehr schnell ansteigen, wie Abb. 7 (rechts) zeigt. Nach einer Einschwingphase von ca. 5 sek., was 500 Abtastungen entspricht, steigt der Winkelfehler deutlich langsamer. Das ist ein Indiz dafür, dass das gelernte Wissen im Groben weitestgehend konvergiert ist. Analog dazu reduziert sich die Steigung auch für das Integral der absoluten Lernvorgaben und damit die Änderungen am Winkelregler nach ca. 5 sek. ebenfalls. Die verbleibende Steigung ist darauf zurückzuführen, dass nun ständig Änderungen im Kleinen, also um den Arbeitspunkt herum, durchgeführt werden. Das lernende System konvergiert also nicht stabil gegen einen festen Winkelregler.

Der SILKE-Ansatz reduziert sowohl den integralen Winkelfehler als auch die Lernvorgaben deutlich. D. h., das System konvergiert schneller und in der Regel auch besser, solange die Anpassungsrate richtig gewählt ist. Wird sie zu groß, fehlt dem lernenden System die Flexibilität sich schnell ein- bzw. umzustellen, weil durch das mittelnde Template eine glatte Kennfläche erzwungen wird. Diese Flexibilität könnte hier aber nötig sein, weil die Reibungskompensation nur grob voreingestellt wurde, aber gerade im Arbeitspunkt wirkt. Eine suboptimale Einstellung würde einen Sprung im Arbeitspunkt erfordern, der aber dem Template widersprechen würde. Für eine Anpassungsrate von 0,5 wird insgesamt eine günstige Wirkung erzielt. Denn das Lernen konvergiert schnell und weist auch nach der Einschwingphase die geringste Regelbewegung auf. Auch die Änderungen am lernenden System fallen für eine Anpassungsrate von 0,5 über eine große Zeitspanne am geringsten aus.

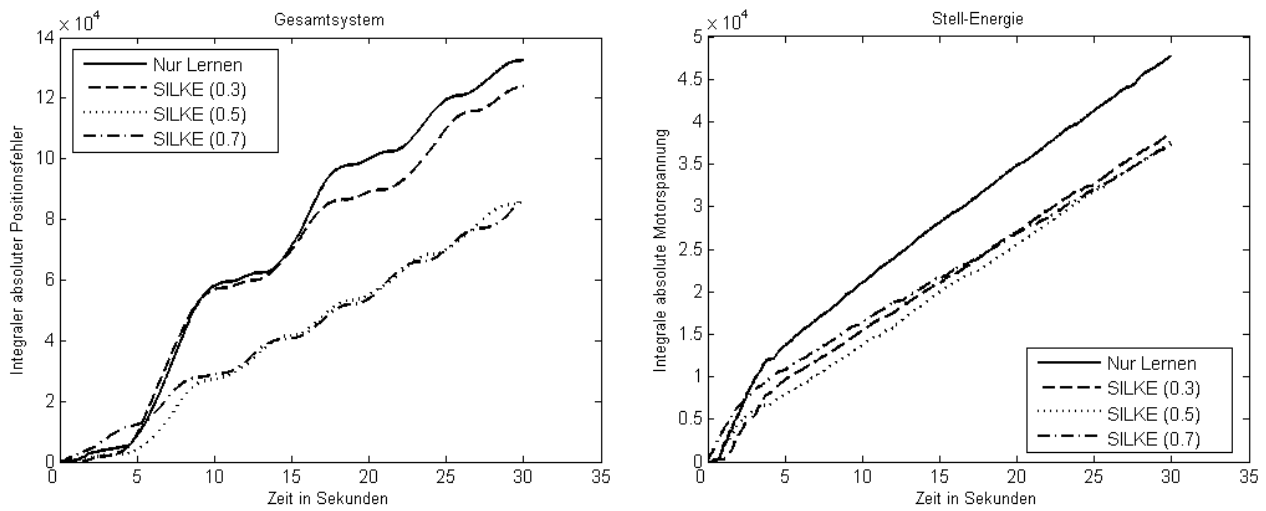


Abb. 8: Integraler absoluter Positionsfehler (links) und Stellgröße (rechts) für verschiedene Anpassungsraten

Eine Anpassungsrate von 0,5 stellt auch einen guten Kompromiss für das Systemverhalten im Großen dar, d. h. für den Pendelwagen als Ganzes, wie Abb. 8 zeigt. Die integralen absoluten Positionsänderungen fallen hierbei recht gering aus (Abb. 8 links) und es wird auch vergleichsweise wenig Stellenergie verbraucht (Abb. 8 rechts). Das bedeutet auch weniger Stress für die Aktoren und ein ruhigeres Verhalten des Gesamtsystems.

5 Lernen mit dem SILKE-Ansatz und dynamischer Veränderung der Lernrate

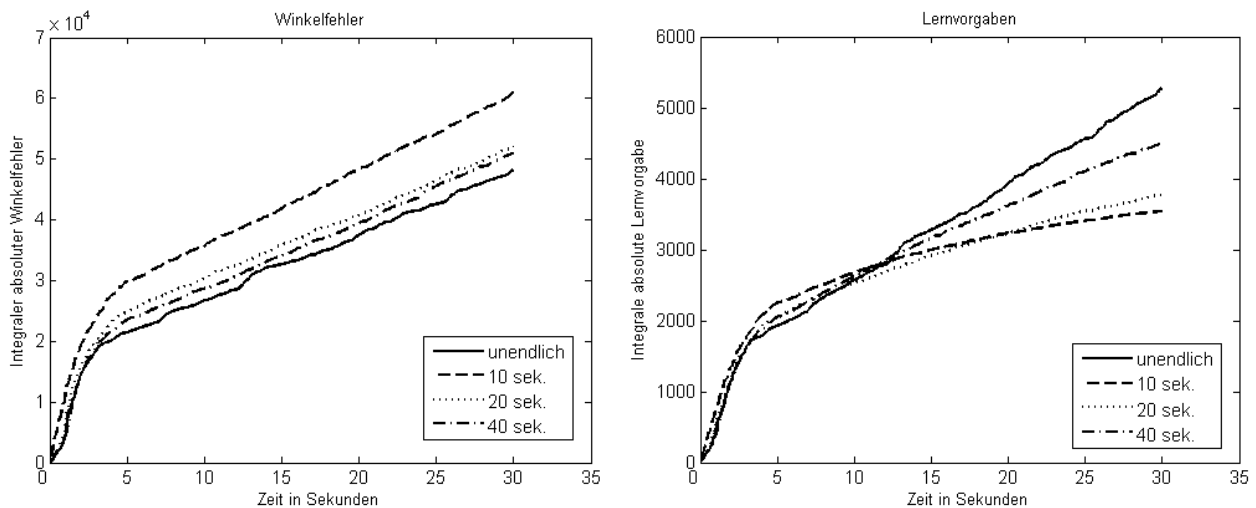


Abb. 9: Integraler absoluter Winkelfehler (links) und effektive Lernvorgabe (rechts) für eine Anpassungsrate von 0,5 und einer Reduktion der Lernrate mit verschiedenen Halbwertszeiten

Allerdings ist bei der Anpassungsrate von 0,5 eine größere Steigung bei der integralen Lernvorgabe zu verzeichnen als bei anderen Anpassungsraten. D. h., die Kennfläche des Winkelreglers wird stärker hin und her geändert als bei anderen Einstellungen. Die Stabilität des gelernten Wissens ist also geringer. Außerdem ist das Gesamtsystem unruhiger und verbraucht auf lange Sicht mehr Stellenergie als bei einer Anpassungsrate von 0,7 (und höher), wie Abb. 8 (rechts) zeigt. Deshalb wird hier untersucht, den SILKE-Ansatz zwar mit einer Anpassungsrate von 0,5 zu betreiben, um eine gute Konvergenzgeschwindigkeit zu erzielen, aber die Lernrate dynamisch zu reduzieren, um so die Langzeitstabilität des gelernten Wissens zu verbessern, ohne die Konvergenzgeschwindigkeit zu verschlechtern.

Die Lernrate wird beginnend bei 1 exponentiell über der Zeit reduziert. Zur Charakterisierung wird die Zeit angegeben, nach der die Lernrate auf die Hälfte gesunken ist (Halbwertszeit). Abb. 9 (links) zeigt die Entwicklung des integralen absoluten Winkelfehlers für verschiedene Halbwertszeiten bei einer Anpassungsrate von 0,5 im Vergleich zu einer konstanten Lernrate von 1 (Halbwertszeit unendlich), was der vorhergehenden Untersuchung entspricht. Generell ist eine Verschlechterung festzustellen. Insbesondere bei kurzen Halbwertszeiten, also einer schnellen Reduktion der Lernrate, hat das lernende System zu wenig Gelegenheit sich schnell auf den Prozess einzustellen (Konvergenzgeschwindigkeit). Das ist am steileren Anstieg und höheren Verlauf zu erkennen. Auch im weiteren Lernverlauf sind die Möglichkeiten das auszugleichen beschränkt (Konvergenzgüte). Für hinreichend große Halbwertszeiten (ab ca. 20 sek.) scheint aber das Lernen so gut zu konvergieren, dass schnell ein gutes Regelverhalten erreicht wird. Denn der Winkelfehler steigt nach der Einschwingphase von ca. 5 sek. praktisch mit der gleichen Steigung wie ohne Reduktion der Lernrate und bleibt auf einem etwas höheren Niveau.

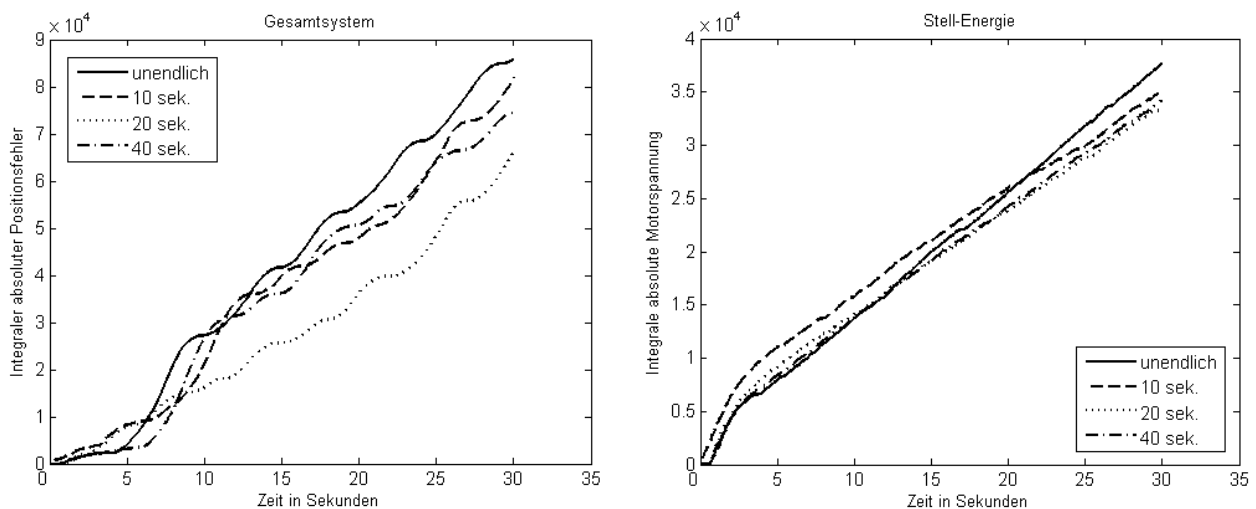


Abb. 10: Integraler absoluter Positionsfehler (links) und Stellgröße (rechts) für eine Anpassungsrate von 0,5 und einer Reduktion der Lernrate mit verschiedener Halbwertszeit

Abb. 9 (rechts) zeigt, dass der Reduktion der Lernrate in der Einschwingphase durch erhöhte Lernvorgaben entgegengewirkt wird, weil sich größere Regelfehler ergeben. Das ist ein Indiz dafür, dass die Wahl der Halbwertszeit nicht so kritisch ist, zumindest was das Lernverhalten betrifft. Abb. 10 zeigt aber, dass das Gesamtsystem für eine Halbwertszeit von 20 sek. das beste Verhalten aufweist. Die Bewegung des Wagens ist

für diesen Fall am geringsten (Abb. 10 links) und auch die Stellenergie hat den günstigsten Verlauf (Abb. 10 rechts).

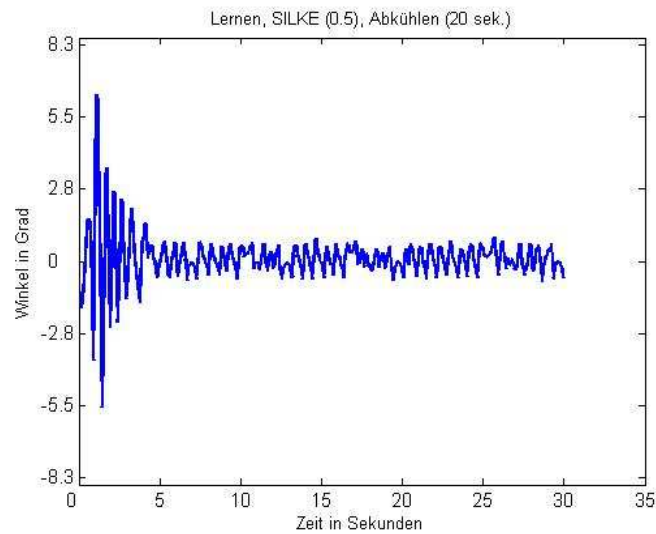


Abb. 11: Verhalten des Winkelreglers mit Lernen und dem SILKE-Ansatz bei optimierter Anpassungsrate von 0,5 und einer Reduktion der Lernrate mit einer Halbwertszeit von 20 sek.

Der Vergleich von Abb. 11 und Abb. 6 zeigt, dass durch die Kombination aus SILKE-Ansatz und dynamischer Reduktion der Lernrate sowohl die Konvergenzgeschwindigkeit als auch das Konvergenzgüte nach dem Abklingen der Einschwingphase weiter verbessert wurden.

6 Diskussion

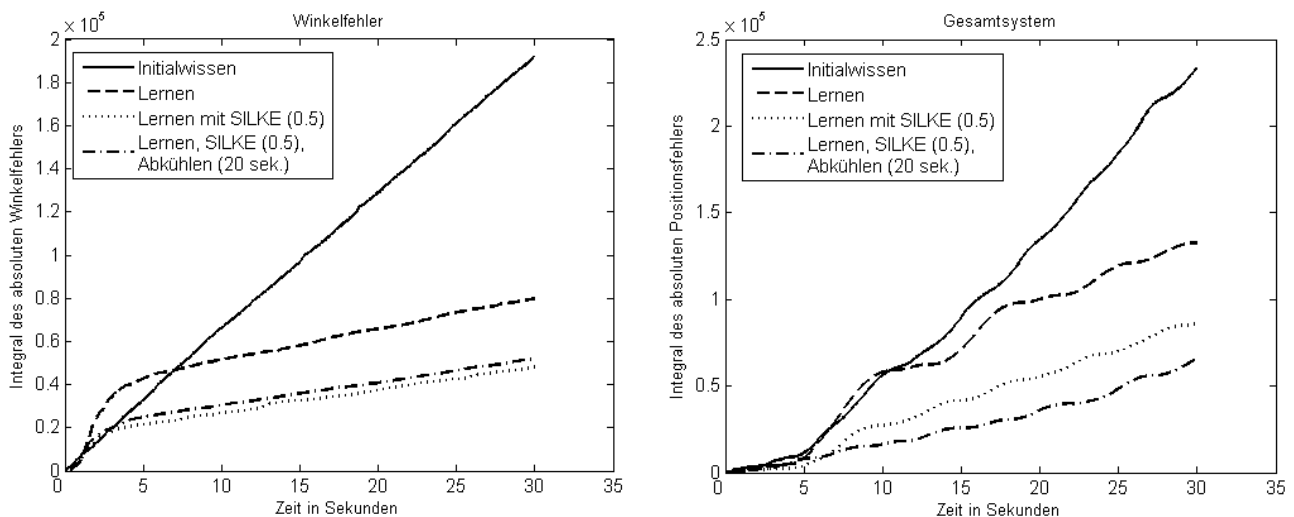


Abb. 12: Integraler absoluter Winkelfehler (links) und Positionsfehler (rechts) für die verschiedenen Szenarien

Die vorangegangenen Untersuchungen haben gezeigt, dass eine Kontrolle des dynamischen Lernvorgangs als solches möglich und sinnvoll ist. Sie beeinflusst das Systemverhalten im Kleinen (Winkelregelung, Abb. 12 links) und im Großen (z. B. Wagenbewegung und Stellaktionen, Abb. 12 rechts) positiv. Abb. 12 zeigt aber, dass nicht gleichzeitig ein optimales Verhalten im Großen wie im Kleinen erreicht wird. Die Entwicklung des Winkelfehlers ist beim Einsatz des SILKE-Ansatzes mit einer dynamischen Reduktion der Lernrate geringfügig schlechter als ohne. Das kann an zufälligen Komponenten in der Einschwingphase liegen. Die Kombination aus beiden Verfahren zur Kontrolle des Lernvorgangs erweist sich aber im Großen, also bei der Bewegung des Wagens, deutlich besser als nur der SILKE-Ansatz für sich, so dass insgesamt diese Kombination hier die besten Ergebnisse liefert.

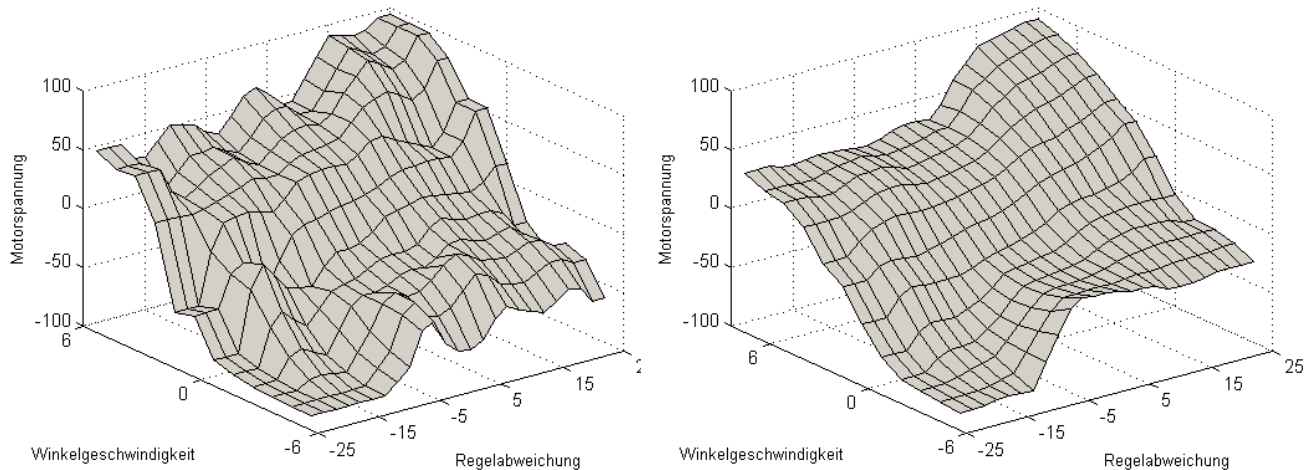


Abb. 13: Kennflächen des Winkelreglers für das Lernen ohne Kontrolle des Lernvorgangs (links) und mit einer Anpassungsrate von 0,5 und einer Reduktion der Lernrate mit einer Halbwertszeit von 20 sek. (rechts)

Die positive Wirkung dieser Kombination spiegelt sich auch in den sich durch das Lernen einstellenden Kennflächen des lernenden Winkelreglers wider. Ohne Kontrolle des Lernvorgangs ist die Kennfläche des Winkelreglers nach der oben betrachteten Lerndauer von 30 sek. sehr kraus und im Nullpunkt (Arbeitspunkt) recht steil (Abb. 13 links). Außerdem weist sie eine Reihe von Nichtmonotonien auf. Einige Randbereiche sind durch den Lernvorgang nicht oder nur kaum angepasst worden.

Der SILKE-Ansatz bewirkt durch das mittelnde Template auch eine Anpassung des lokalen Umfelds der Lernvorgabe und damit auch eine gewisse lokale Extrapolation. Zusammen mit seiner ausgleichenden Wirkung wird so eine deutlich glattere Kennfläche auch in den Randbereichen erzeugt (Abb. 13 rechts). Die Reduktion der Lernrate verstärkt und stabilisiert diesen glättenden Effekt nur noch. Insgesamt ergibt sich dadurch auch aus regelungstechnischer Sicht ein deutlich besseres Verhalten.

7 Zusammenfassung, Ausblick

In Anbetracht der Tatsache, dass Lernen online und closed-loop zu einem potentiell chaotischen Systemverhalten führen kann, ist eine hohe Konvergenzgeschwindigkeit und damit Plastizität des lernenden Systems wichtig. Um auch eine gute Stabilität des

gelernten Wissens zu erreichen, wurden hier zwei Methoden kombiniert. Wie dieses Beispiel zeigt, führt allein der SILKE-Ansatz auch in der praktischen Anwendung an einem realen, nicht-trivialen Prozess zu einem schnelleren und besserem Konvergieren des gelernten Wissens. Es wird eine Verbesserung des Selbsteinstellungsprozesses erreicht, selbst wenn wie hier das lernende System in einen größeren Kontext eingebettet ist, durch es zu starken Interaktionen kommt.

Wie das einfache Adaptionsgesetz im Hypersystem zeigt (Abb.3 rechts), ist der Aufwand, der für die Integration des Lernens getrieben werden muss, relativ gering. Die Einstellung der Anpassungsrate beim SILKE-Ansatz und der Halbwertszeit bei der Reduktion der Lernrate gehen vergleichsweise schnell und gut beherrschbar vonstatten.

Die Einstellung der Anpassungsrate hat bei dem verwendeten mittelnden Template einen merkbaren Einfluss nicht nur auf die Konvergenz des Lernens, sondern natürlich auch auf das Systemverhalten im Großen. Eine zu große Anpassungsrate behindert den Lernvorgang, weil durch das mittelnde Template die Plastizität des Lernens zu stark eingeschränkt wird. Bei einer zu kleinen Anpassungsrate ist die stabilisierende Wirkung des SILKE-Ansatzes zu gering. Allerdings deuten auch andere Untersuchungen darauf hin, dass der Bereich zwischen diesen Extrema relativ groß und gutmütig ist. Die genaue Einstellung der Anpassungsrate ist also nicht so kritisch.

Um das gelernte Wissen weiter zu stabilisieren, scheint die Kombination mit einer dynamischen Reduktion der Lernrate ein viel versprechender Ansatz zu sein, um gleichzeitig sowohl eine hohe Konvergenzgeschwindigkeit als auch eine hohe Konvergenzgüte zu erreichen. Die hier dargestellten Untersuchungen zeigen das Potenzial zur Reduktion des Plastizitäts-Stabilitäts-Dilemmas für selbstoptimierende Systeme, das im SILKE-Ansatz und dieser Kombination steckt. Aber sie stellen erst den Anfang dar. Die Wirkung der globalen Parameter Anpassungsrate und Reduktionsgeschwindigkeit der Lernrate werden in weiter führenden Arbeiten insbesondere hinsichtlich ihres Zusammenwirkens untersucht, um die Grenzen und allgemein gültigere Einstellregeln abzuleiten. Eine weitere Fragestellung betrifft die Verwendung anderer Templates in dieser Kombination.

Danksagung

Dieses Projekt wurde durch die DFG im Rahmen des Schwerpunktprogramms *Organic Computing* unter BR 1980/1-1 unterstützt.

8 Literatur

- [1] Girosi, F.; Jones, M.; Poggio, T.: *Regularization theory and neural networks architectures*. Neural Computation 7(2), 1995, 219-269
- [2] Cucker, F.; Smale, S.: *Best Choices for Regularization Parameters in Learning Theory: On the Bias-Variance Problem*. Foundations of Computational Mathematics 2(4), 2002, 413-428
- [3] Hagiwara, K.: *Regularization learning, early stopping and biased estimator*. Neuralcomputing 48(1), 2002, 937-955
- [4] Diehl, C.; Cauwenberghs, G.: *SVM Incremental Learning, Adaptation and Optimization*. Proc. 2003 Int. Joint Conf. on Neural Networks, 2003, 2685-2690

- [5] Syed, N.A.; Liu, H.; Sung, K.K.: *Incremental learning with support vector machines*. Proc. Workshop on Support Vector Machines at Int. Joint Conf. Artificial Intelligence - IJCAI-99, 1999
- [6] Atkeson, C.; Moore, A.; Schaal, S.: *Locally Weighted Learning*. Artificial Intelligence Review 11(1), 1997, 11-73
- [7] Schaal, S.; Atkeson, C.: *Constructive Incremental Learning from Only Local Information*. Neural Computation 10(8), 1998, 2047-2084
- [8] Jang, J.; Sun, C.; Mizutani, E.: *Neuro-fuzzy and soft computing*. Prentice Hall Upper Saddle River, NJ, 1997
- [9] Brockmann, W.: *Online Machine Learning for Adaptive Control*. In: Dillon, T.S.; Zurawski, R. (eds.): *Emerging Technologies in Factory Automation*, CRL Publishing, London, 1992, 190-195
- [10] Brockmann, W.; Meyer, D.M., Horst, A.: *Ein immunsystem-inspirierter Ansatz zum Überwachen des Lernens in Neuro-Fuzzy-Systemen*. In: Mikut, R.; Reischl, M. (eds): Proc. 16. Workshop „Computational Intelligence“, Universitätsverlag Karlsruhe, Karlsruhe, ISSN 1614-5267, 2006, 277-290
- [11] Brockmann, W.; Horst, A.: *Stabilizing the Convergence of Online-Learning Neuro-Fuzzy Systems by an Immune System-inspired Approach*. IEEE Int. Conf. On Fuzzy Systems - FUZZ-IEEE 2007, London, 23.-26.7.2007, 351-356
- [12] Rosemann, N.; Brockmann, W.: *Concept for Controlled Self-optimization in Online Learning Neuro-fuzzy Systems*. In: Hertzberg, J.; Beetz, M.; Englert, R. (eds.): *KI 2007: Advances in Artificial Intelligence*, LNCS SL 7, Springer, Berlin, Heidelberg, New York, 2007, 498-501
- [13] Brownlee, J.: *The Pole Balancing Problem - A Review of a Benchmark Control Theory Problem*. Technical Report ID: 7-01, Centre for Intelligent Systems and Complex Processes (CISCP), Swinburne University of Technology; Victoria, Australia, 2005
- [14] Hougen, D.; Fischer, J.; Johnam, D.: *A neural network pole balancer that learns and operates on a real robot in real time*. Proc. MLC-COLT Workshop on Robot Learning, 1994, 73-80

Sichere Exploration für Reinforcement-Learning-basierte Regelung

Alexander Hans^{1,2}, Daniel Schneegaß¹, Anton M. Schäfer¹ und
Steffen Udluft¹

¹ Siemens AG, Corporate Technology, Information & Communications,
Learning Systems, Otto-Hahn-Ring 6, 81739 München

² Technische Universität Ilmenau, FG Neuroinformatik & Kognitive Robotik,
Postfach 100565, 98684 Ilmenau

Zusammenfassung

In dieser Arbeit definieren und adressieren wir die Problemklasse der sicheren Exploration im Kontext von Reinforcement Learning. Wir führen die Konzepte einer Sicherheitsfunktion, die den Grad der Sicherheit eines Zustands bestimmt, und die einer Rückkehr-Strategie, die das System aus einem kritischen Zustand wieder in einen sicheren überführen kann, ein. Wenn diese beiden Komponenten gegeben sind, dann lässt sich das Problem der sicheren Exploration lösen. Darüberhinaus stellen wir ein level-basiertes Explorationsschema vor, das unter Einhaltung von Sicherheitskriterien eine möglichst umfassende Datenbasis bereitstellt. Wir untersuchen diesen Ansatz exemplarisch an einer vereinfachten Simulation einer Gasturbine und zeigen, dass diese erfolgreich sicher exploriert werden kann.

1 Einleitung

Reinforcement Learning (RL) [1] ist ein Ansatz des Maschinellen Lernens zur Lösung von Optimalsteuerungsproblemen. Das Ziel besteht darin, eine Aktionsauswahlregel, im weiteren Verlauf Policy genannt, zu identifizieren, die einen Agenten optimal innerhalb einer vorgegebenen Umgebung steuert. Diese ist definiert als Markov-Entscheidungsprozess (MDP) [1], welcher aus einem Zustandsraum \mathcal{S} , einer Menge von Aktionen \mathcal{A} und einer Dynamik besteht. Letztere ergibt sich wiederum aus den Transitionswahrscheinlichkeiten $\mathcal{P}_{s,s'}^a : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$, die vom aktuellen und dem Folgezustand sowie der Aktion, die zum Folgezustand führt, abhängen. Bei jeder entsprechenden Transition erhält der Agent einen sogenannten Reward $\mathcal{R}_{s,s'}^a$. Die Rewards folgen ebenfalls einer Verteilung und sind nur im Spezialfall deterministisch, so dass \mathcal{R} eine Funktion darstellt. In der Regel ist man daran interessiert, die diskontierende Value Function (Bewertungsfunktion)

$$V^\pi(s) = \mathbf{E}_s^\pi \left(\sum_{i=0}^{\infty} \gamma^i \mathcal{R}_{s^{(i)}, s^{(i+1)}}^{\pi(s^{(i)})} \right)$$

mit Diskontierungsfaktor $0 < \gamma < 1$, für alle Zustände s über den Policy-Raum $\pi \in \Pi$, $\pi : \mathcal{S} \rightarrow \mathcal{A}$ zu maximieren. Als Zwischenschritt bedient man sich sogenannter Q-Funktionen

$$Q^\pi(s, a) = \mathbf{E}_{s'}^\pi (\mathcal{R}_{s,s'}^a + \gamma Q^\pi(s', \pi(s'))),$$

die von Zustand und Aktion abhängen. Wir definieren $V^* = V^{\pi_{\text{opt}}}$ als die optimale Value Function und entsprechend

$$Q^*(s, a) = \mathbf{E}_{s'} (\mathcal{R}_{s,s'}^a + \gamma V^*(s')) = \mathbf{E}_{s'} \left(\mathcal{R}_{s,s'}^a + \gamma \max_{a'} Q^*(s', a') \right)$$

als die optimale Q-Funktion. Die Lösung dieser Bellman-Optimalitäts-Gleichung [2] liefert die optimale Policy $\pi(s) = \arg \max_a Q^*(s, a)$.

Simulationen haben gezeigt, dass dateneffiziente RL-Verfahren in der Lage sind, komplexe technische Anlagen, wie etwa Gasturbinen, anhand einer vertretbaren Anzahl von Interaktionen zu optimieren [3]. Sobald der RL-Agent jedoch beginnt, mit einer realen im operativen Betrieb genutzten Anlage zu interagieren, stellt sich die Frage, inwieweit sich sicher stellen lässt, dass die Exploration, also die Erkundung des Zustandsraums, die Anlage nicht beschädigt. Insbesondere besteht die Forderung, dass durch die Exploration die Betriebssicherheit nur in vertretbarem Maße verringert wird.

Konventionelle Explorations-Strategien, wie beispielsweise ε -greedy- oder Boltzmann-Exploration [1], enthalten eine Zufallskomponente, die dazu führen kann, dass die Anlage in einen fatalen Zustand gelangt. Sie sind daher in unveränderter Form für eine sichere Exploration ungeeignet.

Im Folgenden werden zwei Komponenten vorgestellt, die für die sichere Exploration erforderlich sind. Dabei handelt es sich um eine *Sicherheitsfunktion* zur Bestimmung des Grades der Sicherheit eines Zustands und eine *Rückkehr-Strategie* (Backup-Policy), die das System aus einem kritischen (aber noch nicht überkritischen) Zustand wieder in einen sicheren Zustand überführen kann. Auf diese Weise wird das Problem der sicheren Exploration grundlegend gelöst. Ein zusätzliches level-basiertes Explorationsschema garantiert eine umfassende Datenbasis und damit Ergebnisse hoher Qualität.

2 Sichere Exploration

Im RL existieren vielfältige Konzepte von Sicherheit und Unsicherheit [4, 5]. Beispielsweise kann sich Unsicherheit auf die Varianz der Transitionswahrscheinlichkeiten oder eines Schätzers der Value Function beziehen. In der vorliegenden Arbeit betrachten wir Unsicherheit jedoch im Sinne nicht wünschenswerter Zustände oder, allgemeiner, zu vermeidender Transitionen. Eine besondere Schwierigkeit entsteht dadurch, dass der MDP zu Beginn der Exploration unbekannt ist, was Unkenntnis über den Sicherheitsstatus einzelner Zustände mit einschließt.

Diese Art von Unsicherheit wurde bereits unter verschiedenen Aspekten untersucht. Wir verweisen insbesondere auf Heger [6], in dem ein zum normalerweise im RL verwendeten Entscheidungskriterium, dem Erwartungswert, alternatives Kriterium vorgestellt wird, das sogenannte *Minimax-Kriterium*. Darauf aufbauend wird der RL-Algorithmus \hat{Q} -Learning entwickelt, der anstelle des Erwartungswertes das Minimax-Kriterium nutzt. Das Verfahren bestimmt eine Policy, deren minimaler Reward maximiert wird. In deterministischen Problemen kollabiert das Verfahren zum konventionellen Q-Learning [7].

Neuneier und Mihatsch kritisieren in [8] das Minimax-Kriterium als zu restriktiv, weil es sehr seltene Ereignisse berücksichtigt, obwohl diese in der Praxis nahezu niemals auftreten. Als Alternative dazu schlagen sie eine neue Familie von Optimalitätsgleichungen vor, die mittels eines Parameters κ bezüglich ihrer „Risikobereitschaft“ beeinflusst werden können.

Geibel stellt in [9] ein RL-Verfahren vor, das die Existenz von *fatalen* oder *unerwünschten* Zuständen vorsieht. Bestimmt wird schließlich die optimale Value Function, jedoch unter der Nebenbedingung, dass das Risiko einen fatalen Zustand zu erreichen einen bestimmten Wert nicht übersteigt.

Alle diese Arbeiten verändern jedoch ausschließlich die Zielfunktion. Sie berücksichtigen nicht, dass bereits während der Exploration unsichere Zustände vermieden

werden müssen. Die Arbeiten zum Apprenticeship Learning, z.B. [10], befassen sich dagegen mit dem Thema der Exploration. Die grundlegende Idee des Apprenticeship Learning besteht darin, keine explizite Exploration durchzuführen, sondern stattdessen von einem Lehrer generierte Trajektorien zu beobachten und anschließend zu nutzen, um ein Systemmodell zu lernen und für dieses eine optimale Policy zu bestimmen. Diese Policy wird dann auf dem realen System evaluiert, die von ihr generierten Beobachtungen ebenfalls gespeichert und im Anschluss genutzt, um eine verbesserte Approximation des Systemmodells zu lernen, das wiederum als Grundlage einer neuen optimalen Policy dient. Dies wird so lange wiederholt, bis die Performance der gelernten Policy mindestens so gut ist wie die des Lehrers. Nachteilig ist, dass der Lehrer nicht nur sicher, sondern auch möglichst gut sein muss, da aufgrund der fehlenden expliziten Exploration nicht zu erwarten ist, dass Bereiche des Zustandsraums besucht werden, die weit von den Lehrer-Trajektorien entfernt sind. Außerdem ist es schwer möglich, eine Aussage über die Sicherheit bei diesem Ansatz zu treffen. Bewegt sich der Lehrer am Rand des sicheren Bereichs, so kann der Agent bei der „unfreiwilligen Exploration“ in einen unsicheren Bereich gelangen.

Zur besseren Abgrenzung von den noch folgenden Definitionen bezeichnen wir in der vorliegenden Arbeit unerwünschte Zustands-Aktions-Paare $\mathcal{U} \subset \mathcal{S} \times \mathcal{A}$ als fatal. Welche Zustands-Aktions-Paare fatal sind, ist vom jeweiligen Problem abhängig. Bei der Regelung einer Gasturbine beispielsweise darf die Dynamik der Brennkammer nicht zu stark werden, weil diese sonst beschädigt wird. Die sicherheits-optimale Policy soll nun fatale Transitionen vermeiden und, unter dieser Nebenbedingung, den Erwartungswert der (diskontierten) Summe der künftigen Rewards maximieren. Durch Abändern der Reward-Funktion nach

$$\hat{\mathcal{R}}_{s,s'}^a = \begin{cases} \mathcal{R}_{s,s'}^a & : (s, a) \notin \mathcal{U} \\ -\infty & : (s, a) \in \mathcal{U} \end{cases}, \quad (1)$$

wird die Sicherheitsfunktion in die Struktur der Rewards integriert und die optimale Policy wird, per Konstruktion, zur sicherheits-optimalen. Wir definieren Unsicherheit im Folgenden durch einen Grenzwert τ in den Rewards, der nicht unterschritten werden darf. Der allgemeine Fall (Gl. 1) lässt sich offensichtlich auf diesen Fall zurückführen. Konventionelle Explorationstechniken wie zufällige, ε -greedy- oder Boltzmann-Exploration [1] sind grundsätzlich nicht sicher, da sie per Konstruktion die mögliche Fatalität von Zuständen nicht berücksichtigen. Trotz synchronen Lernens der optimalen Policy besteht nach wie vor eine gewisse Wahrscheinlichkeit, dass unsichere Zustände exploriert werden. Dies ist insbesondere dann nicht vermeidbar, wenn die optimale Policy eine unsichere Policy ist. Damit ein RL-Problem überhaupt sicher explorierbar ist, stellen wir zwei Voraussetzungen an das Problem:

1. Die Exploration muss in einem sicheren Zustand starten.
2. Der Grad der Sicherheit muss mindestens dreistufig sein, z.B. „sicher“, „kritisch“, „überkritisch“, um direkte Übergänge von sicheren in fatale Zustände zu vermeiden. Er folgt direkt aus der Höhe der Rewards.

3 Komponenten der sicheren Exploration

Unter den oben genannten Voraussetzungen an einen sicher explorierbaren MDP sind zwei wesentliche Bausteine für eine sichere Exploration nötig: eine *Sicherheitsfunktion* und eine *Backup-Policy*. Die Sicherheitsfunktion hat die Aufgabe zu bestimmen, ob ein Zustands-Aktions-Paar sicher ist, während die Backup-Policy

Nicht fatale Transition	$\mathcal{R}_{s,s'}^a \geq \tau$
Fatale Transition	$\mathcal{R}_{s,s'}^a < \tau$
Sicherer Zustand	$\forall a \in \mathcal{A} : \exists \pi : \forall i : \mathcal{R}_{s^{(i)},s^{(i+1)}}^{\pi(s^{(i)})} \geq \tau$
Kritischer Zustand	$\exists a \in \mathcal{A} : \mathcal{P}_{s,s'}^a > 0$ mit s' überkritisch
Überkritischer Zustand	$\forall a \in \mathcal{A} : \forall \pi : \exists i : \mathcal{R}_{s^{(i)},s^{(i+1)}}^{\pi(s^{(i)})} < \tau$

Tab. 1: Definition der Sicherheitsbegriffe

von kritischen oder unbekanntenen Zuständen in sichere und bereits bekannte zurückführen soll. Um diese beiden Komponenten zu verbinden, wird ein level-basiertes Explorations-Schema vorgestellt.

3.1 Sicherheitsfunktion

Die Sicherheitsfunktion muss für ein Zustands-Aktions-Paar eine Aussage über ihren Sicherheitsstatus treffen: *sicher*, *kritisch*, *überkritisch*. Das Ziel der Exploration ist dabei, überkritisch niemals zu beobachten, da überkritische Zustände zwangsläufig in fatale Zustände führen. Aus kritischen Zuständen kann der Agent, bei sicherem Verhalten, wieder in sichere Zustände überführt werden. Wird vor dem Ausführen einer Aktion diese als sicher eingestuft, kann sie in der Exploration ausgeführt werden. Ist in einem Zustand keine zu explorierende Aktion sicher, muss dafür gesorgt werden, dass von diesem in einen sicheren Zustand zurückgekehrt wird.

Die triviale Lösung des Problems der Sicherheitsfunktion besteht darin, sie im Vorhinein zu definieren. Dies setzt aber detailliertes Wissen über die RL-Umgebung voraus, das oftmals nicht vorhanden ist. Daher ist es erforderlich, die Sicherheitsfunktion aus bereits vorhandenen Explorationsdaten (Beobachtungen von Zustandsübergängen in Form von (s, a, r, s') -Tupeln) zu lernen.

Eine unsichere Aktion ist dadurch definiert, dass mittelbar nach deren Ausführen ein fataler Zustand erreicht wird. Beim Betreten eines solchen Zustands wird ein Reward kleiner τ beobachtet. Eine Möglichkeit der Umsetzung der Sicherheitsfunktion ist die *Minimumschätzung des Rewards* (Min-Reward-Schätzung). Diese besteht darin, den minimalen Reward zu schätzen, den man erhält, nachdem man in einem Zustand s Aktion a ausgeführt hat und der Backup-Policy folgt.

1. Während der Exploration werden Min-Reward-Samples der Form (s, a, r_{\min}) gesammelt. Dabei bezeichnet (s, a) das Ausführen von Aktion a in Zustand s und r_{\min} den minimalen beobachteten Reward auf der Backup-Trajektorie, die im Anschluss an diese Transition von der Backup-Policy generiert wird.
2. Auf Basis der während der Exploration gesammelten Samples wird ein Funktionsapproximator gelernt, der von einem Zustands-Aktions-Paar auf den minimalen zu erwartenden Reward abbildet: $R_{\min} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$.

Die Min-Reward-Samples hängen von der verwendeten Backup-Policy ab. Ähnlich wie die Werte einer Q-Funktion $Q^\pi(s, a)$ die erwartete (diskontierte) Summe der zukünftigen Rewards liefern, wenn man in Zustand s Aktion a ausführt und anschließend Policy π folgt, liefert $R_{\min}(s, a)$ den minimalen zu erwartenden Reward, wenn man in Zustand s Aktion a ausführt und danach der Backup-Policy folgt.

3.2 Backup-Policy

Die Backup-Policy hat die Aufgabe, den Agenten in einen bekannten Bereich zurückzuführen, wenn dieser keine abgesicherte Entscheidung mehr treffen kann, weil

er in einen neuen Zustand geraten ist, in dem er die Sicherheit einzelner Aktionen nicht hinreichend gut einschätzen kann. Sie darf dabei selbst nicht in kritische Zustände führen. Dies wird z.B. erreicht, indem die Backup-Policy einen eventuellen stationären Punkt anfährt oder versucht, sich in einem bestimmten Bereich des Zustandsraums aufzuhalten.

Für reale Systeme, die bereits ohne RL betrieben werden, ist oftmals bereits ein Regler vorhanden. Sofern sich dieser sicher verhält, ist er als vorweg spezifizierte Backup-Policy einsetzbar. Wenn ein solcher Regler nicht vorhanden ist, ist es notwendig, eine Backup-Policy aus bereits vorhandenen Explorationsdaten (also (s, a, r, s') -Tupeln) zu lernen.

Da wir Sicherheit über den Reward ausdrücken, muss eine aus Explorationsdaten gelernte Backup-Policy den Reward berücksichtigen. Ein naheliegender Ansatz besteht daher darin, die Backup-Policy mittels konventionellem RL zu bestimmen. Im Allgemeinen ist jedoch eine optimale Policy nicht gleichzeitig sicher. Darüberhinaus kann eine vermeintlich optimale und sichere Policy durch Fehler in der Generalisierung unsicher sein. Um dies jedoch abzusichern, wird anstelle des Erwartungswerts der *Summe* der Rewards der Erwartungswert des *Minimums* maximiert. Die entsprechende Bellman-Optimalitäts-Gleichung lautet dann

$$Q^{m*}(s, a) = \sum_{s' \in S} \mathcal{P}_{s, s'}^a \min \left[\mathcal{R}_{s, s'}^a, \max_{a' \in \mathcal{A}} Q^{m*}(s', a') \right]. \quad (2)$$

Auf den Diskontierungsfaktor γ verzichten wir, weil durch die Umformulierung der Zielfunktion das Problem einer divergierenden Summe nicht mehr besteht.

Wenn man für ein gegebenes RL-Problem ein Q^{m*} ermittelt, das diese Gleichung geeignet erfüllt, so lässt sich daraus eine bezüglich Q^{m*} greedy Policy¹ π^{m*} bestimmen, die bestrebt ist, niedrige Rewards zu vermeiden.

Eine solche Max-Min-Q-Funktion, die Gl. 2 erfüllt, lässt sich zum Beispiel mit Dynamischer Programmierung [2, 1] bestimmen wie auch die Q-Funktion der optimalen Policy. Jedoch existiert hier kein eindeutiger Fixpunkt, da etwa $Q = -\infty$ eine zwar unerwünschte, aber zulässige Lösung ist. Die Initialisierung mit $Q = \infty$ garantiert das Erreichen einer korrekten Lösung, da im ersten Iterationsschritt die Rewards als Minimum verwendet werden und nicht die initialen Q-Werte.

3.3 Level-basierte Exploration

In dem Fall, dass die Sicherheitsfunktion und/oder die Backup-Policy aus den Daten gelernt werden muss, ist man daran interessiert, eine möglichst breite Datenbasis zu gewinnen. Daher realisieren wir eine graduelle Exploration, die den Agenten vom als sicher bekannten Bereich langsam entfernt und zunächst versucht, möglichst viele lokale Beobachtungen zu sammeln.

Dazu wird jedem Zustand ein *Level* $l \in \mathbb{N}$ zugewiesen. Davon ausgehend, dass die Exploration im sicheren Bereich startet, wird für den Ausgangszustand s_0 das Level $l(s_0) = 0$ gesetzt. Wird während der Exploration ein bisher unbekannter Zustand s_i betreten, so wird sein Level als $l(s_i) = l(s_{i-1}) + 1$ gesetzt, wobei $l(s_{i-1})$ dem Level des Vorgängerzustands entspricht. Wird beim Verfolgen der Backup-Policy ein neuer Zustand s_i betreten, wird sein Level $l(s_i)$ von dem des Vorgängerzustands s_{i-1} übernommen: $l(s_i) = l(s_{i-1})$. Dies wird dadurch motiviert, dass das Verfolgen der Backup-Policy nicht zu einer Verschlechterung der Sicherheit führt. Auf diesem Level-Konzept aufbauend läuft die Exploration wie folgt ab:

¹Eine Policy ist greedy (gierig), falls sie in jedem Zustand die Aktion mit maximalem Q-Wert ausführt.

- Beginnend mit Level $l(s_0) = 0$ wird jedem Zustand ein Level zugewiesen.
- Es wird ein aktuell zu explorierendes Level l_c vorgehalten, wobei zu Beginn $l_c = 0$ initialisiert wird. Ziel der Exploration ist jeweils für ein Level l_c , in allen Zuständen, deren Level l_c entspricht, alle dort möglichen Aktionen auszuführen (sofern das Ausführen der Aktion als sicher eingeschätzt wird). Sind alle Zustände des aktuellen Levels vollständig exploriert, wird l_c erhöht.
- Die Exploration wird so lange fortgesetzt, bis ein Abbruchkriterium erfüllt ist, etwa die vollständige Exploration des gesamten sicher explorierbaren Bereiches. In der Praxis wird dies jedoch häufig kaum umsetzbar sein, weshalb das Erreichen einer gewünschten Mindestperformance zweckmäßiger ist.

Um auf diese Art und Weise explorieren zu können, braucht man eine Möglichkeit, Zustände gezielt anzufahren: Nachdem der Agent nach Erreichen eines neuen Zustands mit erhöhtem Level der Backup-Policy folgt, gelangt dieser früher oder später wieder in einen bekannten Zustand. Von dort aus muss die Möglichkeit bestehen, einen Zustand des aktuellen Levels zu erreichen.

3.3.1 Graph-basierte Wegfindung

Zum Auffinden eines beliebigen bekannten Zustands bietet sich bei deterministischen RL-Problemen ein graph-basierter Ansatz an. Diesem liegt die Idee zugrunde, während der Exploration einen Graphen aufzubauen, dessen Knoten Zustände und Kanten ausgeführte Aktionen repräsentieren. Neben dem Zustand ist an einem Knoten das Level des Zustands vermerkt.

- Wenn im aktuellen Zustand keine Aktion mehr zu explorieren ist, wird im Graph nach einem Zustand gesucht, dessen Level dem aktuell zu explorierenden entspricht und für den noch sicher explorierbare Aktionen existieren. Wird kein solcher Zustand gefunden, wird das aktuell zu explorierende Level erhöht oder die Exploration angehalten.
- Um einen Weg vom aktuellen Zustand zu einem zu explorierenden Zustand zu finden, wird eine Breitensuche auf dem Graphen durchgeführt. Durch das Ausführen der Aktionen, mit denen die Kanten entlang des gefundenen Weges beschriftet sind, gelangt man vom aktuellen Zustand zum Zielzustand.

Die Methode führt zu praktischen Problemen, wenn der MDP mehr als einige tausend Zustände umfasst. In vertretbarer Zeit können nur niedrige zweistellige Levels vollständig exploriert werden. Eine mögliche Lösung dieses Problems besteht in einer geeigneten Zusammenfassung von benachbarten Zuständen zu einem Cluster. Im Kontext des graph-basierten Ansatzes entspräche ein Cluster einem Knoten im Graphen. Leider wird dabei im Allgemeinen die Markov-Bedingung verletzt. Es muss daher ein geeigneter Kompromiss gefunden werden zwischen Aufwand und Lösbarkeit.

3.3.2 Wegfindung mittels Reinforcement Learning

Als alternativen Ansatz, der auch im stochastischen Fall eingesetzt werden kann, greifen wir auf das konventionelle RL zurück. Dazu wird ein MDP definiert, dessen Zustände den zuvor definierten Clustern entsprechen. Nun definiert man die Reward-Funktion $\mathcal{R}_{s,s'}^a = 1$, falls s' der angestrebte Zustand ist und $\mathcal{R}_{s,s'}^a = 0$ sonst. Für das so definierte RL-Problem ermittelt man eine optimale Policy und folgt dieser so lange, bis der gewünschte Cluster erreicht ist. Es ist auch möglich, mehrere Cluster

als Ziel zu definieren, in diesem Fall würde die Policy versuchen den zum aktuellen Cluster nächstgelegenen Cluster zu erreichen.

Algorithmus 1 beschreibt das gesamte Vorgehen der Wegfindung mittels RL. Die Aktualisierung der Policy wird darin berücksichtigt, ebenso die maximale Anzahl an Aktionen, nach denen das Verfahren (erfolglos) abgebrochen wird.

Algorithmus 1 : Reinforcement Learning zur Wegfindung.

```

Data :  $\mathcal{O}$  – Liste Beobachtungen als  $(s, a, s')$ -Tupel,  $\mathcal{Z}$  – Menge der Ziel-Cluster,  $n$  –
        Anzahl Aktionen, nach denen Policy zu aktualisieren ist,  $m$  – maximale Anzahl
        Aktionen, nach denen ein Ziel-Cluster erreicht worden sein muss
// Initialisierung
1  bestimme Reward-Funktion:  $\mathcal{R}_{s,s'}^a = \begin{cases} 1, & \text{falls } s' \in \mathcal{Z} \\ 0, & \text{sonst} \end{cases} \quad \forall (s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ ;
2  bestimme Schätzung  $T$  der Übergangswahrscheinlichkeiten aus  $\mathcal{O}$ ;
3  löse das durch  $T$  und  $\mathcal{R}$  definierte RL-Problem, erhalte daraus Policy  $\pi_w$ ;
// Policy-Anwendung zur Wegfindung
4   $c := 0$ ;
5  repeat
    // aktualisiere Policy nach  $n$  Schritten
6    if  $c > 0 \wedge c \bmod n = 0$  then
7      aktualisiere Schätzung  $T$  basierend auf  $\mathcal{O}$ ;
8      löse das durch  $T$  und  $\mathcal{R}$  definierte RL-Problem  $\rightarrow$  Policy  $\pi_w$ ;
9    end
10    $s :=$  aktueller Zustand;
11    $a := \pi_w(s)$ ;
12   führe  $a$  aus;
13    $s' :=$  aktueller Zustand;
14   hänge  $(s, a, s')$  an  $\mathcal{O}$  an;
15    $c := c + 1$ ;
16   if  $c = m$  then // maximale Anzahl Aktionen erreichen?
17     // Wegfindung wird erfolglos abgebrochen
18     return false;
19   end
20 until  $s' \in \mathcal{Z}$ ;
// Wegfindung war erfolgreich
21 return true;

```

4 Experimente

Die bisher vorgestellten Verfahren wurden am Beispiel des nachfolgend eingeführten Benchmark-Problems *BurnSim* implementiert und experimentell untersucht.

4.1 Benchmark-Problem *BurnSim*

Die *BurnSim* ist motiviert durch ein real auftretendes Problem bei der Regelung von Gasturbinen. Eine Turbine optimal zu betreiben heißt, sie mit möglichst hoher Leistung zu fahren. Im Hochleistungsbereich einer Turbine kommt es jedoch zu einer

unerwünschten Dynamik in der Brennkammer, ein „Brummen“, das, wenn es zu stark wird, zu einer Beschädigung der Turbine führen kann. Aus diesem Grund ist das Ziel, die Gasturbine mit möglichst hoher Leistung zu betreiben, gleichzeitig aber darauf zu achten, dass das Brummen nicht zu stark wird.

Der Zustandsraum der *BurnSim* ist 2-dimensional:

$$\mathcal{S} := \{(f, h) \mid f \in [0, 1), h \in [0, \infty)\}$$

Dabei bezeichnet f *fuel*, was der Treibstoffzufuhr und somit direkt der Leistung der Turbine entspricht. h steht für *humming* und bezeichnet das Brummen. Es sind drei Aktionen erlaubt, Fuel verringern, beibehalten, erhöhen:

$$\mathcal{A} := \{\text{decrease, keep, increase}\}$$

Die Aktionen *decrease* und *increase* verändern dabei f um 0.05, jedoch muss f in den Grenzen $[0, 1)$ bleiben. Die Systemdynamik, also die Veränderungen des Zustands von einem Zeitschritt zum nächsten bei Ausführen von Aktion a , ist gegeben durch:

$$f_{t+1} = \begin{cases} f_t - 0.05, & \text{falls } a = \text{decrease} \wedge f_t - 0.05 \geq 0 \\ f_t + 0.05, & \text{falls } a = \text{increase} \wedge f_t + 0.05 < 1 \\ f_t, & \text{sonst} \end{cases}$$

$$h_{t+1} = \begin{cases} 2f_t h_t, & \text{falls } h_t > f_t/5 \\ f_t/5, & \text{sonst} \end{cases}$$

Die Reward-Funktion ist nur vom Folgezustand $s_{t+1} = (f_{t+1}, h_{t+1})$ abhängig und lautet:

$$r_t = 4f_{t+1}^2 - \left(\frac{h_{t+1} - f_{t+1}/5}{5} \right)^2$$

Der Grenzwert der Sicherheit wird gesetzt als

$$\tau = -1,$$

somit ergibt sich der Status der Sicherheit als

$$r_t \geq -1 \Rightarrow \text{Zustand } s_t \text{ ist fatal,}$$

$$r_t < -1 \Rightarrow \text{Zustand } s_t \text{ ist nicht fatal.}$$

Das Ziel besteht darin, die erwartete Summe der Rewards zu maximieren, ohne jemals einen Reward kleiner als -1 zu beobachten. Offenbar wird ein möglichst hoher Reward erreicht, wenn f möglichst hoch und gleichzeitig h möglichst niedrig ist. Abb. 1 veranschaulicht die Reward-Funktion. Die in dieser Grafik eingezeichnete Trajektorie erhält man, wenn man bei $f = 0$ beginnend wiederholt die Aktion *increase* ausführt. Wie man der Darstellung entnehmen kann, erhöht sich dabei auch der Reward. Allerdings kommt es auch zu einer Erhöhung des Brummens, was sich negativ auf den Reward auswirkt. Im Sinne einer sicheren Exploration wäre der Zustand am Ende der Trajektorie ($f = 0.95$; $h = 3.35$) bereits überkritisch, denn selbst wenn man Fuel sofort verringert, wird ein Reward von -138.45 nicht zu vermeiden sein, da das Brummen trotz der Fuel-Verringerung bis auf 59.14 steigt. Die Trajektorie markiert außerdem die Grenze des erreichbaren Bereichs des Zustandsraums. Der Bereich „unterhalb“ dieser Grenze kann aufgrund der Systemdynamik nicht erreicht werden. Durch diese Beispieltrajektorie wird eine grundlegende Schwierigkeit der sicheren Exploration bereits deutlich: Es muss ein Zustand als unsicher erkannt

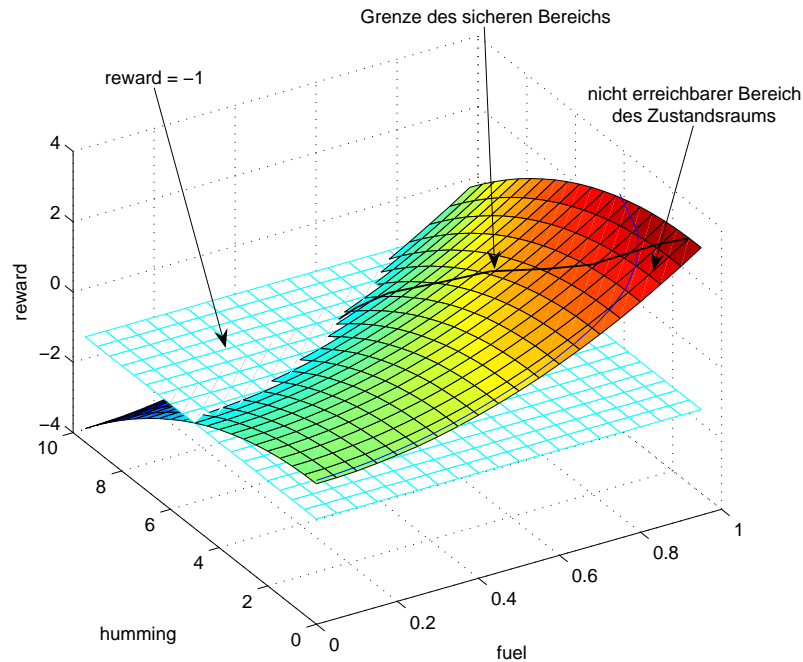


Abb. 1: Visualisierung der BurnSim-Reward-Funktion. Zusätzlich zur Reward-Funktion selbst ist die Grenzwert-Ebene eingezeichnet, unter die der Reward für einen sicheren Betrieb niemals fallen darf. Teile der Funktion werden nie erreicht, weil aufgrund der Systemdynamik hohes Fuel in Kombination mit niedrigem Brummen nicht möglich ist. Schwarz markiert ist die Grenze des sicheren Bereichs – bei höherem Fuel oder Brummen würde man zwangsläufig (später) in fatale Zustände gelangen.

werden, selbst wenn dieser zwangsläufig, aber nur mittelbar in einen fatalen Zustand führt. In Abb. 1 ist die Grenze des sicheren Bereichs schwarz markiert. Von jedem Zustand „oberhalb“ dieser Grenze, also bei höherem Fuel und/oder höherem Brummen, ist es nicht möglich, Rewards kleiner des Grenzwerts von -1 zu vermeiden. Sollte sich der Agent also während der Exploration dorthin bewegen, hat er das Ziel der sicheren Exploration verfehlt. Die *BurnSim* besitzt einen stationären Bereich bei $f = 0.5$, an dieser Stelle wird das Brummen nicht verändert. Wenn dort die Aktion *keep* ausgeführt wird, verändern sich weder Zustand noch Reward.

4.2 Überblick

In Abb. 2 ist eine Übersicht der für die Exploration nötigen Komponenten dargestellt. Bei der Sicherheitsfunktion, Explorations-Strategie und Backup-Policy sind verschiedene Varianten implementiert worden, die in der Darstellung entsprechend vermerkt sind. Der Agent steht im Zentrum der Exploration, er hält die einzelnen Komponenten zusammen und koordiniert das Zusammenspiel. Der Gesamtprozess der sicheren Exploration kann wie folgt zusammengefasst werden.

1. **Initialisierung.** Es wird davon ausgegangen, dass sich der Agent zu Beginn der Exploration in einem sicheren Bereich des Zustandsraums befindet. Die Liste der Beobachtungen wird als leer initialisiert, ebenso die Menge der Zustände

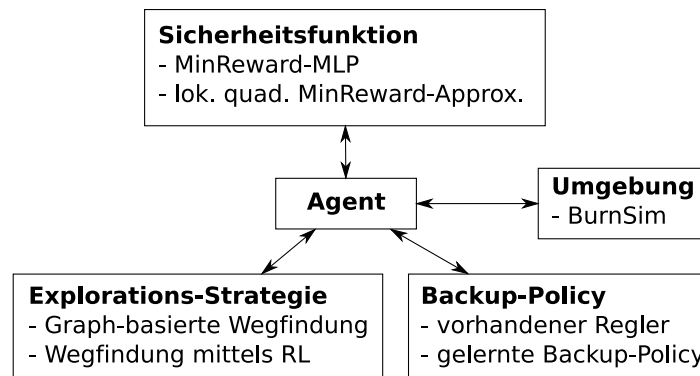


Abb. 2: Komponenten der Implementierung mit den jeweiligen Realisierungen.

mit noch zu explorierenden Aktionen, das aktuelle Level wird auf 0 gesetzt.

2. **Initiales Folgen der Backup-Policy.** Der Backup-Policy wird so lange gefolgt, bis diese einen stationären Punkt oder einem bestimmten Bereich des Zustandsraums erreicht und nicht mehr verlässt. Alle dabei besuchten Zustände erhalten Level 0 und werden in die Liste der Zustände mit noch zu explorierenden Aktionen eingetragen, sofern die Backup-Policy nicht bereits alle möglichen Aktionen in einem Zustand ausgeführt hat.
3. **Explorationsschritt.** Die Explorations-Strategie wählt eine zu explorierende Aktion aus. Falls im aktuellen Zustand keine Aktion zu explorieren ist (entweder, weil bereits alle exploriert wurden, oder das Level des aktuellen Zustands nicht dem aktuell zu explorierenden Level entspricht), generiert sie die nötige(n) Aktion(en), durch die der Agent in einen Zustand mit aktuell zu explorierenden Aktionen gelangt, um anschließend die dort zu explorierende Aktion zu wählen. Vor dem Ausführen der Aktion prüft der Agent mithilfe der Sicherheitsfunktion ihre Sicherheit. Wird die Aktion als sicher eingeschätzt, wird sie ausgeführt. Andernfalls wird sie als unsicher markiert und die Backup-Policy genutzt, um in den sicheren Bereich zurückzukehren. Für den Fall, dass beim Ausführen des Explorationsschritts ein neuer, bisher unbekannter Zustand oder ein bekannter Zustand mit höherem Level als dem aktuell zu explorierenden erreicht wurde, wird ebenfalls die Backup-Policy genutzt.
4. **Rückkehr mittels Backup-Policy.** Wurde ein neuer Zustand oder ein Zustand mit höherem als dem aktuell zu explorierenden Level betreten, wird die Backup-Policy genutzt, um in den bekannten Bereich zurückzukehren. Im einfachsten Fall wird dabei die Backup-Policy so lange genutzt, bis ein Zustand mit Level 0 erreicht wurde.
5. **Level-Erhöhung.** Wenn alle sicheren Aktionen in allen Zuständen des aktuellen Levels exploriert wurden, wird das Level erhöht und die Exploration mit Schritt 3 fortgesetzt.
6. **Explorations-Ende.** Die Exploration ist beendet, wenn sämtliche sicheren Aktionen in allen Zuständen ausgeführt wurden oder wenn ein Abbruch-Kriterium erfüllt ist. Dies kann das Erreichen eines Maximal-Levels sein. Ebenso ist es möglich, auf Basis der bereits gesammelten Beobachtungen eine optimale Policy zu bestimmen und deren Performance zu ermitteln. Wenn die Policy die gewünschte Qualität erreicht hat oder sich mit weiteren Beobachtungen nicht mehr wesentlich ändert, kann die Exploration beendet werden.

4.3 Bestimmung der Sicherheitsfunktion

Für die Realisierung der Sicherheitsfunktion wird die Min-Reward-Funktion lokal-quadratisch approximiert. Aus diesem Grund wurde die Schätzung der Min-Reward-Funktion linear im Feature-Raum

$$\Phi_i = (f_i, h_i, f_i^2, h_i^2, f_i h_i, 1)$$

mit f_i als Fuel und h_i als Brummen der jeweiligen Transitionen i , realisiert. Die dabei entstehende Feature-Matrix

$$\mathbf{X} = \begin{pmatrix} f_1 & h_1 & f_1^2 & h_1^2 & f_1 h_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f_k & h_k & f_k^2 & h_k^2 & f_k h_k & 1 \end{pmatrix}$$

wird schließlich mit der Methode der kleinsten Quadrate $\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$ an die beobachteten Min-Rewards

$$\mathbf{y} = \begin{pmatrix} r_{\min,1} \\ \vdots \\ r_{\min,k} \end{pmatrix}$$

approximiert. Standardmäßig wurden die zehn nächsten Nachbarn herangezogen. Wird der Feature-Raum nicht vollständig aufgespannt, ist $\mathbf{X}^T \mathbf{X}$ also nicht regulär, werden weitere Nachbarn hinzugezogen. Die Singularität von \mathbf{X} wurde bei Überschreiten von $\text{cond}(\mathbf{X}^T \mathbf{X}) > 1000$ angenommen.

4.4 Backup-Policy

Ein für die *BurnSim* im Vorhinein bekannter Standard-Regler, der als Rückkehrstrategie genutzt werden kann, ist

$$\pi^{\text{backup}}(\mathbf{s} = (f, h)) = \begin{cases} \text{increase,} & \text{falls } f < 0.425 \\ \text{decrease,} & \text{falls } f > 0.425 \\ \text{keep,} & \text{sonst} \end{cases} \quad (3)$$

Dieser strebt einen Zustand unterhalb des stationären Bereichs bezüglich Fuel an und führt damit zu einer permanenten Reduktion des Brummens.

Falls eine Backup-Policy im Vorhinein nicht bekannt ist, muss diese unter Verwendung des in Abschn. 3.2 vorgestellten RL-Verfahrens, das anstelle des Erwartungswerts der Summe das Minimum maximiert, aus bereits vorhandenen Beobachtungen gelernt werden. Inspiriert von der Idee des Apprenticeship Learning [10] wurden die Beobachtungen aus einer Trajektorie gewonnen, die ein Lehrer vorgegeben hat (Abschn. 2). Abb. 3 zeigt die Trajektorie sowie die resultierende Policy. Die Trajektorie umfasst alle für eine sichere Backup-Policy nötigen Beobachtungen. Insbesondere sind Beobachtungen im Bereich rechts von $f = 0.5$ wichtig, damit gelernt werden kann, dass dort Fuel-Verringerung (decrease) die sichere Aktion ist.

Da nur eine überschaubare Anzahl von Zuständen in der Lehrer-Trajektorie vorkommen, war es möglich, diese Zustände als diskret aufzufassen und tabellenbasiert eine Q-Funktion unter Anwendung der Variation der Bellman-Optimalitätsgleichung aus Abschn. 3.2 zu bestimmen. Nicht determinierte Q-Werte, deren Transitionen vom Lehrer nicht durchgeführt wurden, werden darüberhinaus im Nachhinein auf $-\infty$ gesetzt, so dass auch eine auf der resultierenden Q-Funktion basierende

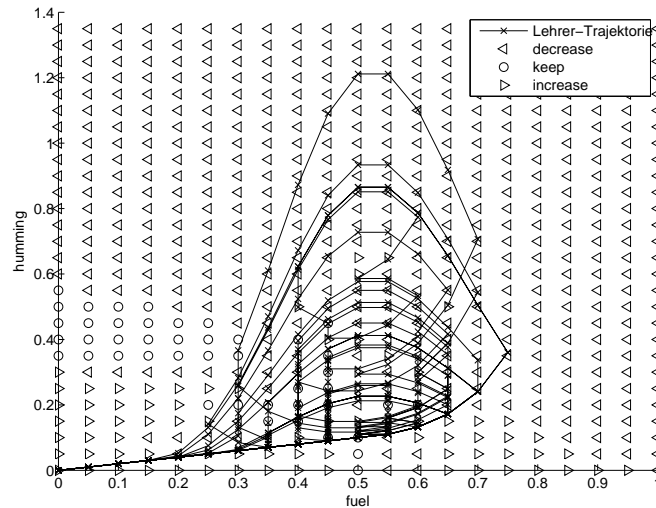


Abb. 3: Lehrer-Trajektorie und gelernte Backup-Policy.

Policy diese Aktion niemals ausführen wird. Dies ist erforderlich, um Sicherheit garantieren zu können.

Die Q-Funktion enthält nur für einige Zustände des Zustandsraums Einträge. Eine Backup-Policy muss jedoch für alle Zustände des gesamten Zustandsraums eine Aktion liefern können. Dies wurde durch eine einfache Nächster-Nachbar-Generalisierung [11] erreicht, für die zu einem gegebenen Zustand s der Zustand \hat{s} in Q gesucht wird, dessen euklidischer Abstand zu s minimal ist.

Neben der verwendeten Lehrer-Trajektorie zeigt Abb. 3 die resultierende Policy. Lediglich für die Punkte der Trajektorie enthält die Q-Funktion Einträge, für alle anderen Punkte wird die Aktion anhand der Q-Werte des nächsten Nachbarn gewählt. Man erkennt, dass in fast allen Fällen die Aktion gewählt wird, die Fuel in Richtung 0.5 verändert. Wenn die Lehrer-Trajektorie für einen Zustand nur eine Aktion enthält, so wählt auch die gelernte Policy für diesen Zustand und seine Nachbarn diese Aktion, selbst wenn sie nicht optimal im Sinne Maximierung des minimalen Rewards ist. Dieser Effekt ist insbesondere im Bereich von $f \leq 0.3$ und $h \approx 0.4$ (Aktion *keep*) sowie im Bereich von $0.65 \leq f \leq 0.85$ und $h \leq 0.1$ (Aktion *increase*) zu beobachten. In beiden Fällen ist das Fehlverhalten jedoch unkritisch: Das Ausführen von *keep* in Bereichen mit $f < 0.5$ führt zur Verringerung des Humming; das unerwünschte *increase* im anderen genannten Bereich ist unproblematisch, weil dieser Bereich des Zustandsraums nie betreten werden kann.

4.5 Explorations-Strategie und Zustands-Repräsentation

Zusammen mit der Vorstellung der level-basierten Explorations-Strategie in Abschn. 3.3 wurden zwei Möglichkeiten zur Realisierung der für diese Explorations-Strategie notwendigen Wegfindung aufgezeigt – durch Breitensuche in einem Graphen sowie durch Definition und Lösung eines Wegfindungs-RL-Problems.

Für den graph-basierten Ansatz (Abschn. 3.3.1) wurde der Zustandsraum künstlich verkleinert, indem in der BurnSim-Implementierung nach jedem Schritt das Humming auf die Genauigkeit von 1/50 gerundet wurde. Es wurde also das RL-Problem selbst verändert. Bei einem Real-Welt-Problem ist die Verkleinerung des Zustandsraums in dieser Form natürlich nicht möglich.

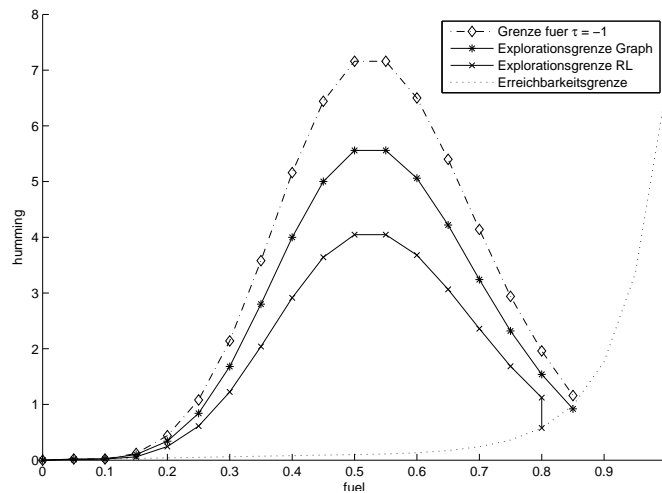


Abb. 4: Explorierter Bereich der graph-basierten Exploration sowie der Exploration auf aggregierten Zuständen mit RL-Wegfindung. Zusätzlich ist die Grenze des maximal explorierbaren Bereichs für $\tau = -1$ eingezeichnet. Zustände zwischen einer Kurve und der Erreichbarkeitsgrenze (punktierte Kurve) wurden von der Exploration besucht bzw. sind sicher erreichbar für $\tau = -1$.

Alternativ wurde der Wegfindungsansatz mittels RL verfolgt (Abschn. 3.3.2), bei dem Zustände zu Clustern aggregiert werden. Die Aggregation wurde dabei wie folgt vorgenommen:

- Da Fuel nur um 0.05 erhöht und verringert werden kann und immer im Intervall $[0, 1]$ gehalten wird, kann Fuel nur 20 diskrete Werte annehmen.
- Brummen hingegen nimmt kontinuierliche Werte an. Zur Aggregation von Zuständen wurde Brummen auf eine Genauigkeit von $1/50$ gerundet. Dabei handelt es sich nicht um eine Änderung des RL-Problems, *BurnSim* selbst blieb unverändert. Beispielsweise sind die Zustände $s_1 = (0.25; 0.1239\dots)$ und $s_2 = (0.25; 0.1277\dots)$ zwei unterschiedliche Zustände in *BurnSim*; beide gehören jedoch zum aggregierten Zustand $\hat{s}_1 = (0.25; 0.12)$.

Die Exploration wurde auf den so aggregierten Zuständen nach dem gleichen Prinzip wie die graph-basierte Variante durchgeführt.

4.6 Ergebnisse

Abbildung 4 zeigt, in welchen Bereichen des Zustandsraums eine graph-basierte Exploration sowie eine Exploration mit Zustands-Aggregation und RL-Wegfindung Beobachtungen sammeln konnte. Zusätzlich sind die Grenzen des maximal (sicher) explorierbaren Bereichs für $\tau = -1$ eingezeichnet. Der eigentliche Grenzwert der *BurnSim* lautet $\tau = -1$, somit sind sämtliche Zustände unterhalb der eingezeichneten Grenze sicher.

Graph-basierte Exploration Für dieses Experiment wurde die lokal-quadratische Approximation (Abschn. 4.3) verwendet, als Backup-Policy kam der Standard-Regler (Gl. 3) zum Einsatz. Da während der Exploration die Grenze für $\tau = -1$ nicht unterschritten wurde, hat die Exploration das Kriterium der sicheren Exploration erfüllt.

Exploration mit Zustands-Aggregation und RL-Wegfindung Wie bei der graph-basierten Exploration wurde die lokal-quadratische Approximation als Sicherheitsfunktion verwendet. Bei der Backup-Policy handelte es sich um eine wie in Abschnitt 4.4 beschriebene gelernte Policy. Der bei diesem Experiment explorierte Bereich ist kleiner als der von der graph-basierten Exploration abgedeckte.

In manchen Bereichen des Zustandsraums enthält die zum Lernen verwendete Lehrer-Trajektorie keine alternativen Aktionen, sondern nur eine nicht optimale. Mangels Alternativen in den Trainingsdaten wählt die resultierende Policy auch diese Aktion. Das Ergebnis ist eine Backup-Policy, die das Brummen langsamer senkt als der Standard-Regler. Dadurch werden auf den Backup-Trajektorien dieser Policy geringere Rewards beobachtet als auf denen des Standard-Reglers. Als Folge stuft die Sicherheitsfunktion früher Aktionen als unsicher ein.

Trotz des gegenüber der graph-basierten Exploration kleineren abgedeckten Bereichs, wurden Informationen aus dem gesamten Bereich des Zustandsraum gewonnen, der von der Trajektorie der optimalen Policy betreten wird. Somit sind die Beobachtungen in diesem Fall ausreichend, um daraus die optimale Policy abzuleiten.

Die graph-basierte Exploration benötigte ca. 52000 Schritte, bis keine weitere Exploration mehr möglich war. Im Falle der Exploration auf aggregierten Zuständen mit RL-Wegfindung wurden dagegen ca. 98000 Schritte benötigt. Die größere Zahl an Explorationsschritten begründet sich durch die für die Exploration weniger geeignete gelernte Backup-Policy sowie der Unsicherheit in der Wegfindung aufgrund der Verletzung der Markov-Bedingung durch die Zustands-Aggregation.

Erreichte Performance Für den Nachweis, dass die Datenbasis bei Anwendung der beiden Explorationsstrategien geeignet ist, um optimale sichere Policies zu bestimmen, wurden verschiedene RL-Verfahren eingesetzt. Wir haben Neural Fitted Q-Iteration (NFQ) [12], Dynamische Programmierung mit Nächster-Nachbar-Generalisierung und die optimale Policy verglichen. Die erreichten Durchschnitts-Rewards sind in Tabelle 4.6 zusammengefasst. Beide Verfahren konnten gute Policies bestimmen. Darüber hinaus war NFQ in der Lage, mit den Beobachtungen beider Explorationen die optimale Policy zu ermitteln.

RL-Verfahren	Avg. Reward	
	Graph-basiert	RL-Wegfindung
Dynamische Programmierung	1.164	1.132
Neural Fitted Q-Iteration	1.166	1.166
Optimale Policy	1.166	

Tab. 2: Performance

5 Ausblick

In dieser Arbeit haben wir das Problem der sicheren Exploration beschrieben und Lösungsmöglichkeiten aufgezeigt. Diese stellen einen Ausgangspunkt für die Beherrschbarkeit der Exploration in industriellen Anwendungen dar. Weitere Fragestellungen umfassen insbesondere die Untersuchung des mit Zustands-Aggregation arbeitenden Verfahrens für stochastische MDPs, die Behandlung von hochdimensionalen Zustandsräumen und die Fehlerabschätzung der Sicherheitsfunktion. Darüber hinaus ist der Einsatz an einem realen System angestrebt.

Literatur

- [1] Sutton, R. S.; Barto, A. G.: *Reinforcement Learning: An Introduction*. MIT Press. 1998.
- [2] Bellman, R.: *Dynamic Programming and Stochastic Control Processes*. Rand Corporation. 1957.
- [3] Schaefer, A. M.; Schneegass, D.; Sterzing, V.; Udluft, S.: Controlling a gas turbine with a Recurrent Control Neural Network. In: *Proc. of the International Joint Conference on Neural Networks*. Orlando: MIT Press. 2007.
- [4] Dearden, R.; Friedman, N.; Russell, S. J.: Bayesian Q-Learning. In: *Proc. of AAAI/IAAI*, S. 761–768. 1998.
- [5] Poupart, P.; Vlassis, N.; Hoey, J.; Regan, K.: An analytic solution to discrete Bayesian reinforcement learning. In: *Proc. of the International Conference on Machine Learning*, S. 697 – 704. 2006.
- [6] Heger, M.: Consideration of risk in reinforcement learning. In: *Proc. of the International Conference on Machine Learning*, S. 105–111. Morgan Kaufmann. 1994.
- [7] Watkins, C.: *Learning from Delayed Rewards*. Dissertation, University of Cambridge. 1989.
- [8] Neuneier, R.; Mihatsch, O.: Risk Sensitive Reinforcement Learning. In: *Proc. of NIPS*, S. 1031–1037. 1998.
- [9] Geibel, P.: Reinforcement Learning with Bounded Risk. In: *Proc. of the International Conference on Machine Learning*, S. 162–169. Morgan Kaufmann, San Francisco, CA. 2001.
- [10] Abbeel, P.; Ng, A. Y.: Exploration and Apprenticeship Learning in Reinforcement Learning. In: *Proc. of the International Conference on Machine Learning*, S. 1–8. 2005.
- [11] Hastie, T.; Tibshirani, R.; Friedman, J.: *The Elements Of Statistical Learning Theory, Data Mining, Inference, and Prediction*. New York: Springer. 2001.
- [12] Riedmiller, M.: Neural Fitted Q-Iteration - First Experiences with a Data Efficient Neural Reinforcement Learning Method. In: *Proceedings of the 16th European Conference on Machine Learning*, S. 317–328. 2005.

Datenanalyse und Prozessoptimierung für Kanalnetze und Kläranlagen mit CI-Methoden

**T. Bartz-Beielstein, M. Bongards,
C. Claes, W. Konen, H. Westenberger**

Fakultät für Informatik und Ingenieurwissenschaften
Fachhochschule Köln
Tel.: (02261) 8196-0
Fax: (02261) 8196-15
E-Mail: thomas.bartz-beielstein@fh-koeln.de

Zusammenfassung

Prozesse optimal zu steuern und Prognosen über ihren zukünftigen Verlauf anzustellen, gehört zu den wichtigsten, aber auch schwierigsten Aufgaben der industriellen Praxis. Wir demonstrieren, wie aktuelle Methoden der Computational Intelligence (CI) und des Data Mining ebenso wie klassische Ansätze zur Modellierung, Simulation und Optimierung von Kanalnetzen und Kläranlagen eingesetzt werden können. Dabei zeigt sich, dass die Kombination von Standardverfahren aus der Zeitreihenanalyse mit dem Verfahren der sequentiellen Parameteroptimierung schnell zu problemspezifischen Vorhersagemodellen führen kann.

1 Modellierung von Füllständen in Regenüberlaufbecken

Böden und Kanalnetze stellen ein komplexes dynamisches System dar. Offensichtlich haben die aktuellen Zustände der Böden einen wichtigen Einfluss auf den Füllstand der Überlaufbecken, da z.B. trockene Böden ein anderes Abflussverhalten zeigen als feuchte oder gar gefrorene Böden. Hinzu kommen noch weitere Einflussfaktoren wie die Sonnenscheindauer, die landwirtschaftliche Nutzung, Temperatur usw. Im Forschungsprojekt KANNST (KANalNetz-Steuerung) wird die Modellierung und Prognose von Füllstandshöhen in Regenüberlaufbecken auf Basis einzelner Regenmessungen untersucht [1]. Die Modellierung beruht auf empirisch erhobenen Regendaten, die vom Aggerverband zur Verfügung gestellt wurden. Der Aggerverband ist ein Wasserverband in Nordrhein-Westfalen. In seinem Verbandsgebiet übernimmt er für seine Mitglieder (Kreise, Städte, Gemeinden, Industrie und öffentliche Wasserversorgungsunternehmen) alle Aufgaben der Wasserwirtschaft. Die Regendaten dienen als Eingabedaten für ein Kanalnetzsimulationsprogramm. Wir wählten hierzu das Storm Water Management Model (SWMM) [2]. SWMM wird bereits seit 1971 weltweit zur Simulation von Kanalnetzen eingesetzt.

Die Regendaten wurden minütlich über einen Zeitraum von 108 Tagen erhoben, so dass insgesamt $n = 155.521$ Datensätze zur Verfügung standen. Untersuchungen mit anderen Datensätzen wurden zusätzlich durchgeführt, da von Anwenderseite ein großes Interesse an diesen Prognosen besteht. Basierend auf den gemessenen Regendaten x_t ($t = 1, 2, \dots, n$), die im Folgenden als „Rain data“ bezeichnet werden, sollen die simulierten Füllstandshöhen in den Becken möglichst genau vorhergesagt werden. Die simulierten Werte y_t werden als „Outflow (target)“ und die von unseren Modellen vorhergesagten Werte \hat{y}_t werden als „Outflow (prediction)“ bezeichnet. Bekannt sind somit x_t und y_t , zu bestimmen ist ein

Modell zur Berechnung von \hat{y}_t , so dass der Fehler zwischen y_t und \hat{y}_t möglichst gering ist.

Unterschiedliche Ansätze wie *neuronalen Netze* (NN), *Echo State Networks* (ESN), Differentialgleichungen oder die Modellierung mit Integralgleichungen werden dabei von uns eingesetzt [3].

Im zweiten Abschnitt stellen wir das Vorhersagemodell vor und beschreiben die zugehörigen Schritte zur Datenvor- und nachbearbeitung. Ebenfalls in diesem Abschnitt wird dargelegt, wie die Parameter des von uns favorisierten Modells eingestellt wurden. Im letzten Abschnitt geben wir eine kurze Zusammenfassung.

2 Verfahren

2.1 Das Vorhersagemodell

Erste Ansätze mit NN brachten nicht die gewünschten Ergebnisse. Der Einsatz von FIR-Filtern (Filter mit endlicher Impulsantwort, engl.: „finite impulse response filter“) führte mit geringem Modellierungsaufwand zu guten Ergebnissen. FIR-Filter verfügen über eine Impulsantwort mit garantiert endlicher Länge. Daher können FIR-Filter, unabhängig davon, wie die Filterparameter gewählt werden, niemals instabil werden oder zu einer selbstständigen Schwingung angeregt werden.

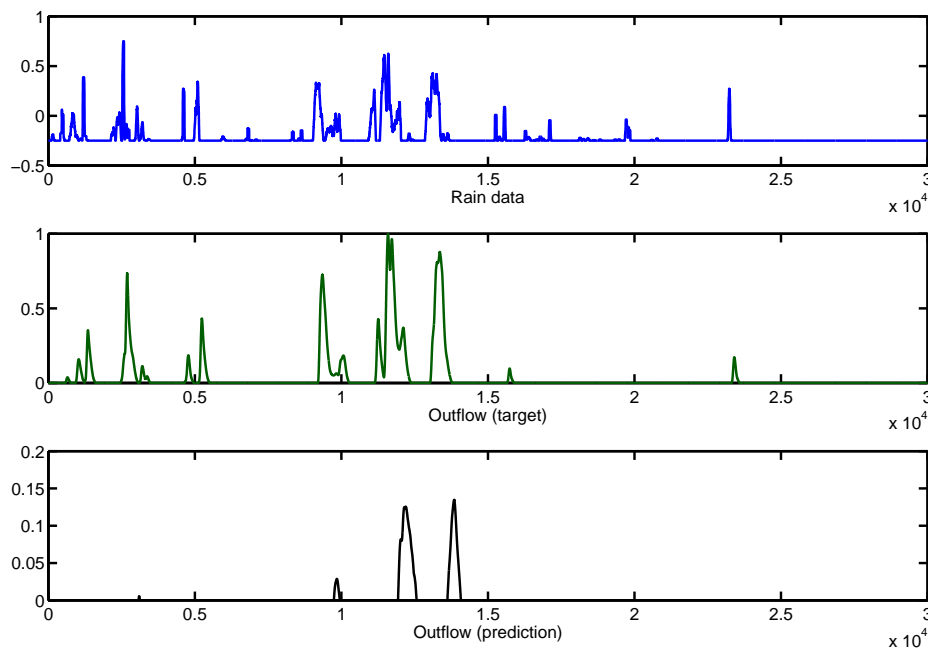


Bild 1: Vorhersage mit „intuitiv“ gewählten Modellparametern. Oben: Gemessener Niederschlag x_t . Mitte: Simulierter Abfluss y_t . Unten: Durch das FIR-Filter Modell vorhergesagter Abfluss \hat{y}_t .

In unserem Fall war die einfache Anwendung eines FIR-Filters nicht ausreichend. Eine passende Datenvor- und Nachbearbeitung war, ebenso wie eine Einstellung der Filterparameter, erforderlich. Die Regendaten wurden in vier Schritten verarbeitet: Normierung, Translation, Glättung und Skalierung.

Normierung: Im ersten Schritt wurden die Regendaten x_t auf das Intervall $[0; 1]$ abgebildet.

Translation: Im zweiten Schritt wurde die Verdunstung des Niederschlags modelliert. Von den gemessenen und normierten Werten wurde ein konstanter Betrag s abgezogen.

Exponentielle Glättung: Im dritten Schritt wurden l Gewichte c_i berechnet, die nach einer gewissen Verzögerung d (*delay*) mit wachsendem i exponentiell abnahmen. Hierzu war ein Abklingfaktor a zu bestimmen. Mit den so berechneten Gewichten wurden mit dem FIR-Filter die normierten und translatierten Regendaten geglättet.

Skalierung: Die geglätteten Werte wurden mit dem Faktor s_2 skaliert und negative Vorhersagewerte wurden auf Null gesetzt.

Die ersten Modellierungsversuche mit den so angepassten FIR-Filtern verliefen vielversprechend, so dass dieser Ansatz weiter verfolgt wurde. Abbildung 1 zeigt, dass einzelne Simulationswerte tendenziell erfasst werden (z.B. im Bereich von 70.000 bis 150.000), andere Bereiche jedoch überhaupt nicht.

Für das aus der exponentiellen Glättung und der Datenvorverarbeitung bestehende Vorhersagemodell sind fünf Parameter zu bestimmen. In vielen Situationen wird an dieser Stelle versucht, manuell, d.h. durch Probieren, günstige Parametereinstellungen für das Modell zu finden. Diese Vorgehensweise hat in vielerlei Hinsicht große Nachteile im Vergleich zu einer systematischen Vorgehensweise. Im Bereich der Simulation wurde dies von Kleijnen ausführlich dargestellt [4]. Wir setzten die sog. *sequentielle Parameteroptimierung* (SPO) ein [5].

2.2 Sequentielle Parameteroptimierung

SPO wurde speziell für Optimierungsprobleme entwickelt und kombiniert Ansätze aus der klassischen Regression und Varianzanalyse mit modernen statistischen Verfahren. [6] geben eine kurze Einführung in die sequentielle Parameteroptimierung. SPO verwendet eine sequentielle Vorgehensweise zur Bestimmung guter Parametereinstellungen. Zudem werden statistische Daten generiert, die während oder nach Abschluss der Optimierung wertvolle Informationen über das FIR-Filter-Modell liefern. Interessant ist z.B. die Analyse, welcher der fünf Faktoren den größten Einfluss auf die Vorhersagegüte hat.

2.2.1 Region of Interest

SPO erfordert die Spezifikation des Suchraums, der sog. *region of interest*, für mögliche Parametereinstellungen. Erfahrungswerte für gute Einstellungen lagen in unserem Fall nicht vor, da dieses Modell zum ersten Mal für diese Daten herangezogen wurde. Es gab aber Bereiche, die für das Modell als „sinnvoll“ zu bezeichnen waren. So waren zu große Werte für die Verzögerung (d) nicht sinnvoll, wohingegen die Auswirkung kleiner Werte noch zu untersuchen war. Es war zu Beginn der Untersuchungen nicht klar, ob die Verzögerung überhaupt einen positiven Effekt auf die Modellierungsgüte hat. Für die Verzögerung wählten wir daher den Bereich von 0 bis 1000 als *region of interest*. Die Translation s wurde im Bereich von 0 bis 1 variiert. Für alle anderen Parameter wurde der Bereich von 1 bis 1000 gewählt. Diese Einstellungen sind in Tabelle 1 zusammengestellt.

Tabelle 1: Region of interest für die Parameter des Vorhersagemodells.

Parameter	Abkürzung	Symbol	untere Grenze	obere Grenze
Verdunstung	SHIFT	s	0	1
Anz. Gewichte	LENGTH	l	1	1000
Abklingfaktor	EXP	a	1	1000
Verzögerung	DELAY	d	1	1000
Skalierung	SCALE2	s_2	1	1000

2.2.2 Gütefunktion

Zur Güte der berechneten Vorhersage wurde die Summe der quadratischen Abstände $\sum_1^n (y_t - \hat{y}_t)^2$ zwischen simulierten und vorhergesagten Werten berechnet, so dass ein Minimierungsproblem vorlag.

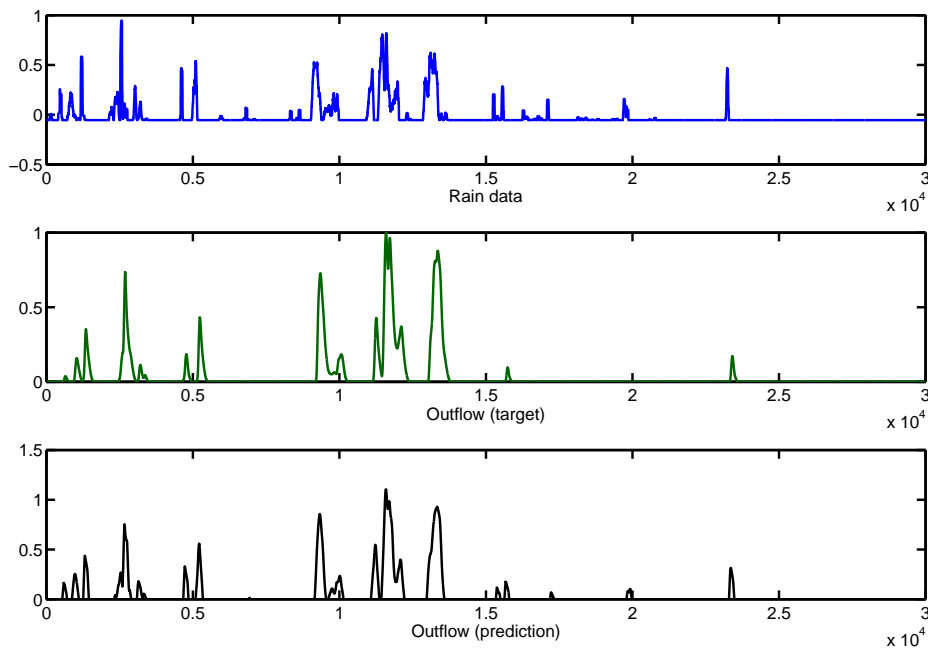


Bild 2: Vorhersage mit dem durch SPO verbesserten Modell. Es wurden 50 SPO Schritte durchgeführt. Die gesamte Optimierung benötigt auf einem Standard-PC (2GHz, 2GB RAM) nur wenige Minuten.

2.2.3 Ergebnis der Optimierung mit SPO

Mit der sequentiellen Parameteroptimierung konnte der Fehler auf 0,000248612 reduziert werden. Dieses Ergebnis wurde mit den Einstellungen $s = 0,055576$, $l = 154$, $a = 212,12$, $d = 78$ und $b = 61,62$ erzielt. Die auf diesen Einstellungen basierende Vorhersage ist in Abbildung 2 dargestellt.

Während eines SPO Laufs kann der Einfluss und die Veränderung der einzelnen Parameter im FIR-Filter-Modell visualisiert werden. In Abbildung 3 werden diese Informationen

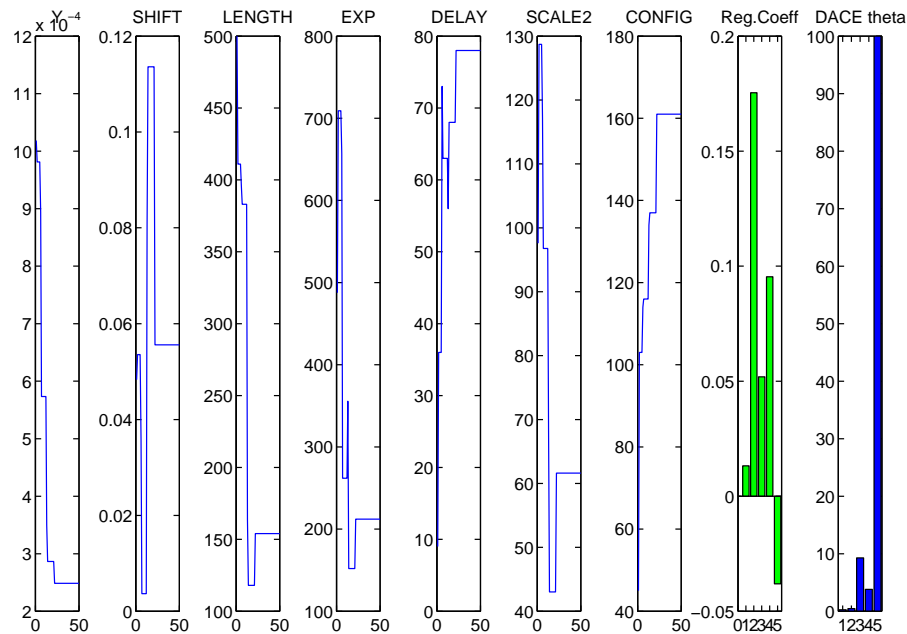


Bild 3: Sequentielle Parameter Optimierung. Während eines Laufs können die folgenden Informationen grafisch dargestellt werden (Abbildungen von links nach rechts): 1. Gütefunktion: Wie verändert sich die Vorhersagegüte? 2.-6. Parametereinstellungen für Verdunstung (SHIFT), Anz. Gewichte (LENGTH), Abklingfaktor (EXP), Verzögerung (DELAY) und Skalierung (SCALE2). 7. Nummer der besten Konfiguration (CONFIG). Die Parametereinstellungen werden durchnummeriert. Neue Einstellungen erhalten eine höhere Konfigurationsnummer. Steigen die CONFIG-Werte an, so werden neue Einstellungen erzeugt, die besser sind als die bisher gefundenen. 8. Koeffizienten eines linearen Regressionsmodells (üblicherweise β_i). Diese können als Indikatoren für die Wichtigkeit einzelner Faktoren herangezogen werden. 9. Koeffizienten eines stochastischen Prozessmodells (üblicherweise θ_i). Diese können ebenfalls als Indikatoren für die Wichtigkeit einzelner Faktoren herangezogen werden.

exemplarisch dargestellt. Das erste Schaubild zeigt die Veränderung der Gütefunktion, die nächsten Abbildungen zeigen die Veränderungen der fünf Parameter (SHIFT, LENGTH, EXP, DELAY und SCALE) während 50 Schritten der SPO. Da SPO sequentiell vorgeht und in jedem Schritt neue Parametereinstellungen generiert werden, kann das „Innovationspotential“ der SPO im siebten Schaubild (CONFIG) abgelesen werden. Die beiden letzten Abbildungen zeigen den Effekt der fünf Faktoren, einerseits gemessen als Regressionskoeffizienten eines klassischen linearen Regressionsmodells (Reg.Coeff) und andererseits als Korrelationskoeffizienten eines stochastischen Prozessmodells (DACE theta) [7]. Der Einfluss einzelner Faktoren kann auch durch einen Regressionsbaum dargestellt werden [8]. Aus Abbildung 4 wird ersichtlich, dass der Faktor SCALE2, also die abschließende Skalierung, den größten Einfluss auf die Vorhersagegüte besitzt.

Dies stimmt überein mit den Werten der Korrelationskoeffizienten des stochastischen Prozessmodells. Die Koeffizienten des linearen Regressionsmodells legen den Schluss nahe, dass die Anzahl der Gewichte (LENGTH) den größten Einfluss besitzt. Dies ist nicht sonderlich überraschend. Aus anderen Studien ist uns bekannt, dass der Effekt der einzelnen Faktoren von dem zugrundeliegenden Modell abhängt und dass insbesondere Unterschiede zwischen linearen Regressionsmodellen und stochastischen Prozessmodellen

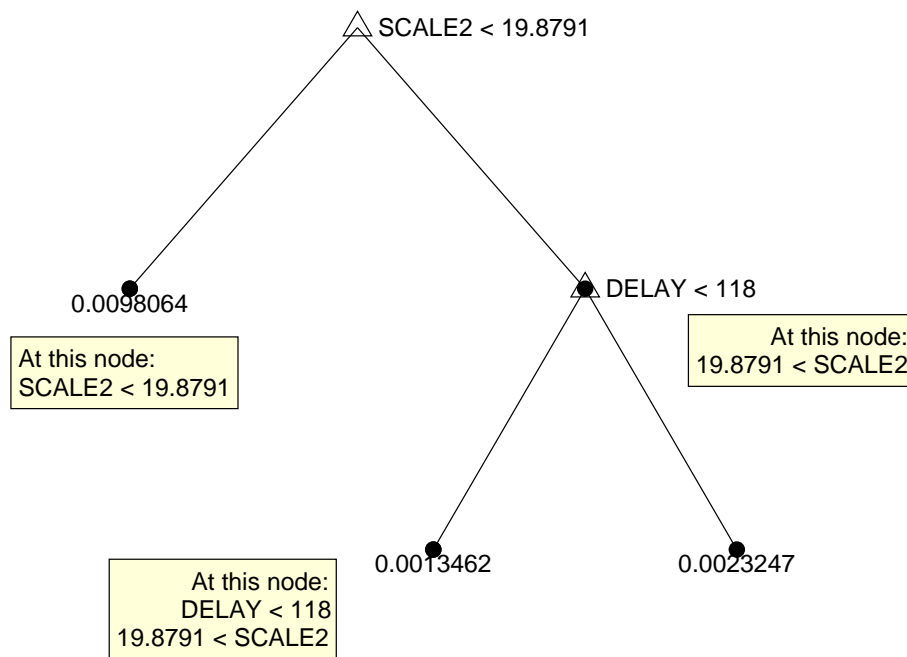


Bild 4: Regressionbaum zur Darstellung der Effekte der Filterparameter auf die Vorhersagegüte. Die besten Vorhersagen werden erzielt, wenn die Verzögerung (DELAY) kleiner als 118 und der Faktor für die Skalierung (SCALE2) größer als 19 gewählt werden.

auftreten.

3 Zusammenfassung

Als erstes Ergebnis erhalten wir schon eine relativ gute Prognose der mit SWMM simulierten Füllstände (Abbildung 2). Da die eingesetzten Verfahren - in diesem Fall neuronale Netze und FIR-Filter - mehrere Parametern benötigten, setzten wir Methoden der statistischen Versuchsplanung zu deren Bestimmung ein. Bei den NN konnten auch durch den Einsatz von SPO keine geeigneten Vorhersagemodelle generiert werden. Bei den FIR-Filter Modellen führte SPO schnell zu guten Parametereinstellungen und zu einem an das Problem angepassten Vorhersagemodell.

Der Anwender erhält somit statistisch abgesicherte Informationen, dass das empfohlene Prognosemodell besser geeignet ist als die anderen betrachteten Modelle und zusätzlich die Sicherheit, dass diese Aussagen nicht von zufälligen Messdaten abhängig sind, sondern dass diese Ergebnisse für ein großes Spektrum an unterschiedlichen problemspezifischen Parametern (wie z.B. die Bodenbeschaffenheit, Niederschlagsmengen) Gültigkeit behält.

Zukünftige Arbeiten untersuchen die Einsatzmöglichkeiten dieses Ansatzes für Daten, die nicht durch Simulationsmodelle wie SWMM erzeugt worden sind, sondern für empirisch gewonnene Messwerte. Hier treten durch Messungenauigkeiten und Messfehler weitere Schwierigkeiten auf, die momentan noch durch kein Prognosemodell zufriedenstellend gelöst werden können.

Danksagung

Für viele interessante Diskussionen und die Bereitstellung anwendungsbezogener Daten danken wir Dipl.-Ing. Tanja Hilmer und Dipl.-Ing. Andreas Stockmann.

Literatur

- [1] Bongards, M.: Online-Konzentrationsmessung in Kanalnetzen – Technik und Betriebsergebnisse. Techn. Ber., Cologne University of Applied Sciences. 2007.
- [2] U.S. Environmental Protection Agency: Storm Water Management Model. <http://www.epa.gov/ednrmrl/models/swmm/index.htm>, Online; Stand 26.08.07. 2007.
- [3] Jaeger, H.; Haas, H.: Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. *Science* (2004), S. 78–80.
- [4] Kleijnen, J. P. C.: *Statistical Tools for Simulation Practitioners*. New York NY: Marcel Dekker. 1987.
- [5] Bartz-Beielstein, T.: *Experimental Research in Evolutionary Computation—The New Experimentalism*. Berlin, Heidelberg, New York: Springer. 2006.
- [6] Bartz-Beielstein, T.; Preuss, M.: Moderne Methoden zur experimentellen Analyse evolutionärer Verfahren. In: *Proc. 16th Workshop Computational Intelligence* (Mikut, R.; Reischl, M., Hg.), S. 25–32. Universitätsverlag, Karlsruhe. 2006.
- [7] Santner, T. J.; Williams, B. J.; Notz, W. I.: *The Design and Analysis of Computer Experiments*. Berlin, Heidelberg, New York: Springer. 2003.
- [8] Breiman, L.; Friedman, J. H.; Olshen, R. A.; Stone, C. J.: *Classification and Regression Trees*. Monterey CA: Wadsworth. 1984.

Learning Binary Classifiers for Applications in Safety-Related Domains

Sebastian Nusser^{1,2}, Clemens Otte¹, and Werner Hauptmann¹

¹ Siemens AG, Corporate Technology, Otto-Hahn-Ring 6, 81730 Munich

² School of Computer Science, Otto-von-Guericke-University, Magdeburg

E-Mail: {sebastian.nusser.ext, clemens.otte, werner.hauptmann}@siemens.com

Abstract

This paper introduces two binary classification approaches to cope with the specific requirements of safety-related application problems. Both approaches are based on ensembles of local models, each utilizing only a small subspace of the complete input space. Thus, the interpretability and verifiability of the local models is ensured, which is a crucial prerequisite for applications in safety-related domains. Similar to the Boosting approaches, the ensemble of local models enhances the predictive performance of the final global model – compared to the limited predictive performance of the local models.

1 Introduction

Safety-related systems are systems whose malfunction or failure may lead to death or serious injury of people, loss or severe damage of equipment, or environmental harm. They are deployed, for instance, in aviation, automotive industry, medical systems and process control. In this paper, two machine learning methods are introduced for use in safety-related problems, an application domain where a wrong decision cannot be rectified. An overview of alternative approaches for handling safety-related problems with machine learning methods can be found in [1].

In practical application tasks, the available training data is often too sparse and the number of input dimensions is too large to sufficiently apply statistical risk estimation methods. In most cases, high-dimensional models are needed to solve a given problem. Unfortunately, such high-dimensional models are hard to verify (*curse of dimensionality*), may tend to overfitting, and the interpolation and extrapolation behavior is often unclear or intransparent. An example of such counterintuitive behavior is illustrated in Fig. 1, where the prediction of the model changes in a region not covered by the given data set. Such behavior becomes even more likely and much more difficult to discover in the high-dimensional case. Thus, a model building method is required, which provides a well-defined interpolation and extrapolation behavior¹.

¹“Well-defined” states here that the decision bounds of the learned models can exactly be determined for every point of the input space.

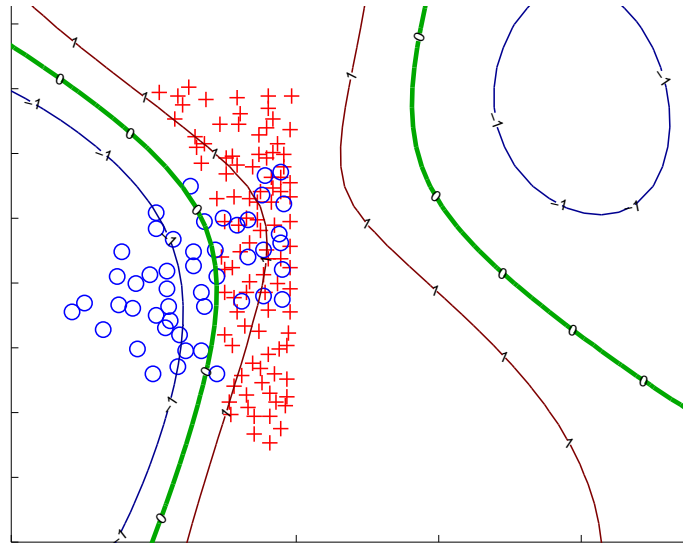


Figure 1: Counterintuitive extrapolation behavior in a region not covered by the data set. This two-class problem can be solved by a support vector machine (SVM, the decision boundary is labeled with zero) with an acceptable classification performance on the given data. In a region not covered by the data the decision of the SVM changes arbitrarily.

Our main focus of interest is summarized in the following issues:

1. Generation of an interpretable and verifiable model,
2. Realization of a high predictive accuracy, and
3. Provision of the capability to include domain and/or expert knowledge.

The crucial aspect is to find a suitable trade-off between (1) and (2). It is obvious that complex models will be able to achieve a better performance on the available data. However, a higher complexity will lead to an increased effort for model verification. In this paper, we focus mainly on the issues (1) and (2).

In the following, we introduce two additive modeling strategies developed with the objective to provide a good trade-off between the interpretability and the predictive performance of a model: the first approach is the DecisionTree-like Additive Model (see Sect. 2.3) and the second one is the Non-hierarchical Additive Model (see Sect. 2.4). Sect. 2.5 summarizes both approaches and discusses related approaches. In Sect. 3 experiments performed with both algorithms are discussed. Sect. 4 concludes and points out some issues of current research.

2 Additive Models

This section first introduces the common definition of additive models as found in the literature (e.g. [2]). In the second part of this section, two approaches (Sect. 2.3 and Sect. 2.4) are introduced utilizing the advantages of additive modeling to deal with safety-related problems. Currently, both approaches are designed to solve a binary classification

problem $f : \mathbb{R}^n \rightarrow \{0, 1\}$ given an n -dimensional input space:

$$\begin{aligned} V^n &= X_1 \times X_2 \times \dots \times X_n = \prod_{i=1}^n X_i, \text{ where } X_i \subseteq \mathbb{R} \\ Y &= \{0, 1\} = f(X_1, X_2, \dots, X_n) \end{aligned} \quad (1)$$

i.e. the task is to determine the function estimate of f given an observed data set $\mathfrak{D} = \{(\vec{v}_1, y_1), \dots, (\vec{v}_m, y_m)\} \subset V^n \times Y$.

2.1 Generalized Additive Models

Additive models were proposed by [3] for the estimation of a multivariate regression function by an additive approximation. This approach yields two main advantages: the curse of dimensionality can be avoided and (local) interpretability can be provided.

A *generalized additive model* in the regression setting can be written as [4]:

$$E(Y|X_1, X_2, \dots, X_n) = \alpha + f_1(X_1) + f_2(X_2) + \dots + f_n(X_n), \quad (2)$$

where f_i are arbitrary smooth functions (e.g. cubic smoothing splines [2]).

To solve a binary classification problem, generalized additive models can be transformed to *generalized additive logistic models* by applying the *logit* link function ($\text{logit}(p) = \log(p/(1-p))$):

$$\log \left(\frac{P(Y = 1|(X_1, X_2, \dots, X_n))}{P(Y = 0|(X_1, X_2, \dots, X_n))} \right) = \alpha + \sum_{i=1}^n f_i(X_i). \quad (3)$$

The functions f_1, f_2, \dots, f_n are estimated by a backfitting algorithm (for details see e.g. [2]).

2.2 Additive Models for Safety-Related Problems

Basic Idea. The methods being introduced in the following are both greedy approaches to find an additive estimate of the unknown function $f : V^n \rightarrow \{0, 1\}$. Both methods are based on the projection of the high-dimensional data to low-dimensional subspaces. *Local models* g_j are trained on these subspaces. By regarding only low-dimensional subspaces a visual interpretation becomes feasible and, thus, the avoidance of unintended extrapolation behavior is possible. The ensemble of local models boosts the overall predictive accuracy and overcomes the limited predictive performance of each single local model, while the global model remains interpretable.

Projection of High-Dimensional Data. The projection π maps the n -dimensional input space V^n (see Eq. 1) to an arbitrary subspace of V^n . This mapping is determined by a given index set $\beta \subset \{1, \dots, n\}$. The index set defines the dimensions of V^n that will be included in the subspace V_β . Thus, the projection π on the input space V^n given the index set β is defined as:

$$\pi_\beta(V^n) = V_\beta = \prod_{i \in \beta} X_i. \quad (4)$$

Algorithm 1 Building a DecisionTree-like Additive Model.

parameter: \mathcal{D} – data set; dim_limit – limit of dimensions (fixed, typically 2 or 3)

```
function  $T := \text{build\_tree}(\mathcal{D})$ 
  find  $\beta \subset \{1, \dots, n\}$ , where  $|\beta| = \text{dim\_limit} \wedge \forall (\vec{v}, y) \in \mathcal{D} : |y - g(\pi_\beta(\vec{v}))| \rightarrow \min$ 
  create new node  $T$ 
   $\mathcal{D}_0 := \{(\vec{v}, y) | g(\pi_\beta(\vec{v})) = 0\}$ 
   $\mathcal{D}_1 := \{(\vec{v}, y) | g(\pi_\beta(\vec{v})) = 1\}$ 
  if  $((\mathcal{D}_0 \neq \emptyset) \wedge (\mathcal{D}_1 \neq \emptyset))$ 
     $T.\text{model} := g$ 
     $T.\text{beta} := \beta$ 
     $T.\text{child}[0] := \text{build\_tree}(\mathcal{D}_0)$ 
     $T.\text{child}[1] := \text{build\_tree}(\mathcal{D}_1)$ 
  fi
```

Algorithm 2 Classifying new samples with a DecisionTree-like Additive Model.

parameter: $\vec{v} \in V^n$ – new sample data point; T – tree returned by Algorithm 1

```
function class := evaluate_tree( $\vec{v}, T$ )
  votes := evaluate_tree_rek( $\vec{v}, T$ )
  class :=  $\Phi(\text{votes})$ 
function votes := evaluate_tree_rek( $\vec{v}, T$ )
   $g := T.\text{model}$ 
   $\beta := T.\text{beta}$ 
  votes :=  $g(\pi_\beta(\vec{v}))$ 
  if  $((g(\pi_\beta(\vec{v})) = 0) \wedge (T.\text{child}[0] \neq \emptyset))$ 
    votes := [votes evaluate_tree_rek( $\vec{v}, T.\text{child}[0]$ )]
  elseif  $((g(\pi_\beta(\vec{v})) = 1) \wedge (T.\text{child}[1] \neq \emptyset))$ 
    votes := [votes evaluate_tree_rek( $\vec{v}, T.\text{child}[1]$ )]
  fi
```

Local models. The j -th local model is defined as:

$$g_j : \pi_{\beta_j}(V^n) \rightarrow \{0, 1\}, \quad (5)$$

where β_j denotes the index set of the subspace where the classification error of the local model g_j is minimal. The final function estimate \hat{f} of the global model is determined by the aggregation of the results of all local models $g_j(\pi_{\beta_j}(\vec{v}))$. The summation of the original additive model, Eq. 2, is replaced by an appropriate aggregation function Φ . Thus, the final function estimate is:

$$\hat{f}(\vec{v}) = \Phi(g_j(\pi_{\beta_j}(\vec{v}))). \quad (6)$$

2.3 DecisionTree-like Additive Model

The basic idea of this approach is to replace the classification nodes in a common decision tree (e.g. CART or C4.5) by strong classifiers (e.g. support vector machines (SVM)). The procedures for building such a tree T and to evaluate a new sample \vec{v} with a given tree T are shown in Algorithm 1 and Algorithm 2, respectively.

Algorithm 3 Building a Non-hierarchical Additive Model.

parameter: \mathcal{D} – data set; c_{pref} – label of preferred class; dim_limit – limit of dimensions (fixed, typically 2 or 3)

```
function models := build_model( $\mathcal{D}$ ,  $c_{\text{pref}}$ )
  find  $\beta \in \{1, \dots, n\}$ , where  $|\beta| = \text{dim\_limit} \wedge \forall (\vec{v}, y) \in \mathcal{D} : |y - g(\pi_{\beta}(\vec{v}))| \rightarrow \min$ 
    w.r.t.  $\forall y = c_{\text{pref}} : |y - g(\pi_{\beta}(\vec{v}))| = 0$ 
     $\mathcal{D}_{\text{new}} := \{(\vec{v}, y) \mid g(\pi_{\beta}(\vec{v})) = c_{\text{pref}}\}$ 
    if ( $\mathcal{D} \setminus \mathcal{D}_{\text{new}} \neq \emptyset$ )
      models :=  $\{g(\pi_{\beta}(\cdot))\} \cup \text{build\_model}(\mathcal{D}_{\text{new}}, c_{\text{pref}})$ 
    else
      models :=  $\emptyset$ 
  fi
```

Algorithm 4 Classifying new samples with a Non-hierarchical Additive Model.

parameter: $\vec{v} \in V^n$ – new sample data point; models – set of models returned by Algorithm 3

```
function class := evaluate_model( $\vec{v}$ , models)
  class :=  $\bigvee_{g_j \in \text{models}} g_j(\pi_{\beta_j}(\vec{v}))$ 
```

The dimensionality of the classification nodes in the tree is restricted to two or three. This facilitates the visualization of the relevant decision region and avoids overfitting. The best local model g_j is used to divide the training set into two new subsets $\mathcal{D}_{\theta}^{\text{new}} := \{(\vec{v}, y) \mid g_j(\pi_{\beta_j}(\vec{v})) = \theta\}$, where $\theta \in \{0, 1\}$. The local models for these subsets are built recursively, until an appropriate termination criterion is fulfilled. Finally, the class label of a new sample \vec{v} is determined by aggregating the votes of the nodes for the given sample:

$$\hat{f}(\vec{v}) = \Phi(\{g_j(\pi_{\beta_j}(\vec{v})) \mid (g_j, \beta_j) \in \text{Exp}\}), \quad (7)$$

where Φ is an appropriate aggregation function and Exp is the set of nodes of tree T being involved by applying Algorithm 2 on \vec{v} . In our experiments (see Sect. 3) with SVMs as classifiers the following aggregation function provides adequate results:

$$\Phi(\vec{v}) = \frac{1}{|\text{Exp}|} \sum_{i \in \text{Exp}} \frac{1}{1 + e^{2m_i(\pi_{\beta_i}(\vec{v}))}}, \quad (8)$$

where $m_i(\pi_{\beta_i}(\vec{v}))$ is the margin of the i -th SVM model on subspace V_{β_i} , represented by $\text{exp}_i \in \text{Exp}$ – i.e. we compute a weighted average of the votes of the experts. Using Eq. 8 as aggregation function yields the classification function: $\hat{f}(\vec{v}) = 1 \Leftrightarrow \Phi(\vec{v}) \geq 0.5$, otherwise $\hat{f}(\vec{v}) = 0$.

2.4 Non-hierarchical Additive Model

The basic idea of this method is to incorporate prior knowledge about the subgroups of the given problem and to avoid hierarchical dependencies of the local models as in the previous approach. This method requires that the so-called *preferred class* c_{pref} must not be misclassified by any of the trained local models: $\forall y = c_{\text{pref}} : |y - g(\pi_{\beta}(\vec{v}))| = 0$. This requirement typically leads to imbalanced misclassification costs. The algorithms for building such a Non-hierarchical Additive Model and for evaluating a new sample \vec{v} are shown in Algorithm 3 and Algorithm 4, respectively.

Similar to the DecisionTree-like Additive Model of Sect. 2.3, the local models are trained on projections of the n -dimensional input space. In contrast to this approach, the local models are trained to avoid the misclassification of the preferred class. This can be achieved by imbalanced misclassification costs of the different classes or an additional penalty term for misclassified preferred class samples. The local models greedily separate the samples of the other class from the preferred class samples. Missed samples of the other class are used to build further sub-experts.

This yields the following disjunctive combination as final function estimate:

$$\hat{f}(\vec{v}) = \bigvee_{g_j \in Exp} g_j(\pi_{\beta_j}(\vec{v})), \quad (9)$$

where Exp is the set of all local models. For the sake of simplicity it is defined for Eq. 9 that the preferred class c_{pref} is always encoded as 0 and the other class is always encoded as 1 by the local models g_j .

2.5 Summary and Related Work

The *DecisionTree-like Additive Model* is a completely data-driven method, where no prior information is needed to build the model. The tree-like structure might become complex and might therefore lead to a high evaluation effort for the global model, since the sub-experts depend on their antecessors. On the other hand, misclassified samples can be rectified by succeeding nodes. Despite the increasing complexity of the tree structure this approach greatly facilitates the interpretation of the global model – compared to high-dimensional non-symbolic models. This approach is appropriate for problems with balanced misclassification costs.

The *Non-hierarchical Additive Mode* facilitates the integration of expert knowledge about the *preferred class* and about the existence of subgroups within the data set (see Sect. 3.2). All local models are trained to avoid the misclassification of the preferred class – even if this results in a higher misclassification rate of the other class. Dealing with the preferred class yields a flat structure of the global model leading to a reduced evaluation effort for the overall model, since all sub-experts can be interpreted separately. This approach is appropriate for problems with imbalanced misclassification costs.

Related Work. Here we will focus on algorithmically-related approaches. An overview of other machine learning approaches for safety-related domains can be found in [1]. In contrast to hierarchical mixture of experts methods [5], the local models in our approaches are built sequentially. The approaches introduced here use a much simpler aggregation function of the local models compared to the gating networks of a hierarchical mixture of experts. This ensures an enhanced interpretability of the interaction of the local models, where the interpretability of each local model is ensured by limiting its number of input dimensions. Similar to Boosting [6, 7] the basis functions of the additive model are replaced by individual classifiers g_j . The AdaBoost algorithm² [8] trains a large number of rough and moderately inaccurate classifiers, so-called *weak classifiers*, on weighted versions of the training data, where currently misclassified samples are given a higher weight. Our approaches use a much smaller number of local models, which are built

²AdaBoost, short for Adaptive-Boosting, is the most widely used Boosting algorithm.

greedily. Since the number of local models influences the interpretability of the global model, a small number of local models is preferred. Furthermore, for safety-related problems, it has to be ensured that the local models show a better predictive performance than a weak classifier.

There are several approaches that are related to the methods introduced in this paper:

- In [9] a learning framework called *probabilistic boosting tree* is proposed for learning two- and multi-class discriminative models: the target distribution is learned by constructing a tree in which each node is a strong decision maker. In contrast to our approaches, it is used to model conditional probability distributions given previous decisions.
- *Cascade generalization* is proposed by [10]. The idea of this approach is to extend the original data set with new attributes generated by previously trained local models. The new attributes facilitate succeeding local models to build more complex decision surfaces. Unfortunately, the rising complexity of the new attributes increases the effort to evaluate the resulting model.
- In classical *separate-and-conquer* approaches, also known as *covering*, like rule induction – rules are learned to explain parts of the training samples. A survey can be found in [11]. The samples covered by a rule are removed from the training set (the so-called *separate* part) and further rules are learned on the remaining samples (the *conquer* part). Often, the resulting rules cannot represent complex relationships sufficiently, because they are limited in their expressive power compared to the local models of our approaches.
- An algorithm to solve multi-class problems, which is based on *pairwise coupling* of the local models, is given in [12]. This algorithm uses pairwise combined local models to extend binary classifiers to solve multi-class problems. The important similarity of this approach to our approaches is that it is also based on a dimension reduction on the level of local models. In contrast to our approaches, the number of dimensions of the local models is not limited – all input dimensions that provide statistically sufficient information are included in the training set to build a single local model to separate the pair of classes. Our approach may use several local models with limited dimensionality to solve the same subproblem.

3 Experiments

This section discusses some of the investigated experiments: in Sect. 3.1 an illustrative example is discussed. In Sect. 3.2, the applicability of our approaches for real-world data sets with high safety requirements is shown.

In all experiments, the best projection is determined by a wrapper method for feature selection [13]. This ensures the best possible performance on the given training data with the chosen learning algorithm. In all experiments support vector machines (SVMs) with a Gaussian kernel are used as local models and the number of input dimensions used in the projections is restricted to two.

3.1 An Illustrative Example

The CUBES data set is generated from four Gaussian components in a three-dimensional space. This data set is illustrated in Fig. 2: Fig. 2(a) visualizes the data set in 3D and

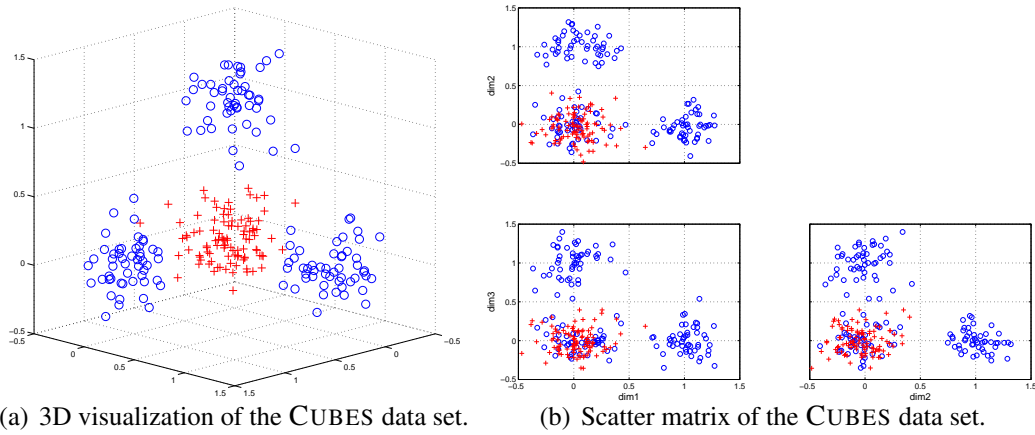


Figure 2: Three-dimensional CUBES data set: CLASS 1 samples are marked with a blue circle and CLASS 0 samples are marked with red crosses.

(a) Non-hierarchical Additive Model.			(b) DecisionTree-like Additive Model.		
true class	predicted class		true class	predicted class	
	CLASS 0	CLASS 1		CLASS 0	CLASS 1
CLASS 0	100	0	CLASS 0	97	3
CLASS 1	4	146	CLASS 1	0	150

Table 1: Confusion matrices of the three-dimensional CUBES data set.

Fig. 2(b) visualizes all possible two-dimensional projections of the three-dimensional data set as a scatter matrix. For each CLASS 1 cluster 50 samples are drawn from $N(e_i, 0.2 \cdot \mathbf{I})$, where e_i is a unit vector and \mathbf{I} is the identity matrix. 100 samples of the CLASS 0 cluster are scattered around the origin, drawn from $N((0, 0, 0)^T, 0.2 \cdot \mathbf{I})$. All local models are trained as SVMs with Gaussian kernel and the parameter set $\gamma = 0.2$ and $C = 5$.

Non-hierarchical Additive Model. For the Non-hierarchical Additive Model CLASS 0 is selected as the preferred class, $c_{\text{pref}} = 0$, i.e. this class must not be misclassified by any learned local model. In this example, this is achieved by using imbalanced misclassification costs for CLASS 1 and CLASS 0, where the misclassification penalty of CLASS 0 is ten times higher than for CLASS 1. At the initial state, all two-dimensional projections of the CUBES data set are very similar, see Fig. 2(b). The best local model g_1 , see Fig. 3(a), uses the projection $\pi_{\beta_1}(\vec{v})$ with $\beta_1 = \{1, 2\}$. 53 data points from CLASS 1 are misclassified by this local model. Thus, in the next iteration new local models are trained only on samples which are predicted as CLASS 0 by the first local model: $\mathcal{D}_{\text{new}} = \{(\vec{v}, y) | g_1(\pi_{\beta_1}(\vec{v})) = 0\}$. In Fig. 3(b) the projection $\pi_{\beta_2}(\vec{v})$ with $\beta_2 = \{2, 3\}$ of the data set \mathcal{D}_{new} and the corresponding local model g_2 are shown. This local model misclassifies four CLASS 1 samples. Further improvements with the given parameter set are not possible. The final additive model is $\hat{f}(\vec{v}) = g_1(\pi_{\beta_1}(\vec{v})) \vee g_2(\pi_{\beta_2}(\vec{v}))$, i.e. the local models can be interpreted independently. The overall performance of the Non-hierarchical Additive Model is shown in Tab. 1(a): avoiding the misclassification of the preferred class $c_{\text{pref}} = 0$ leads to four misclassified CLASS 1 samples. Relaxing the misclassification penalty of CLASS 0 to a factor of four reduces the number of misclassified CLASS 1 samples to one but the risk to misclassify novel CLASS 0 instances increases.

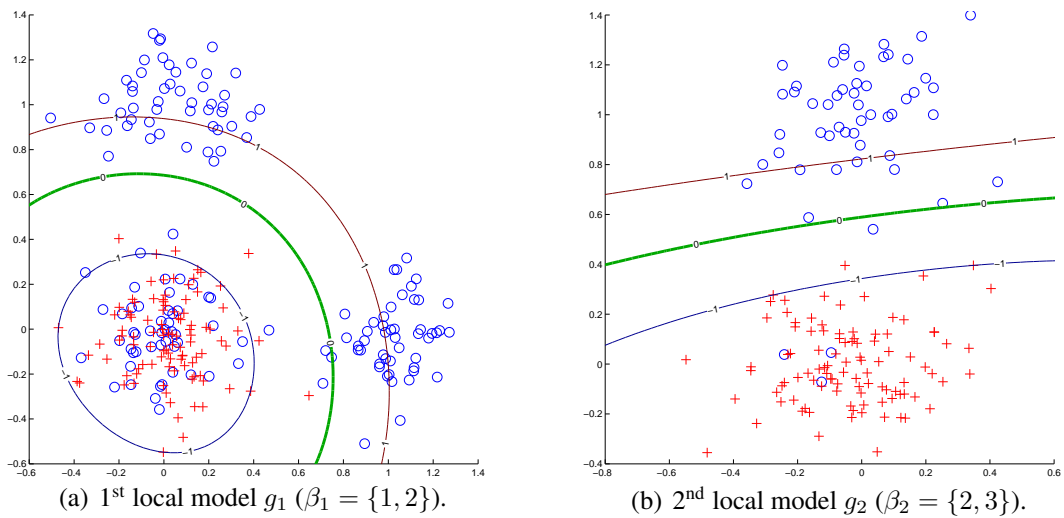


Figure 3: Non-hierarchical Additive Model and the CUBES data set: CLASS 1 samples are marked with blue circles and CLASS 0 samples are marked with red crosses. The decision boundary is labeled with zero and the margin of the SVM model is labeled with -1 and 1.

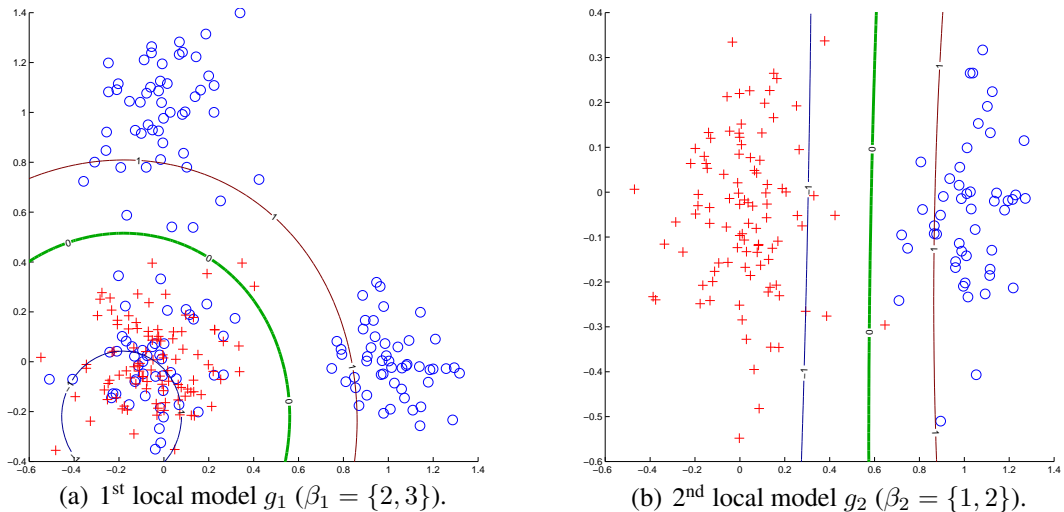


Figure 4: DecisionTree-like Additive Model and the CUBES data set: CLASS 1 samples are marked with blue circles and CLASS 0 samples are marked with red crosses. The decision boundary is labeled with zero and the margin of the SVM model is labeled with -1 and 1.

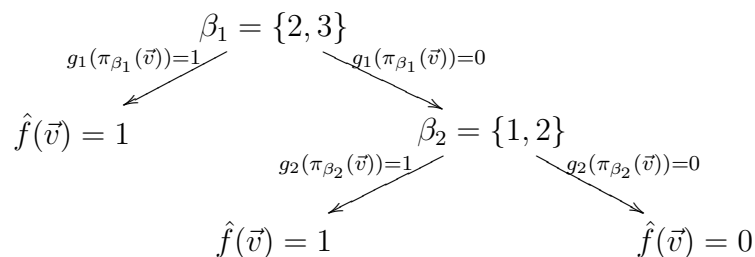


Figure 5: DecisionTree-like Additive Model: the index sets of each local model are given at the inner nodes. The models prediction is denoted by \hat{f} . For the sake of simplicity the aggregation of the votes of the local models (see Eq. 8) is omitted in this example.

DecisionTree-like Additive Model. In contrast to the first local model of the Non-hierarchical Additive Model, the local model g_1 of the root node of the DecisionTree-like Additive Model, as depicted in Fig. 4(a), uses the projection $\pi_{\beta_1}(\vec{v})$ with $\beta_1 = \{2, 3\}$. The decision boundary is shifted toward the region of the CLASS 0 samples because the misclassification penalty term is ignored by this approach. This local model misclassifies two CLASS 0 samples and 49 CLASS 1 samples. Thus, two new data sets are created according to Algorithm 1: $\mathcal{D}_1 = \{(\vec{v}, y) | g_1(\pi_{\beta_1}(\vec{v})) = 1\}$ and $\mathcal{D}_0 = \{(\vec{v}, y) | g_1(\pi_{\beta_1}(\vec{v})) = 0\}$. For \mathcal{D}_1 there are no further improvements possible given the parameter set. But on \mathcal{D}_0 a second local model g_2 with the projection $\pi_{\beta_2}(\vec{v})$ with $\beta_2 = \{1, 2\}$ can be determined, see Fig. 4(b). This local model misclassifies only one CLASS 0 sample – further models cannot improve the overall performance. The final model tree is depicted in Fig. 5. The confusion matrix of the DecisionTree-like Additive Model is shown in Tab. 1(b) – without the incorporation of prior knowledge (i.e. defining the preferred class) three CLASS 0 samples are misclassified.

3.2 Application in Automotive Safety Electronics

In this problem, based on a high-dimensional data set, a decision has to be derived whether to trigger a specific restraint system (1 = FIRE) or not (0 = NOFIRE). An example for such a restraint application is the deployment of an airbag system. This task is challenging because (1) the restraint system can be triggered only once – i.e. a wrong decision cannot be rectified – and (2) a malfunction of the system might be fatal. Thus, it must be ensured that the obtained model is sensitive enough to trigger the restraint system and robust enough to avoid unintended extrapolation or interpolation behavior – as illustrated in Fig. 1. The domain experts in the field of automotive safety electronics tend to use conservative models (e.g. rule-based systems or lookup tables) to avoid an undesired behavior. Unfortunately, by generating such models by purely data-driven methods, the resulting models might become complex and hard to verify.

The data set in this application example is sparse and consists of approx. 40 000 data points, each 30-dimensional, from 40 distinct time series. Each time series represents a certain standardized crash test situation. Due to the limited number of crash tests, the guaranteed extrapolation (and interpolation) behavior of our models is crucial. As the data set consists of time series, it is sufficient to trigger the restraint system once in a defined time interval. The incorporation of domain knowledge facilitates the reduction of the model complexity by dividing the FIRE class into subgroups (e.g. FIRE.1 and FIRE.2). These subgroups represent different types of crash test situation setups, which implicate different input signals. Thus, the original problem can be separated into two distinct subproblems (FIRE.1 versus NOFIRE and FIRE.2 versus NOFIRE).

DecisionTree-like Additive Model. This approach yields a complex tree structure with ten local models, see Fig. 6. The maximal tree depth is five. No NOFIRE sample is triggered. Unfortunately, not all FIRE samples can be correctly classified. This problem arises due to the greedy selection of the best split: it is easier to separate FIRE.1 samples than to separate FIRE.2 samples from the NOFIRE samples. Thus, the tree building procedure focusses mainly on the FIRE.1 samples. All misclassified samples belong to the FIRE.2 subgroup. To capture the missed samples, it is necessary to include this domain knowledge (see Non-hierarchical Additive Model) and to learn distinct models per subgroup.

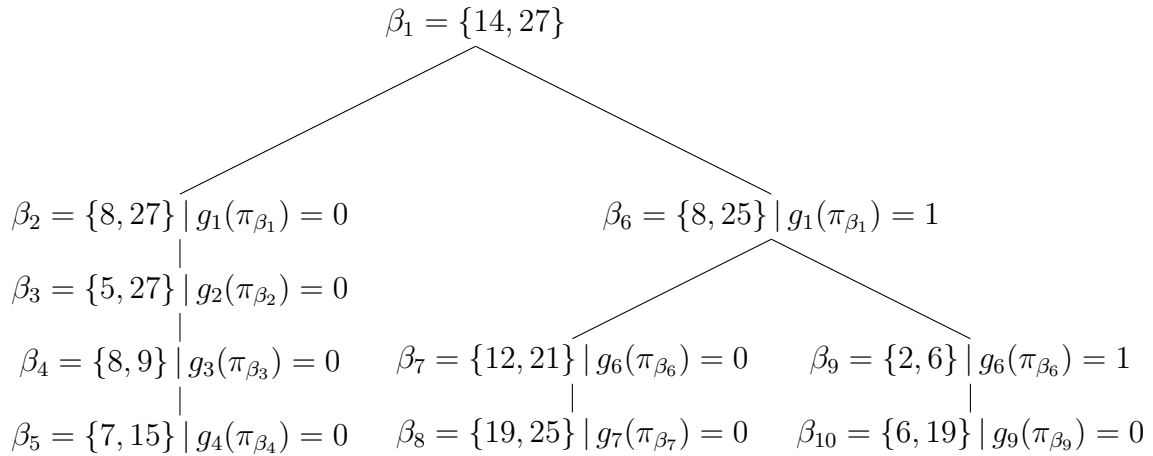


Figure 6: DecisionTree-like Additive Model and the application in automotive safety electronics: index sets β_j per node of the DecisionTree-like Additive Model for the automotive data set. $\beta_l = \{\cdot\} | g_k(\pi_{\beta_k}) = \theta$ states that the l -th local model is trained on samples labeled as θ by the k -th local model.

Non-hierarchical Additive Model. The problem can be solved by an ensemble of four two-dimensional local models – two local models for the FIRE.1 subgroup and two local models for the FIRE.2 subgroup. As an illustration, the local models per subgroup are shown in Fig. 7. All local models have smooth decision boundaries that are adequate according to expert/domain knowledge. As for the DecisionTree-like Additive Model, no NOFIRE sample is triggered. Due to the capability to divide the data set into two different subgroups and to combine the learned local models disjunctively, all FIRE samples can be triggered by at least one expert.

Summary. In the described application, the solution of the DecisionTree-like Additive Model becomes quite complex, because additional domain knowledge cannot be sufficiently incorporated. The Non-hierarchical Additive Model allows the incorporation of additional domain knowledge by separating the FIRE class into two subgroups. The resulting global models per subgroup can be easily combined due to the disjunctive combination of the local models. Thus it is possible to combine the final models of the different subgroups also by a disjunctive combination. This shows that a simple solution of this problem can be achieved by incorporating additional domain knowledge to the model learning procedure.

4 Conclusions

To apply machine learning approaches in the field of safety-related problems, it is crucial to provide interpretable and verifiable models. Unfortunately, it is infeasible to sufficiently interpret high-dimensional models. Therefore, high-dimensional models are not to be applied for safety-related applications. On the other hand, simple models, which are easier to interpret, show a lack of predictive performance.

The binary classification approaches introduced in this paper provide a good trade-off between the interpretation and verification of the learned (local) models, avoiding an unintended extrapolation behavior, and the achievement of a high predictive accuracy. Each

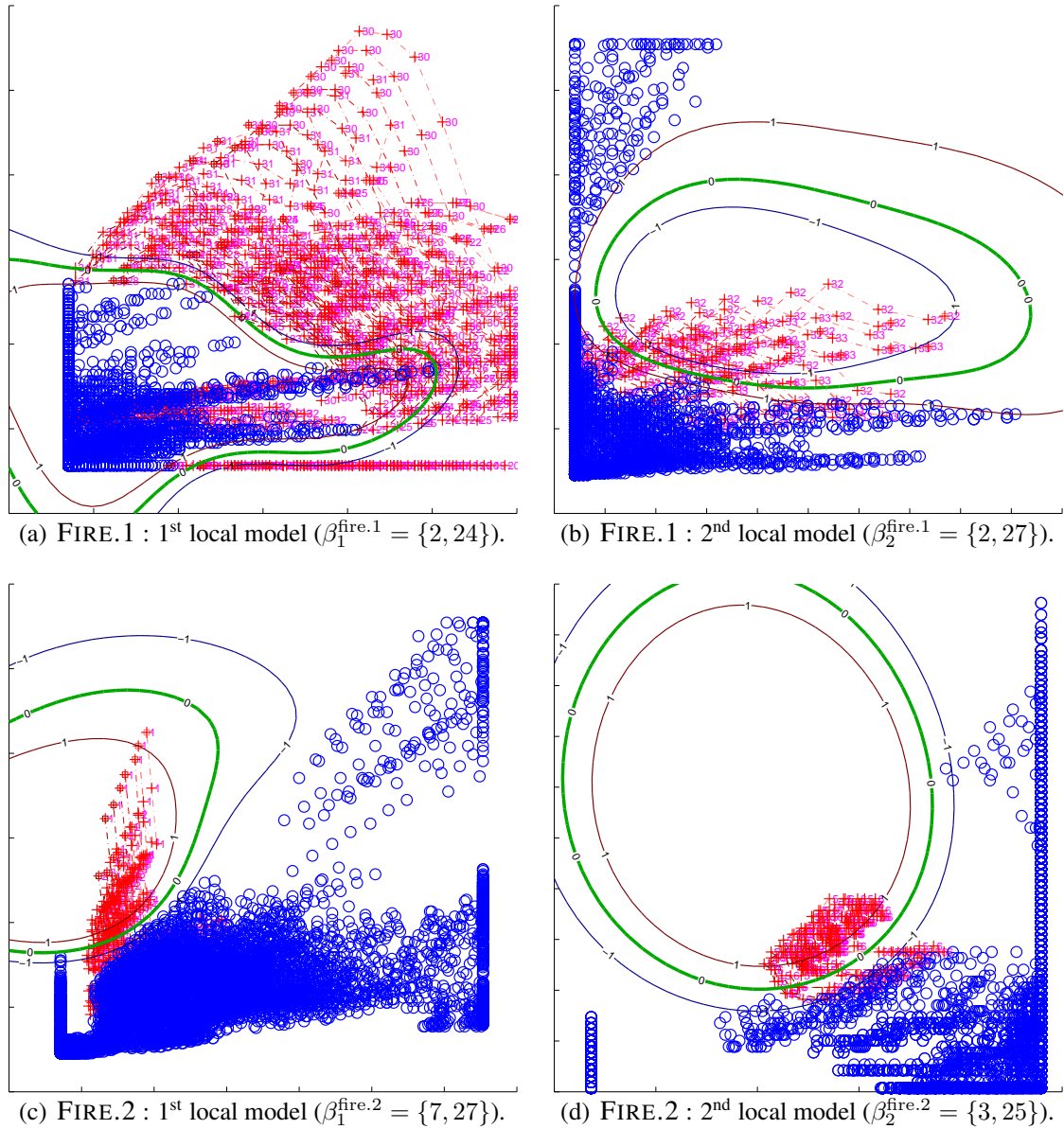


Figure 7: Non-hierarchical Additive Model and the application in automotive safety electronics: NOFIRE samples are marked with blue circles and FIRE samples are marked with red crosses. The trajectories of the FIRE samples are shown as broken lines. The decision boundary is labeled with zero and the margin of the SVM model is labeled with -1 and 1.

local model can be interpreted visually and the ensemble of the local models compensates for the limited predictive performance of each single local model. The local models can be evaluated by a domain expert to avoid unintended extrapolation and interpolation behavior. In contrast to dimensionality reduction methods, which combine several dimensions of the input space, the local models are trained on the original dimensions, allowing the experts to evaluate the trained models directly. In our experiments, the real-world application is tested successfully providing good results. Both additive modeling approaches can be modified to utilize other classification methods instead of SVMs.

Outlook. Currently, the approaches are being extended to also deal with multi-class problems. Furthermore, a feature construction method based on multi-layer-perceptrons to provide additional input dimensions for the additive modeling procedure is investigated. These additional features provide simple linear combinations of several original input dimensions. Thus, enabling the approach to overcome the limitation of two- or three-dimensional local models while the local models still remain interpretable.

References

- [1] Otte, C.; Nusser, S.; Hauptmann, W.: Machine Learning Methods for Safety-Related Domains: Status and Perspectives. In: *Proceedings of FSCS*, S. 139–148. Magdeburg, Germany. 2006.
- [2] Hastie, T.; Tibshirani, R.; Friedman, J.: *The Elements of Statistical Learning*. Springer. 2001.
- [3] Stone, C. J.: Additive Regression and Other Nonparametric Models. *The Annals of Statistics* 13 (1985) 2, S. 689–705.
- [4] Hastie, T.; Tibshirani, R.: *Generalized Additive Models*. Chapman & Hall. 1990.
- [5] Jordan, M. I.; Jacobs, R. A.: Hierarchical mixtures of experts and the EM algorithm. *Neural Computation* 6 (1994) 2, S. 181–214.
- [6] Schapire, R. E.: The Strength of Weak Learnability. *Machine Learning* 5 (1990), S. 197–227.
- [7] Schapire, R. E.: The boosting approach to machine learning: An overview. In: *Workshop on Nonlinear Estimation and Classification*. MSRI. 2002.
- [8] Freund, Y.; Schapire, R. E.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences* 55 (1997) 1, S. 119–139.
- [9] Tu, Z.: Probabilistic Boosting-Tree: Learning Discriminative Models for Classification, Recognition, and Clustering. In: *Proceedings of ICCV*, S. 1589–1596. Beijing, China. 2005.
- [10] Gama, J.; Brazdil, P.: Cascade Generalization. *Machine Learning* 41 (2000) 3, S. 315–343.
- [11] Fürnkranz, J.: Separate-and-Conquer Rule Learning. *Artificial Intelligence Review* 13 (1999) 1, S. 3–54.
- [12] Szepanek, G.; Weihs, C.: Local Modelling in Classification on Different Feature Subspaces. In: *Industrial Conference on Data Mining*, S. 226–238. 2006.
- [13] Kohavi, R.; John, G. H.: Wrappers for feature subset selection. *Artificial Intelligence* 97 (1997) 1-2, S. 273–324.

Schnelles Resource Constrained Project Scheduling mit dem Evolutionären Algorithmus GLEAM

Wilfried Jakob¹, Alexander Quinte¹, Karl-Uwe Stucky¹,
Wolfgang Süß¹, Christian Blume²

¹Forschungszentrum Karlsruhe GmbH, Institut für Angewandte Informatik
Postfach 3640, 76021 Karlsruhe
Tel. (07247) 82-4663

E-Mail: {wilfried.jakob, alexander.quinte, uwe.stucky, wolfgang.suess}@iai.fzk.de

²FH Köln, Campus Gummersbach
51643 Gummersbach, Am Sandberg 1
Tel. (02261)8196-6297
E-Mail: christian.blume.@fh-koeln.de

Zusammenfassung

Probleme aus dem Bereich des *Resource Constrained Project Scheduling* gehören zu den ambitioniertesten Scheduling-Aufgaben, die im OR-Bereich derzeit angegangen werden. Für die hier vorgestellte Anwendung des Scheduling von Jobs und Ressourcen im Grid kommt erschwerend hinzu, dass die Planung innerhalb weniger Minuten erfolgen muss. Vorgestellt wird eine auf dem Evolutionären Algorithmus GLEAM basierende hybride Lösung, deren Ergebnisse zwar suboptimal sind, aber die Resultate einer vorausgegangenen heuristischen Planung deutlich verbessern. Der Beitrag vergleicht zwei Genmodelle und untersucht die Rolle von geno- und phänotypischer Reparatur.

1 Einleitung

Ausgangspunkt ist das Problem der Planung von Anwendungsjobs in einem heterogenen Rechnernetz basierend auf Grid-Technologien unter Berücksichtigung von Kosten, Zeiten, Auslastung und spätesten Fertigstellungsterminen [1]. In Abschnitt 2 wird gezeigt, dass diese Aufgabe wesentlich komplexer ist als das typische Jobshop-Scheduling-Problem der OR-Literatur [2, 3], denn es geht um mehr, als nur n Aufträge auf m Maschinen unter Beachtung von Reihenfolgerestriktionen mit möglichst geringer Bearbeitungsdauer zu verteilen. Abschnitt 3 beschreibt die Lösungsstrategien basierend auf dem Evolutionären Algorithmus (EA) GLEAM (General Learning Evolutionary Algorithm and Method) [4, 5] unter Verwendung einfacher Heuristiken, unterschiedlicher Genmodelle, Koevolution und unterschiedlichen Reparaturmechanismen. Der Frage welcher dieser zum Teil alternativen Ansätze in welcher Kombination die besten Ergebnisse liefert, wird in Abschnitt 4 nachgegangen. Dort werden auch die dazu verwendeten Benchmarkszenarien für die zu planende Last vorgestellt. Abschnitt 5 zieht ein Resümee und gibt einen Ausblick auf weitere Arbeiten.

2 Das „GORBA-Problem“

Die Planung der Zuordnung von Aufträgen zu Maschinen ist bereits ohne die Berücksichtigung weiterer in der Praxis relevanter Restriktionen ein NP-vollständiges Problem [2]. Im Kontext der Auftragsplanung in einem Grid wird das Problem noch dadurch verschärft, dass zur Planung nur wenig Zeit (wenige Minuten) zur Verfügung steht, da

auf Grund der Dynamik des Systems (neue Jobs, längere oder kürzere tatsächliche Bearbeitungszeiten, Ressourcenausfälle, neue Ressourcen, usw.) immer wieder umgeplant werden muss. Bisher werden auf Grund der Komplexität des Planungsproblems meist nur Queuing-Systeme eingesetzt, die unter Verwendung von Benutzerangaben zu den Bearbeitungszeiten der einzelnen Gridjobs nur eine grobe Aussage über die zu erwartenden Kosten machen, die gesamte Bearbeitungszeit hingegen gar nicht abschätzen können [6]. Außerdem nutzen sie die verfügbaren Ressourcen ab einer gewissen Last in der Regel nur suboptimal aus. Der Einsatz von vorausschauender Planung setzt gegenüber Queuing-Systemen die Verfügbarkeit von zusätzlichen Informationen über die zu erwartende Bearbeitungsdauer der Teiljobs einer Anwendung sowie über die langfristige Verfügbarkeit der Ressourcen voraus. Der Lohn für diesen zusätzlichen Aufwand besteht in einer, wie noch gezeigt werden wird, qualitativ besseren Planung und in der Möglichkeit, dem Anwender vor der Ausführung eine Aussage über Kosten und Fertigstellungszeiten seines Rechenauftrages machen zu können.

Zunächst soll das Optimierungs- und Planungsproblem genauer beschrieben werden. Ein Anwendungsjob besteht aus elementaren Schritten, den so genannten Gridjobs, zwischen denen Reihenfolgebeziehungen bestehen, die sich durch einen azyklischen gerichteten Graphen (Workflow) beschreiben lassen. Derartige Workflows sind statisch in dem Sinne, dass sie ohne Modifikationen, die sich erst zur Laufzeit ergeben, abgearbeitet werden können. Dynamische Workflows enthalten dagegen Entscheidungen, die erst während der Abarbeitung getroffen werden können. Einer Planung sind daher nur die statischen Teilstücke eines solchen Workflows zugänglich und zu seiner Abarbeitung sind entsprechend viele Planungen notwendig. Nachfolgend betrachten wir nur statische Workflows.

Jeder Gridjob benötigt zur Ausführung mindestens eine Ressource, nämlich einen Rechner, den er ausschließlich belegt. Zusätzlich können Software-Ressourcen erforderlich sein, deren quantitative Verwaltung auf Grund von Lizenzvorgaben notwendig sein kann. Außerdem können Ressourcen zur Datenhaltung und Kommunikation eine Rolle spielen oder spezielle Gerätesressourcen (z.B. ein Mikroskop) benötigt werden. Im Allgemeinen benötigen Gridjobs primäre Ressourcen, die ihrerseits meist weitere Ressourcen benötigen, so kann z.B. eine SW-Ressource eine Hardware-Ressource aus einer speziellen Gruppe erfordern, auf der sie lauffähig ist (Hardware und Betriebssystem werden dabei als Einheit angesehen). Damit sind in der Regel Koallokationen verschiedener Ressourcen erforderlich. In einem heterogenen Netz kommt noch hinzu, dass die Ausführungszeiten und -kosten je nach eingesetzter Rechnerhardware unterschiedlich sind und auch über der Zeit variieren können (z.B. wegen tageszeit-basierter Kostenmodelle). Zusammen mit der nachfolgend beschriebenen Bewertungsfunktion ergibt sich ein Scheduling- und Ressourcen (Ko-)Allokationsproblem, das auf Grund seiner Spezifika nach dem von uns verwendeten Werkzeug GORBA (**G**lobal **O**ptimising **R**esource **B**roker and **A**llocator) benannt wurde [7]. Wegen den Beschränkungen und den differenzierten Kostenfunktionen der Ressourcen gibt es eine Reihe von Ähnlichkeiten mit den *resource-constrained project scheduling* Problemen [8].

Die Bewertung erfolgt nach folgenden Kriterien, wobei jeweils obere und untere Grenzen abgeschätzt oder vom Benutzer vorgegeben werden (Kosten- und Zeitlimit sowie eine Präferenz für kosten- oder zeitgünstige Planung). Die Bewertung erfolgt dann relativ zu diesen Grenzen an Hand folgender Kriterien:

- Relative Bearbeitungsdauer der einzelnen Anwendungsjobs
Die Bearbeitungszeit eines Anwendungsjobs wird relativ zu seinen geschätzten Grenzen bewertet und der Durchschnitt aller Anwendungsjobs gebildet.

- Gesamtdauer der Bearbeitung aller Anwendungsjobs
- Relative Kosten der einzelnen Anwendungsjobs
Ermittlung der relativen Einsparung gemessen an dem Kostenbudget des Jobs und Bildung des Durchschnitts aller Anwendungsjobs.
- Auslastung der Hardware-Ressourcen
Da die Auslastung am Ende einer Planung naturgemäß sinkt, werden nur die ersten 75% der Bearbeitungsdauer bewertet.

Aus diesen vier Kriterien wird eine gewichtete Summe gebildet, die bei Verletzung von Zeit- oder Kostengrenzen durch einzelne Jobs mit einem Straffaktor multipliziert wird. Bestraft werden der zeitliche Verzug in Form der Zeitüberschreitung und der Anzahl verspäteter Anwendungsjobs sowie Umfang und Anzahl von Kostenüberschreitungen.

Die relative Bewertung unter Verwendung von vor der Optimierung geschätzten Grenzen hat den Vorteil, dass die Bewertungsfunktion sich automatisch den Gegebenheiten einer konkreten Planungsaufgabe anpasst und keine manuelle Justierung erforderlich ist. Dies ist für eine Planungskomponente, die in einem automatisierten Prozess ohne Benutzereingriff arbeiten soll, notwendig. Aus diesem Grund verbietet sich auch eine Pareto-basierte Bewertung.

Die vorgestellte Bewertung berücksichtigt alle den Autoren bekannte praxisrelevanten Kriterien und kann auch leicht um weitere erweitert werden. Allerdings impliziert diese umfassende Sichtweise auch, dass kleine Änderungen wie sie z.B. durch heuristische lokale Suchverfahren vorgenommen werden, nicht mehr lokal bewertet werden können, sondern der gesamte Schedule neu aufgestellt werden muss. Das unterscheidet diese von anderen kombinatorischen Aufgaben, wie z.B. dem in der Literatur häufig zitierten Travelling-Salesman-Problem, bei dem nach Vertauschung zweier Städte sofort und ohne den gesamten Weg zu betrachten, berechnet werden kann, ob die Lösung dadurch besser wird. Ähnliches geht bei der Vertauschung der Reihenfolge zweier Gridjobs bedauerlicherweise nicht.

3 Der Lösungsansatz: Evolutionärer Algorithmus GLEAM erweitert um Heuristiken

GLEAM wurde bereits erfolgreich für ein komplexes kombinatorisches Problem eingesetzt, nämlich die Planung von Roboterbewegungsabläufen [9]. Dabei spielt sowohl die Reihenfolge der Achsbewegungsbefehle als auch ihre Parametrierung eine Rolle, sodass neben dem rein-kombinatorischen auch ein Parameteroptimierungsproblem zu lösen ist. Dazu wurden unter anderem Gene (Aktionen im GLEAM-Jargon) definiert, die jeweils einem Achsmotor zugeordnet sind und deren Parameter die Ansteuerung quantitativ vorgeben. Ähnlich ist es bei der vorliegenden Aufgabenstellung, bei der man jedem Gridjob ein Gen zuordnen kann und die Parameter eine Auswahl aus den möglichen Ressourcen treffen. Bei der evolutionären Veränderung eines solchen im Chromosom abgebildeten Schedules kommen die durch die Roboterbahnplanung motivierten Besonderheiten von GLEAM zum Tragen [5]: Durch die Segmentbildung mit mutierbaren Grenzen und Strukturen ergeben sich Makromutationen wie

- Inversion
Umkehrung der Genreihenfolge in einem Segment
- Segmentverschiebung
Verschiebung eines ganzen Segments im Chromosom

- Segmentverschmelzung
Bei nicht benachbarten Segmenten schließt diese Mutation eine Segmentverschiebung mit ein.

Hinzu kommt, dass die Segmente die Grenzen für das Crossover festlegen. Damit stellt bereits das Standard-GLEAM [4, 5] deutlich mehr Mutationen zur Verfügung als nur eine Verschiebung einzelner Gene und damit Gridjobs.

3.1 Einfache Heuristiken zu Initialisierung der Startpopulation

GORBA plant und optimiert in zwei Stufen: Zuerst erfolgt eine schnelle heuristische Planung, die bei entsprechend geringer Auslastung des Grids bereits ausreichen kann. Im Verlauf dieser Planung erfolgt eine Aufgabenanalyse mit Ermittlung des kritischen Pfads und der Unter- und Obergrenzen für Kosten und Zeiten. Die Ergebnisse der ersten Stufe werden für die nachfolgende Verbesserung des Schedules durch GLEAM zur automatischen Justierung der relativen Bewertung und zur Initialisierung der Startpopulation verwendet.

Die heuristische Planung erstellt zunächst zwei Gridjob-Sequenzen entsprechend der Regeln

- Gridjobs des Anwendungsjobs mit dem frühesten Fertigstellungstermin zuerst
- Gridjobs des Anwendungsjobs mit der kürzesten Bearbeitungszeit zuerst.

Die Zuordnung der Ressourcen erfolgt nach folgenden drei heuristischen Regeln (Ressourcenauswahlstrategien (RAS)), wobei die ersten beiden für alle Anwendungsjobs gleichermaßen gelten:

- billigste Ressource, die zum frühest möglichen Zeitpunkt frei ist, wählen
- schnellste Ressource, die zum frühest möglichen Zeitpunkt frei ist, wählen
- eine im Anwendungsjob hinterlegte Präferenz entscheidet, nach welcher der beiden obigen Regeln bei diesem Anwendungsjob vorzugehen ist.

Damit entstehen insgesamt sechs Schedules, mit denen die Startpopulation von GLEAM initialisiert wird, und die auch ein Maß für den Erfolg der nachfolgenden Planung darstellen.

3.2 Zwei alternative Genmodelle

Der Aufbau des ersten Genmodells GM1 wurde bereits angedeutet: Jedem Gridjob wird ein Gen (oder eine Aktion) zugeordnet. Der zum Gen gehörige Aktionstyp enthält so viele Parameter, wie der Gridjob insgesamt an Ressourcen (primäre und abhängige) benötigt, wobei jeder Parameter einen Index in der Menge der jeweils alternativ wählbaren Ressourcen darstellt. Die dazu notwendige Analyse erfolgt in der ersten Phase und das Ergebnis wird zur Aufstellung des Genmodells für den nachfolgenden GLEAM-Lauf verwendet. Parametermutationen bedeuten bei diesem Genmodell die Auswahl einer anderen Ressource. Die Interpretation eines Chromosoms zur Aufstellung der Belegungsmatrix erfolgt für alle Gene wie in Bild 1 dargestellt.

Beim zweiten Genmodell GM2 gibt es ebenfalls pro Gridjob eine Aktion, aber diesmal ohne Parameter. Dafür steuert eine zusätzliche Aktion die Wahl einer der drei zuvor beschriebenen Ressourcenauswahlstrategien (RAS). Eine Mutation dieses Parameters bedeutet also eine Änderung der Heuristik zur Ressourcenauswahl. Bild 2 zeigt den Pseudocode zur Gen-Interpretation für GM2. Dem offensichtlich komplizierteren (und laufzeitintensiveren) Algorithmus zur Ressourcenbelegung steht eine nicht unerhebliche

```

1. IF Gridjob hat Vorgänger THEN
    Bestimme deren spätestes Arbeitsende in start_time
ELSE
    start_time = früheste Startzeit des Anwendungsjobs
2. Berechne die Bearbeitungsdauer auf den gegebenen Res-
    sourcen
3. Suche ein freies Zeitintervall der berechneten Länge
    ab start_time für alle Ressourcen und belege sie.

```

Bild 1: Pseudocode zur Aufstellung der Belegungsmatrix für Genmodell 1 (GM1)

Reduktion des Suchraums gegenüber. GM1 kann alle alternativen Ressourcen allen Gridjobs zuordnen, während GM2 in dieser Hinsicht eingeschränkt ist: Entweder alle Anwendungsjobs bevorzugen billige oder schnelle Ressourcen oder dies geschieht, etwas flexibler, entsprechend der Präferenz des Anwendungsjobs. Welches Genmodell in dem gewählten Zeitrahmen von drei Minuten für einen GLEAM-Lauf die besseren Resultate bringt, kann nicht vorhergesagt und muss experimentell ermittelt werden. Auch ohne Versuche ist dagegen klar, dass GM1 bei hinreichend langen Laufzeiten mindestens gleich gute, wenn nicht bessere Ergebnisse liefern wird, da für GM2 nicht alle Bereiche des Suchraums von GM1 zugänglich sind.

```

1. IF Gridjob hat Vorgänger THEN
    Bestimme deren spätestes Arbeitsende in start_time
ELSE
    start_time = früheste Startzeit des Anwendungsjobs
2. Aufstellung einer Liste alternativer Ressourcen pro
    primärer Ressource des Gridjobs gemäß der aktuellen
    RAS
3. Berechnung der Bearbeitungszeit beginnend mit den ers-
    ten Ressourcen in den Listen und Suche eines freien
    Zeitintervalls für alle Ressourcen einschließlich der
    abhängigen ab start_time. Wenn kein geeignetes Zeitin-
    tervall gefunden wird, werden die nächsten Ressourcen
    der Listen genommen.
4. Belegung der gefundenen Ressourcen entsprechend der
    berechneten Dauer

```

Bild 2: Pseudocode zur Aufstellung der Belegungsmatrix für Genmodell 2 (GM2)

Beide Genmodelle unterscheiden sich auch im Umfang der Hybridisierung des EA: GM1 verwendet die Heuristiken nur zur Initialisierung der Startpopulation während GM2 auch die Ressourcenallokation heuristisch durchführt. Die evolutionär bestimmte Ressourcenauswahlstrategie bringt ein Element der Koevolution hinzu. Um dies zu unterstützen, wurde das 1- und n-Punkt-Crossover derart modifiziert, dass die RAS des besseren Elter auf das Kind vererbt wird.

3.3 Genetische und phänotypische Reparatur

Zum Standardumfang von GLEAM gehört eine genotypische Reparatur, die überzählige Gene beim Crossover von Chromosomen fester Länge möglichst so zurück sortiert, dass sie nicht dem Ursprungschromosom zugeordnet werden. Damit wird sichergestellt, dass jedes Chromosom Gene für alle Gridjobs hat. Andererseits können durch positionsver-

ändernde Mutationen Verletzungen der Abarbeitungsreihenfolge der Gridjobs entstehen. Dieses Problem kann man grundsätzlich auf drei Arten behandeln:

1. Die entsprechenden Mutationen werden so verändert, dass sie die Reihenfolgebeziehungen beachten. Davon wurde bisher aus Zeitgründen nicht Gebrauch gemacht.
2. Reparatur des fehlerhaften Chromosoms (genotypische Reparatur)
3. Reparatur durch korrigierende Interpretation (phänotypische Reparatur)

Auf Grund der implementierten Ressourcenbelegung beider Genmodelle (siehe Bilder 1 und 2), ist eine Beachtung der Vorgängerbeziehungen sicher gestellt, wenn die Gene in der richtigen Reihenfolge sind. D.h. im Genom darf kein Gen eines Gridjobs vor einem stehen, dessen Nachfolger es ist. Damit ist eine einfache genotypische Reparatur möglich. Der Nachteil dieser Vorgehensweise besteht in der Störung des Umbaus von Chromosomen, in dem zunächst illegale Zwischenschritte quasi „wegrepariert“ werden. Stattdessen muss z.B. ein zeitliches Vorziehen einer Gridjobsequenz so erfolgen, dass zuerst die jeweils früheren Gridjobs vorgezogen werden, um Platz für die nachfolgenden zu schaffen. Es ist offensichtlich, dass so eine Vorverlegung ganzer Gridjobsequenzen schwierig zu bewältigen ist. Experimente haben dementsprechend auch gezeigt, dass nur etwa 20 % aller Nachkommen genotypisch repariert werden sollten. Zur Aussortierung illegaler Schedules ist eine weitere Straffunktion für die Bewertung notwendig, die „voreilig ausgeführte“ Gridjobs abstraf.

Die phänotypische Reparatur stellt dagegen einen ganz anderen Ansatz dar, der das Genom unverändert lässt. Wenn im ersten Schritt von Bild 1 oder 2 festgestellt wird, dass noch nicht alle Vorgängergridjobs verplant sind, wird der betreffende Gridjob zurückgestellt, bis dies der Fall ist. Es wird also so getan, als stünde sein Gen direkt hinter dem Gen des letzten Vorgängergridjobs im Chromosom. Damit sind sinnvolle Veränderungen im Genom auch über verschlechternde oder gar unzulässige Zwischenschritte leicht möglich.

4 Experimente

Für die Experimente wurden die im nächsten Abschnitt besprochenen sechzehn Benchmarkszenarien entworfen. Die Kosten- und Zeitgrenzen wurden gezielt so eingestellt, dass die heuristische Planung nicht in der Lage war, eine Planung ohne Verletzung dieser Grenzen zu erzeugen. Die durch die Experimente zu beantwortende Fragestellung lautet dementsprechend: Kann GLEAM innerhalb des dreiminütigen Zeitfensters Schedules ohne Grenzverletzungen generieren und haben die erzeugten Planungen eine bessere Qualität. Zur Beantwortung der letzten Frage werden die Fitnesswerte ohne die Straffunktionen für Grenzverletzungen verglichen. Alle Experimente beruhen auf 100 Läufen pro Benchmark und Einstellung des EA wie Reparaturart, Genmodell oder Populationsgröße (200 - 600).

4.1 Benchmarks

Grundsätzlich können Benchmarks zum Testen von Schedulingalgorithmen auf zwei Arten erzeugt werden: Entweder man extrahiert sie aus realen Anwendungsdaten oder sie werden synthetisch erzeugt. Die Praxisnähe scheint zunächst für die erste Erzeugungsart zu sprechen. Allerdings haben Untersuchungen gezeigt, dass sie häufig nur diejenigen Anwendungsszenarien widerspiegeln, die bei der Erfassung gerade dominant waren oder typisch für die jeweilige Institution oder Nutzergruppe sind [10]. Außerdem

können charakteristische Eigenschaften, wie die nachstehend beschriebenen, nicht gezielt beeinflusst werden. Daher verwenden wir synthetische Benchmarks.

Beispiele für synthetisch erzeugte Benchmarks liefern Tobita et al. [13] und Hönig et al. [14], wobei die dort vorgestellten Benchmarks von homogenen Ressourcen ausgehen und einen Anwendungsjob, d.h. einen Workflow beschreiben und analysieren. Da GORBA für heterogene Ressourcen ausgelegt ist und von einer Vielzahl von Anwendungsjobs ausgegangen wird, wurden eigene nachfolgend beschriebene Benchmarks entwickelt.

Wichtige Kennziffern zur Charakterisierung einer zu planenden Menge von Anwendungsjobs betreffen die Abhängigkeiten ihrer Gridjobs und die Anzahl jeweils alternativer Ressourcen [11]. Der Parameter D beschreibt die Abhängigkeiten (dependencies) wie folgt:

$$D = \frac{spj}{spj_{max}} \quad spj_{max} = \frac{n(n-1)}{2} \quad \text{mit: } spj: \text{ Summe der Vorgängerjobs aller Gridjobs} \\ n: \text{ Anzahl aller Gridjobs}$$

Je kleiner D ist, desto größer ist die Anzahl der Permutationen und damit der Planungsalternativen aller Gridjobs. Bild 3 vergleicht die Situation bei drei Gridjobs.

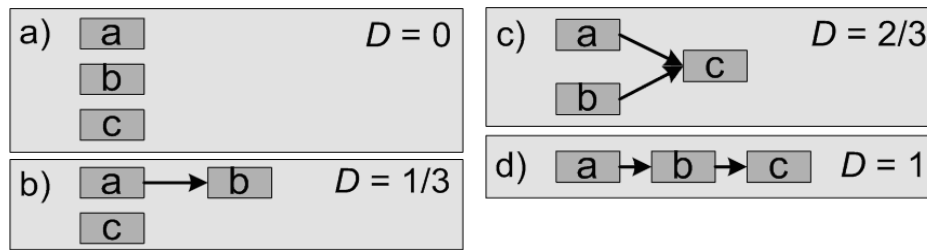


Bild 3: Mögliche Werte für D bei drei Gridjobs

Kleine D -Werte korrespondieren also mit einem hohen potentiellen Parallelisierungsgrad, der allerdings durch den Freiheitsgrad bei der Ressourcenauswahl beschränkt werden kann. Dieser Freiheitsgrad R ist definiert als

$$R = \frac{\sum ar}{n \cdot n_{res}} \quad \text{mit } ar: \text{ Anzahl alternativer Ressourcen eines GridJobs} \\ n: \text{ Anzahl aller Gridjobs} \\ n_{res}: \text{ Anzahl der Ressourcen}$$

Je kleiner R ist, desto geringer ist der Freiheitsgrad der Ressourcenauswahl pro Gridjob. Ein Benchmark mit kleinem D und großem R erlaubt damit sowohl einen hohen Parallelisierungsgrad als auch viele Planungsalternativen. Im Gegensatz dazu bedeuten ein kleines R und ein großes D wenig Parallelisierungsmöglichkeiten bei geringen Ressourcenalternativen. Insgesamt wurden aus den beiden Kennziffern vier Benchmarkklassen gebildet:

- sRsD: kleiner Ressourcenfreiheitsgrad, geringe Abhängigkeiten (small R, small D)
- sRlD: kleiner Ressourcenfreiheitsgrad, viele Abhängigkeiten (small R, large D)
- lRsD: großer Ressourcenfreiheitsgrad, geringe Abhängigkeiten (large R, small D)
- lRlD: großer Ressourcenfreiheitsgrad, viele Abhängigkeiten (large R, large D)

Eine weitere Kennziffer zur Beschreibung der Komplexität einer Planungsaufgabe ist die Anzahl der Gridjobs. Pro Benchmarkklasse wurden Benchmarks mit 50, 100, 200 konstruiert, die alle die gleiche Ressourcenmenge verwenden. Hinzu kommt noch ein

Satz Benchmarks mit 200 Gridjobs bei doppelter Ressourcenmenge (mit 200d in den Bildern 5 - 8 gekennzeichnet).

Bei den Benchmarks handelt es sich um komplette Neuplanungen, wie sie in der Praxis eher selten vorkommen, denn dort dominieren Umplanungen, die eine Anpassung des bisherigen Schedules erfordern. Frühere Experimente mit Umplanungen bei Roboterbahnplanungsaufgaben haben gezeigt, dass diese wesentlich weniger Laufzeit benötigen als komplette Neuplanungen, wenn die alten Ergebnisse bei der Bildung der Startpopulation mit verwendet werden. [12]. Die hier vorgestellten Neuplanungen dienen einer ersten Evaluierung des Konzepts. Die Erweiterung von GORBA auf Umplanungen unter Verwendung des alten Schedules befindet sich zur Zeit der Niederschrift in der Implementierung.

4.2 Ergebnisse

Bild 4 vergleicht die Ergebnisse einer heuristischen Planung mit Bevorzugung schneller Ressourcen und einer nachträglichen Optimierung mit GLEAM am Beispiel des Benchmarks sRID-100 in Form von Gantt-Charts. Einige Ressourcen stehen nur begrenzt zur Verfügung, wie die mit einem Kreuz versehenen schwarzen Balken zeigen. Man erkennt deutlich, dass die Gridjobs im von GLEAM erzeugten Schedule wesentlich dichter gepackt sind.

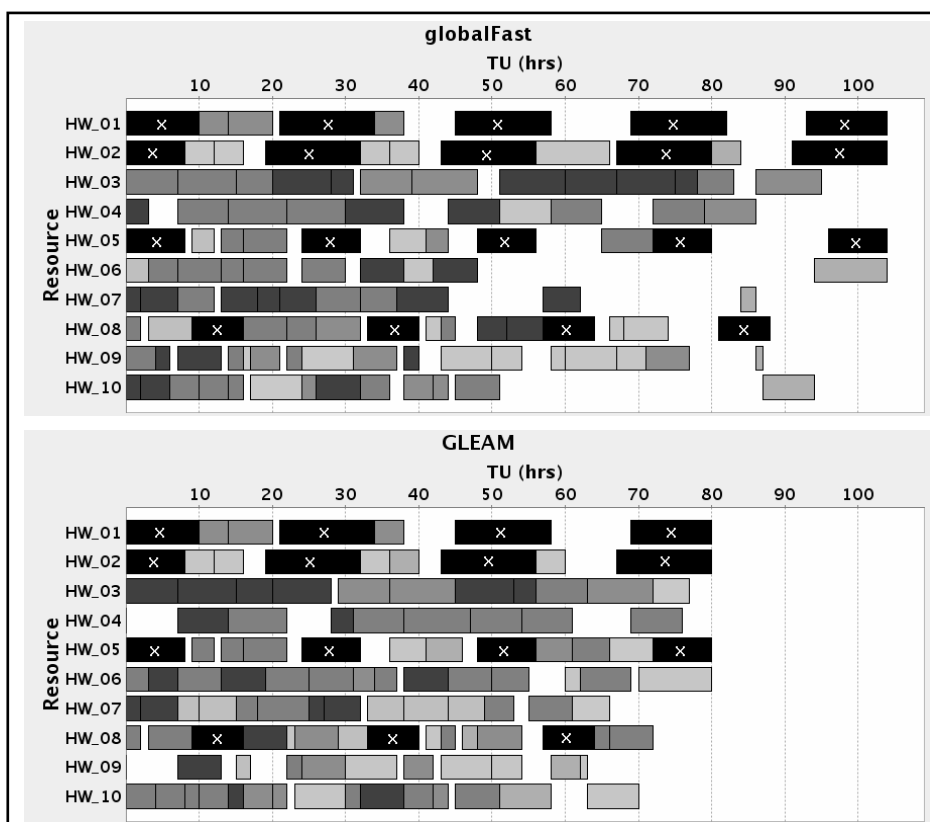


Bild 4: Vergleich der Ergebnisse einer heuristischen Planung mit der Ressourcenauswahlstrategie „schnell“ (oben) mit denen eines GLEAM-Laufs (unten) an Hand der Benchmarkaufgabe sRID-100. Gesperrte Zeiten sind schwarz mit einem **x** markiert. Gridjobs mit gleichem Grauton gehören zum selben Anwendungsjob. TU= Time Units

Die Bilder 5 und 6 geben eine Antwort auf die Frage, ob und wenn ja wie zuverlässig GLEAM in der Lage ist, Schedules zu erzeugen, die die Kosten- und Zeitvorgaben einhalten. Die Erfolgsrate (*success rate* in den Bildern) gibt an, welcher Anteil der GLEAM-Läufe in diesem Sinne erfolgreich war. Bild 5 vergleicht die Ergebnisse für beide Genmodelle und genotypischer Reparatur. Man erkennt deutlich, dass Genmodell 2 besser abschneidet.

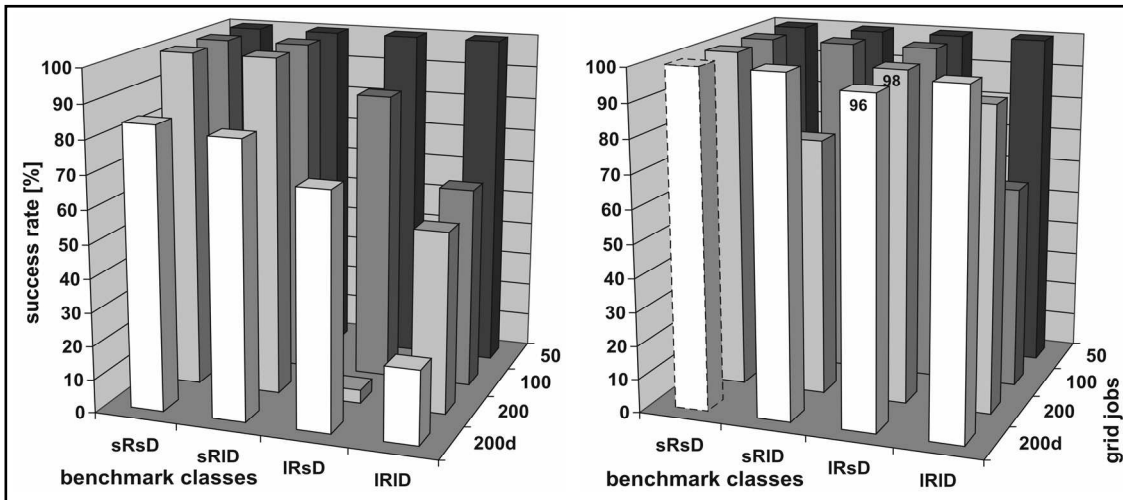


Bild 5: Ergebnisse für Genmodell 1 (links) und Genmodell 2 (rechts) bei **genotypischer Reparatur**. Balken mit gestrichelten Linien erreichen den Erfolgswert nur bei einer der getesteten Populationsgrößen, sodass dieser Erfolg als wenig robust betrachtet werden muss. Zur besseren Unterscheidung ist die Erfolgsrate bei Balken vermerkt, die fast die 100% erreichen.

Bild 6 enthält die gleiche Gegenüberstellung auf der Basis der phänotypischen Reparatur. Auch hier schneidet Genmodell 2 signifikant besser ab.

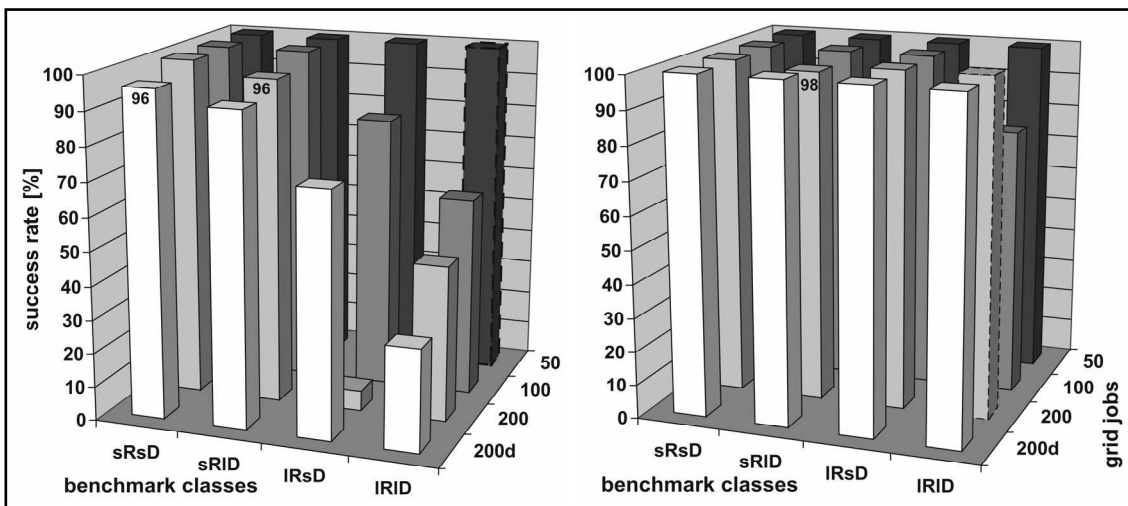


Bild 6: Ergebnisse für Genmodell 1 (links) und Genmodell 2 (rechts) bei **phänotypischer Reparatur**. Weitere Erläuterungen siehe Bild 5

Interessant ist noch der Vergleich zwischen den beiden Reparaturmethoden (jeweils die linke und rechte Hälfte der Bilder 5 und 6). Bei beiden Genmodellen erweist sich die phänotypische Reparatur als überlegen. Somit kann zusammenfassend festgestellt werden, dass GM 2 und phänotypische Reparatur die besten Ergebnisse liefern, wenn man die Erfolgsrate hinsichtlich der Einhaltung von Kosten- und Zeitbudgets zu Grunde legt.

Auf Grund weiterer Untersuchungen kann vom in Bild 7 dargestellten Unterschied der Verläufe einer Optimierung mit beiden Genmodellen ausgegangen werden. GM2 erreicht nach einer gewissen Laufzeit das mit den Ressourcenauswahlstrategien mögliche Suboptimum und stagniert dann. Dies ist bei ausreichend großer Gridjobanzahl (> 50) deutlich nach der drei Minutengrenze der Fall. GM1 braucht mehr Zeit zur Erzeugung guter Schedules und übertrifft GM2 ab einer gewissen Laufzeit in der Regel. Auch hier gilt, dass dies bei ausreichend großer Gridjobanzahl erst nach den drei Minuten erfolgt.

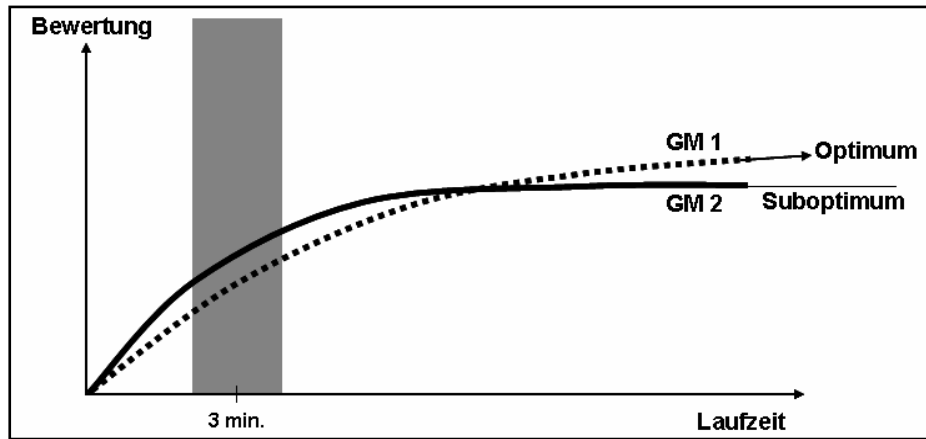


Bild 7: Prinzipieller Verlauf einer Optimierung mit den beiden Genmodellen GM1 und GM2 bei hinreichend vielen Gridjobs (> 50)

Die zweite Frage zur Bewertung der GLEAM-Optimierung betrifft die Qualität der erzeugten Schedules im Vergleich zu den heuristisch geplanten bei Weglassung der Budgetverletzungen. Bild 8 vergleicht die Fitnesswerte beider Genmodelle bei phänotypischer Reparatur auf der Basis der jeweils besten Läufe. Zunächst fällt auf, dass GM1 auch Verschlechterungen produzieren kann (negative Fitnessunterschiede im linken Teil von Bild 8), was bei GM2 dagegen nicht vorkommt. Die Verschlechterungen erklären sich damit, dass unbestrafte Fitnesswerte verglichen werden und der EA mit höchster Priorität versucht, straffreie Schedules zu erzeugen. Erst danach beginnt die schrittweise

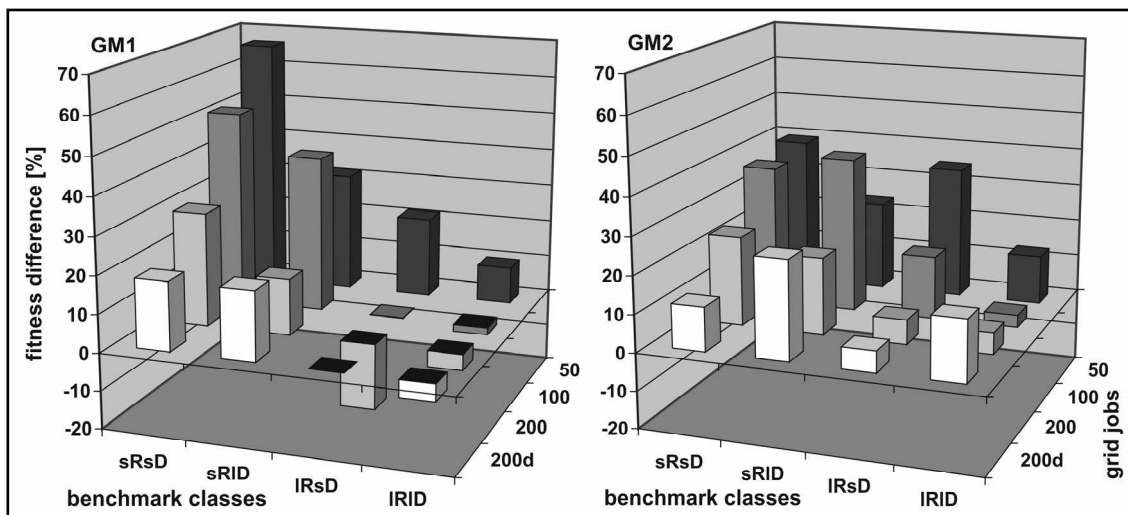


Bild 8: Fitnessunterschiede zwischen der besten heuristischen und der besten GLEAM-Planung bei phänotypischer Reparatur und ohne Berücksichtigung der Zeit- und Kostenbudgets, d.h. ohne Anwendung der entsprechenden Straffunktionen

Verbesserung der gesamten Qualität des Schedules. Wird dieser Prozess nach drei Minuten abgebrochen, können insgesamt schlechtere aber straffreie oder weniger bestrafte Schedules dabei herauskommen. Ein Vergleich zwischen den Bildern 7 und 8 für GM1 zeigt auch, dass die größten Verschlechterungen dann auftreten, wenn auch die Erfolgsrate vergleichsweise gering ausfällt.

Bei den Benchmarks mit geringer Gridjobanzahl und geringem Freiheitsgrad bei der Ressourcenauswahl erreicht GM1 zum Teil bessere Ergebnisse als GM2. Dies ist darauf zurückzuführen, dass in diesen Fällen die Laufzeitgrenze nur eine untergeordnete Rolle spielt und der Zeitrahmen ausreicht, um optimale oder fast optimale Lösungen zu produzieren. Mit anderen Worten, in diesen Fällen liegt der Schnittpunkt von GM1 und GM2 aus Bild 7 vor dem grauen 3-Minutenbalken.

Die Ergebnisse sind vor dem Hintergrund der in den drei Minuten möglichen Evaluationen zu sehen. Bild 9 zeigt für GM2, dass diese Zahl stark von der Anzahl der Gridjobs und der alternativ verfügbaren Ressourcen abhängt. Es ist nicht verwunderlich, dass die Erstellung einer Belegungsmatrix mit zunehmender Jobzahl länger dauert. Dies gilt aber auch für die Ressourcenauswahl, wenn die Anzahl der Alternativen pro Gridjob steigt (Vergleich der sR-Benchmarkklassen mit den IR-Klassen). Die Lage entspannt sich dagegen etwas, wenn die Anzahl der Ressourcen insgesamt im Verhältnis zu den Gridjobs steigt, wie die etwas höhere Anzahl an Evaluationen bei der 200d-Benchmarkserie gegenüber der 200 Job-Benchmarkserie zeigt.

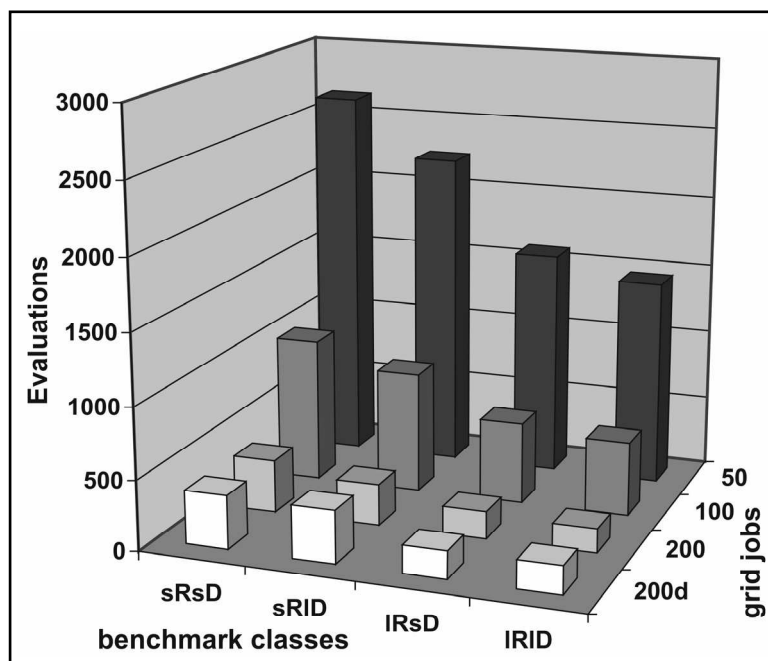


Bild 9: Durchschnittliche Anzahl der durchgeführten Evaluation innerhalb der Laufzeit von drei Minuten bei GM2 und phänotypischer Reparatur

5 Zusammenfassung und Ausblick

Das Problem der Planung von Gridjobs mit Vorgängerbeziehungen und Koallokation heterogener Ressourcen unterschiedlicher Leistungsfähigkeit bei variablen Kosten über die Zeit sowie einer Bewertung nach Kosten, Fertigstellungszeiten und Ressourcenaus-

lastung wurde mit dem reinen Jobshop-Scheduling Problem verglichen. Es ist auf Grund seiner dem *Resource Constrained Project Scheduling* ähnlichen Struktur und der umfassenderen Bewertung komplexer als das bereits NP-vollständige Grundproblem der Maschinenbelegung. An Hand von hier vorgestellten Kennziffern wurden unterschiedlich strukturierte Benchmark-Szenarien entwickelt, die zur Performanceanalyse verwendet wurden. Grundlage dafür waren Zeit- und Kostenvorgaben, die von der heuristischen ersten Planungsphase nicht eingehalten werden konnten. Es konnte gezeigt werden, dass innerhalb eines praxistauglichen Zeitfensters von drei Minuten die Nichteinhaltung dieser Vorgaben durch eine nachfolgende Verbesserung mit Hilfe des EA GLEAM beseitigt werden konnten. Dabei wurden zwei Genmodelle und zwei Reparaturmechanismen miteinander verglichen. Phänotypische Reparatur und ein Genmodell mit koevolutionär bestimmter Auswahl einer Ressourcenzuordnungsheuristik erwies sich dabei als die beste Wahl.

Die hier vorgestellte Performanceanalyse basiert auf Benchmarks vom Typ *Neuplanung*, was zwar gut für Vergleiche ist aber insofern unrealistisch, als in der Praxis die Umplanung auf Grund von z.B. einem neuen Anwendungsjob oder eines Ressourcenausfalls dominiert. Daher sind derzeit Arbeiten im Gang, Umplanungsszenarien unter Ausnutzung der alten Planung zu untersuchen. Hierbei ist insbesondere die Fragestellung von Interesse, wie viel mehr Ressourcen und Gridjobs innerhalb der vorgegebenen Planungszeit bearbeitet werden können.

Außerdem werden neue und problemangepasste genetische Operatoren [7] untersucht werden. Dazu zählen insbesondere Experimente mit dem aus der Literatur bekannten *order crossover* [15, 16] für kombinatorische Problemstellungen. Als ein weiteres Element der Verbesserung wird derzeit ein zusätzliches Bewertungskriterium untersucht, dass die Wartezeiten der GridJobs eines Anwendungsjobs erfasst, um Zwischenschritte zu honorieren, die noch nicht zu einer Verkürzung der gesamten Bearbeitungszeit eines Anwendungsjobs geführt haben. Erste Experimente sind viel versprechend.

Parallel dazu ist die Erprobung von GORBA im CampusGrid des Forschungszentrums [17] in Vorbereitung.

Literatur

- [1] Stucky, K.-U., Jakob, W., Quinte, A.; Süß, W.: *Solving scheduling problems in Grid resource management using an evolutionary algorithm*. In: Meersman, R. (Hrsg.): *On the Move to Meaningful Internet Systems 2006, Conf. Proc. of OTM Confed. Internat. Conf., LNCS 4276*, Springer, Berlin, S.1252-1262, 2006.
- [2] Brucker, P.: *Scheduling Algorithms*. Springer, Berlin, 1995.
- [3] Brucker, P., Knust, S.: *Complex Scheduling*. Springer, Berlin, 2006
- [4] Blume, C.: *GLEAM - A System for Intuitive Learning*. In: Schwefel, H.P., Männer, R. (eds): *Conf. Proc. PPSN I, LNCS 496*, Springer, Berlin, S.48-54, 1991.
- [5] Blume, C., Jakob, W.: *GLEAM – An Evolutionary Algorithm for Planning and Control Based on Evolution Strategy*. In: Cantú-Paz, E. (ed.): *GECCO 2002, Vol. LBP*, S.31-38, 2002.
- [6] Hovestadt, M., Kao, O., Keller, A., Streit, A.: *Scheduling in HPC Resource Management Systems: Queuing vs. Planning*. *Conf. Proc. of the 9th Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP) at GGF8, LNCS 2862*, Springer, Berlin, S.1-20, 2003.
- [7] Hahnenkamp, B.: *Integration anwendungsneutraler lokaler Suchverfahren in den adaptiven memetischen Algorithmus HyGLEAM für komplexe Reihenfolgeoptimierung*. Dip-

- lomarbeit, Uni Karlsruhe, Fak. f. Maschinenbau, Forschungszentrum Karlsruhe, Inst. f. Angew. Informatik, 2007.
- [8] Schmeck, H., Merkle, D., Middendorf, M.: *Ant Colony Optimization for Resource-Constrained Project Scheduling*. In: Whitley, D., et al. (eds.): Conf. Proc GECCO 2000, Morgan Kaufmann, San Francisco, S.893-900, 2000.
 - [9] Blume, C.: *GLEAM - Ein EA für Prozessabläufe am Beispiel von Steuerungen für Industrieroboter*. In: Mikut, R., Reischl, M. (Hrsg.): Conf. Proc. 16. Workshop Computational Intelligence, Universitätsverlag Karlsruhe, S.11-24, 2006.
 - [10] Wiecezorek, M., Prodan, R., Fahringer, T.: *Comparison of Workflow Scheduling Strategies on the Grid*. In: R. Wyrzykowski et al. (Hrsg.): PPAM 2005, LNCS 3911, Springer, Berlin, S.792-800, 2006.
 - [11] Süß, W., Quinte, A., Jakob, W., Stucky, K.-U: *Construction of Benchmarks for Comparison of Grid Resource Planning Algorithms*. In: Filipe, J. et al. (Hrsg.): Conf. Proc. of 2nd Int. Conf. On Software and Data Technologies (ICSOFTE 2007), Vol. PL/DPS/KE/WsMUSE, Inst. f. Systems and Technologies of Information, Control and Com., S.80-87, 2007.
 - [12] Jakob, W., Gorges-Schleuter, M., Blume, C.: *Application of Genetic Algorithms to Task Planning and Learning*. In: R. Männer, B. Manderick (Hrsg.): Conf. Proc. PPSN II, North-Holland, Amsterdam, S.291-300, 1992.
 - [13] Tobita, T., Kasahara, H.: *A Standard Task Graph Set for Fair Evaluation of Multiprocessor Scheduling Algorithms*. Journal of Scheduling, Vol 5(5), S.379-394, 2002.
 - [14] Hönig, U., Schiffmann, W.: *A comprehensive Test Bench for the Evaluation of Scheduling Heuristics*. In: Proc. of the 16th Int. Conf. on Parallel and Distributed Computing and Systems (PDCS), Cambridge (USA), 2004.
 - [15] Davis, L. (Hrsg.): *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.
 - [16] Eiben, A.E., Smith, J.E.: *Introduction to Evolutionary Computing*. Springer, Berlin, 2003.
 - [17] Schmitz, F., Schneider, O.: *The CampusGrid Test Bed at Forschungszentrum Karlsruhe*. In: Sloot, P.M.A. (Hrsg.): Conf. Proc. of Advances in Grid Computing – EGC 2005, LNCS 3470, Springer, Berlin, S.1139-1142, 2005.

Evolutionäre mehrkriterielle Optimierung bei Anwendungen mit nichtzusammenhängenden Pareto-Mengen

Mike Preuß, Boris Naujoks

Lehrstuhl für Algorithm Engineering
Universität Dortmund, 44221 Dortmund

Tel.: (0231) 755 - 77 05

Fax: (0231) 755 - 77 40

E-Mail: {mike.preuss, boris.naujoks}@uni-dortmund.de

1 Einleitung

Evolutionäre Algorithmen werden seit Mitte der 90er Jahre zunehmend als eins der Standardverfahren für die mehrkriterielle Optimierung etabliert. Sie zeichnen sich vor allem durch die Einfachheit der Anwendung auf mathematisch schwer zu formulierenden Problemen und durch die gleichzeitige Verwaltung mehrerer potentieller Lösungen dank des inhärenten Populationskonzeptes aus. Diese bilden bei erfolgreicher Optimierung eine gute Approximation der wahren Pareto-Front. Mehrere *evolutionäre mehrkriterielle Optimierungsalgorithmen* (EMOA) sind mittlerweile weitverbreitet, so z.B. der NSGA-II [1, 2], SPEA2 [3, 4] oder der SMS-EMOA [5, 6].

Allerdings gehen die genannten Standardverfahren für die mehrkriterielle Optimierung implizit von der Annahme aus, die Pareto-Menge im Suchraum sei zusammenhängend. Das dies nicht immer so ist, und insbesondere für Probleme mit enthaltenen multimodalen Funktionen vermutlich eher die Ausnahme darstellt, wurde kürzlich in [7] an einem einfachen Modellproblem gezeigt. Hier und in der daran anknüpfenden Arbeit [8] ist auch die Eignung der Standardverfahren für Probleme untersucht worden, die nicht zusammenhängende Pareto-Mengen enthalten. Dabei wurden deren Schwierigkeiten offenbar, die verschiedenen Teile der Pareto-Menge zu entdecken und hinreichend gut zu approximieren. Allerdings muss festgestellt werden, dass die getesteten EMOA vermutlich aufgrund der enthaltenen Diversitätserhaltungsmechanismen (die allerdings nur auf den Zielraum angewandt werden) im Vergleich zu vielen einkriteriellen Optimierverfahren immer noch recht gut abschneiden. Trotzdem macht es Sinn, nach Verbesserungsmöglichkeiten zu suchen, um Algorithmen zu finden, die zwar weiterhin eine gute Approximation der Pareto-Front liefern, aber auch mit verteilten Pareto-Mengen gut zurecht kommen.

Ziel dieser Arbeit ist es, das bisher vorhandene Wissen über Vorhandensein von und mögliche algorithmische Reaktion auf verteilte Pareto-Mengen in einer geschlossenen Zusammenschau übersichtlich darzustellen, und mögliche Weiterentwicklungen in diesem Bereich anzustoßen.

2 Vom Modell zur Anwendung

In den folgenden Abschnitten werden zunächst 2 einfache, konstruierte Probleme diskutiert. Diese wurden einerseits dazu entworfen, sich über die Beschaffenheit der Pareto-Menge bei einfachen multimodalen Teilfunktionen Klarheit zu verschaffen. Andererseits

sollten auch die Standardalgorithmen einem ersten Test unterzogen werden, in dem sie, wie erwartet, nicht sehr gut abgeschnitten haben. Darauf folgend zeigen wir an einem Beispiel aus der Praxis, dass die postulierten Effekte tatsächlich in der Realität auftreten können.

2.1 TWO-ON-ONE

Im praktischen Einsatz der einkriteriellen Optimierung findet man sehr häufig multimodale Probleme vor, die weder von herkömmlichen Verfahren wie z.B. Gradientenabstiegs- methoden noch von vielen kanonischen evolutionären Algorithmen gut gelöst werden können. Daher haben sich in der *Evolutionary Computation* (EC) mehrere, zumeist unter dem Stichwort *Niching/Einnischung* firmierende Methoden etabliert. Beispiele für diese sind unter anderem die älteren Verfahren *Crowding* [9] und *Sharing* [10], aber auch neuere wie *Sequential Niching* [11], *Clearing* [12], *UEGO* [13], *Shifting-Balance GA* [14], *Species Conservation* [15], Niching durch Clusterbildung [16] und Einnischung in Evolutionsstrategien [17].

Die gebräuchlichen Testprobleme in der mehrkriteriellen evolutionären Optimierung [18, 19] greifen allerdings fast ausschließlich auf zusammengesetzte Funktionen zurück, für die sowohl die Pareto-Menge als auch die Pareto-Front zusammenhängend sind. Ist zumindest eine der beteiligten Funktionen multimodal, kann das jedoch nicht mehr vorausgesetzt werden. Der in [7] untersuchte Fall eines 2-kriteriellen Problems mit einer multimodalen und einer unimodalen Funktion (TWO-ON-ONE) ist daher der einfachste mögliche Test, um sowohl den Zerfall der Pareto-Menge, als auch die Reaktion von EMOA darauf zu beobachten.

$$f = (f_1, f_2) : \mathbb{R}^2 \rightarrow \mathbb{R}^2 : \begin{aligned} f_1(x_1, x_2) &= x_1^4 + x_2^4 - x_1^2 + x_2^2 - cx_1x_2 + dx_1 + 20, \\ f_2(x_1, x_2) &= (x_1 - k)^2 + (x_2 - l)^2 \end{aligned} \quad (1)$$

Die beiden Zielfunktionen sind in Gleichung (1) angegeben, durch Verschieben der Kugelfunktion f_2 und Anheben eines Optimums von f_1 wurden insgesamt 5 verschiedene Fälle definiert, die in Abbildung 1 dargestellt sind. Die zugehörigen Parameterwerte für d , k und l finden sich in Tabelle 1.

Tabelle 1: Parameterwerte für die fünf Fälle von TWO-ON-ONE, c immer gleich 10.

Cases		1	2	3	4	5
Parameters	d	0	0	0.25	0.25	0.25
	k	0	1	0	1	0
	l	0	0	0	0	1

Betrachtet man die durch Gitter- bzw. Zufallstests ermittelten Pareto-Mengen (Abbildung 2), fällt sofort auf, dass es sich scheinbar um flächige Strukturen handelt, obwohl man eher Linien erwartet hätte. Die wahre Pareto-Menge lässt sich auch analytisch bestimmen und ähnelt tatsächlich einer Linie mit leicht gekrümmten Enden. Allerdings muss damit gerechnet werden, dass ein populationsbasierter EMOA die Pareto-Menge während der Suche immer als verwischte Fläche „sieht“, was vermutlich dadurch begründet ist, dass

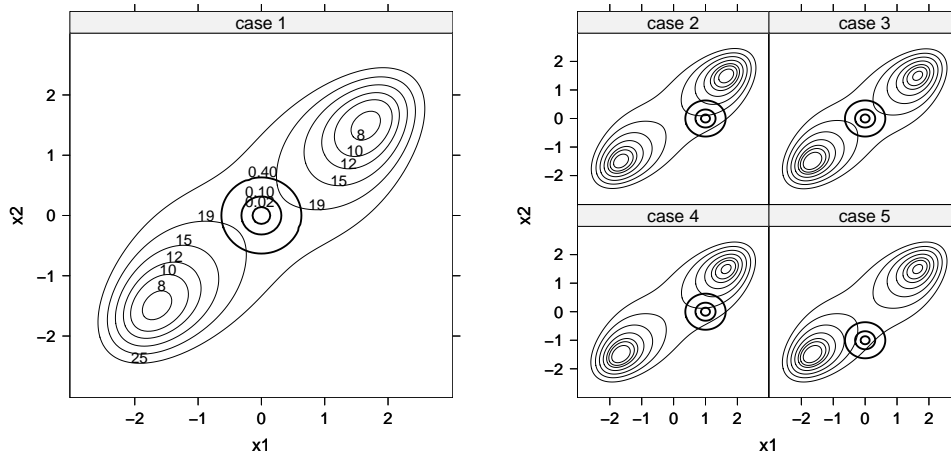


Bild 1: Fälle 1 (links) und 2-5 (rechts) der Testfunktion TWO-ON-ONE, überlagernd gezeichnet. Die beiden Optima von f_1 sind in den Fällen 1 und 2 gleich gut, für 3-5 ist das untere etwas besser.

hier für einen kleinen Schritt im Entscheidungsraum die Unterschiede in der einen Zielfunktion viel größer sind als in der anderen.

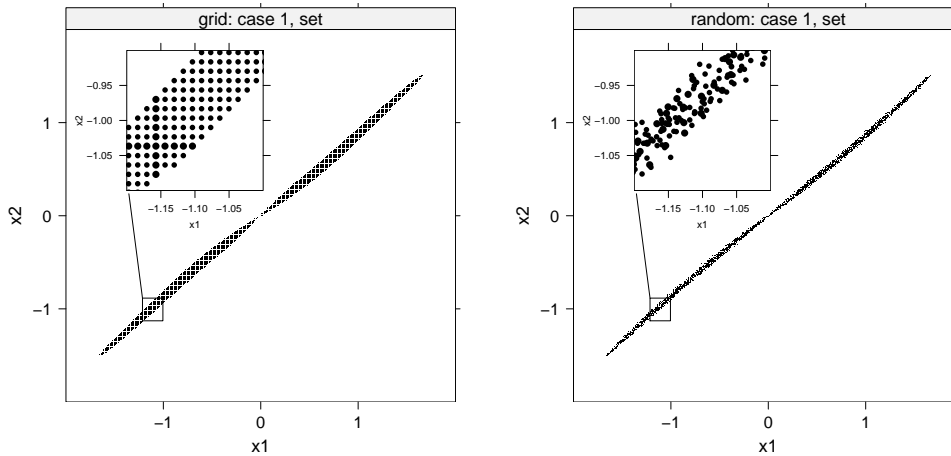


Bild 2: Durch Gittertest (links) bzw. Zufallstest (rechts) empirisch bestimmte Pareto-Mengen von TWO-ON-ONE, Fall 1.

Alle getesteten EMOA (NSGA-II, SPEA2 und SMS-EMOA) verhielten sich auf dem TWO-ON-ONE Problem im Groben ähnlich. Sind die beiden Hälften der Pareto-Menge gleichwertig (symmetrischer Fall), besteht die Gefahr, dass sich die Population durch den Drifteffekt (die beiden Bereiche können ja anhand der Zielfunktionswerte nicht unterschieden werden) vollständig auf eine Hälfte konzentriert. Tritt dies nicht ein, so kann man doch eine gewisse Häufung von Individuen in einzelnen Zonen der beiden Bereiche ausmachen. In jedem Fall können die Algorithmen keine gleichmäßige Abdeckung beider Hälften liefern. Dies verwundert jedoch nicht, da sie für eine solche Aufgabe auch nicht entworfen wurden. Es zeigt jedoch, dass hier Bedarf besteht, Mechanismen einzubauen, die es ermöglichen, mehrere äquivalente Teile der Pareto-Menge gleichzeitig im Fokus der Population zu behalten. Dies ist auch und vor allem dann wichtig, wenn die beiden Seiten der Doppelkugelfunktion asymmetrisch sind, da dann für eine volle Abdeckung der Pareto-Front tatsächlich oft Individuen auf beiden Seiten platziert werden müssen (z.B. bei Fall 4). Gelingt das nicht, wird nur ein Teil der Pareto-Front gefunden.

2.2 SYM-PART

Im Abschnitt zuvor wurde festgestellt, dass bereits in einfachen Fällen Situationen entstehen können, in denen verschiedene Gebiete des Suchraums auf dieselbe Pareto-Front abgebildet werden. Die Veröffentlichung [8] untersucht das Verhalten verschiedener EMOA in derartigen Situationen. Dazu wurde wiederum exemplarisch eine Testfunktion definiert, welche dann mit verschiedenen EMOA behandelt wurde.

Die Abbildung 3 stellt mehrere Möglichkeiten dar, wie Pareto-Mengen auf Pareto-Fronten abgebildet werden können. Die üblichen Testfunktionen, die bei der mehrkriteriellen evolutionären Optimierung Anwendung finden, sind vom Typ I oder II. Der Typ III ist aus industrieller Sicht interessant, da Lösungen der Pareto-Front verschiedene Urbilder aufweisen. Sind diese alle bekannt, kann diejenige Lösung für die Produktion umgesetzt werden, die am wenigsten Aufwand erfordert und für den Ingenieur am einfachsten umzusetzen ist.

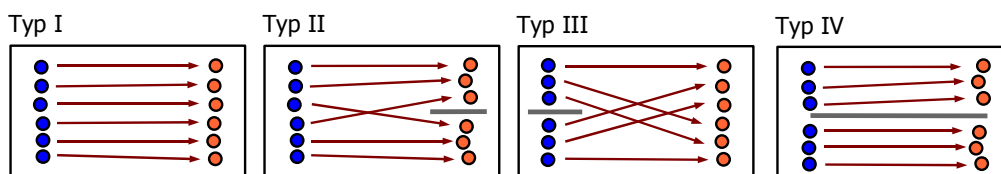


Bild 3: Verschiedene Möglichkeiten, wie Pareto-Mengen auf Pareto-Fronten abgebildet werden können: Eine zusammenhängende Pareto-Menge wird auf eine zusammenhängende Pareto-Front abgebildet (Typ I), eine Pareto-Menge auf eine nicht-zusammenhängende Pareto-Front (Typ II), eine nicht-zusammenhängende Pareto-Menge auf eine zusammenhängende Pareto-Front (Typ III) und nicht-zusammenhängende Pareto-Mengen auf nicht-zusammenhängende Pareto-Front (Typ IV). Probleme des Typs III sind bisher kaum untersucht worden, obwohl ihre Eigenschaft multiple Urbilder für Lösungen der Pareto-Front bereit zu stellen insbesondere im industriellen Umfeld sehr interessant ist.

Für die bereits erwähnte Zielfunktion zerlegten wir den zwei-dimensionalen Suchraum in 3×3 Kacheln und definierten eine Funktion abschnittsweise auf diesen. In Mitten jeder Kachel gibt es eine Gerade, die immer wieder auf denselben Zielfunktionswert abgebildet wird. So ergibt sich eine nicht-zusammenhängende Pareto-Menge aus 9 Abschnitten, die alle auf eine Pareto-Front abgebildet werden.

In [8] wurden zuerst verschiedene Erweiterungen dieser grundlegenden Funktion untersucht. Einflussfaktoren dabei waren

- die Größe des Suchraums außerhalb der Kacheln,
- die Verschiebung aller Kacheln im Suchraum,
- eine Rotation der Kacheln um 45° und
- eine Transformation.

Zudem wurden zwei Maße definiert, die Aufschluss über die Güte der erzielten Ergebnisse geben. Das erste Maß $cs(\mathbf{P}, \mathbf{S})$ (*covered sets*) misst die Anzahl der von einer Population P abgedeckten Teilmengen einer Pareto-Menge S . Dazu wird für jedes Individuum von P überprüft, ob es nahe einer Teilmenge von S liegt und die Teilmenge, für die es derartige Individuen gibt werden durch $cs(\mathbf{P}, \mathbf{S})$ gezählt.

Das zweite Maß $\text{sps}(P, S)$ (*set population spread*) misst die Verteilung der Individuen der Population P über die Teilmengen der Pareto-Menge S . Dazu wird die Standardabweichung der Anzahl der Individuen pro Teilmenge berechnet.

Mit dem bereits bekannten NSGA-II wurde ein *full factorial design* (vgl. beispielsweise Fisher [20]) abgearbeitet mit 30 Wiederholungen pro Einstellung. Anschließend wurden die Effekte 1. Ordnung (Haupt-Effekte) und 2. Ordnung (Interaktionen) untersucht. Haupt-Effekte waren die Größe des Suchraums und die Transformation des Suchraums. Interaktionen waren die Kombinationen Transformation und Suchraumgröße sowie Rotation und Transformation. Auf Basis dieser Ergebnisse wurde entschieden für alle weiteren Untersuchungen den schwierigeren Fall der Suchraumgröße zu betrachten. Ferner sollten für die Untersuchung der Leistungen verschiedener EMOA drei Fälle untersucht werden:

1. Der einfache Fall ohne Rotation und Transformation
2. Der Fall mit Rotation, aber ohne Transformation
3. Der schwierigste Fall mit Rotation und mit Transformation.

Die Abbildung 4 zeigt die Pareto-Mengen zu diesen drei Fällen.

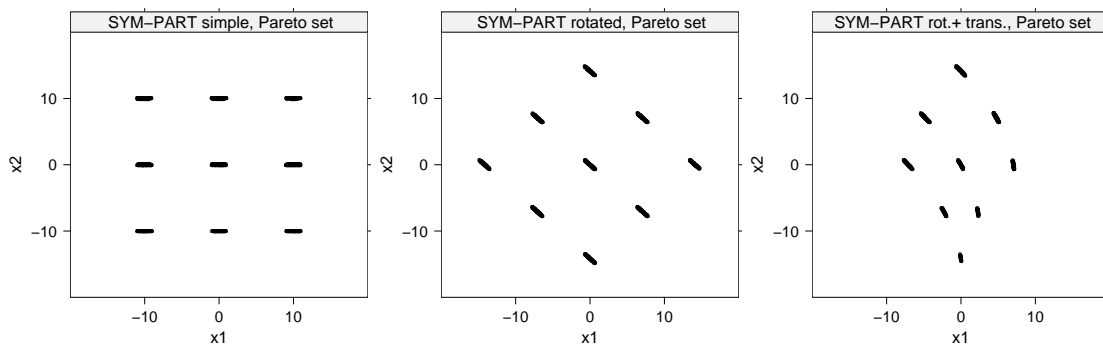


Bild 4: Pareto-Mengen der drei ausgewählten Ausprägungen der SYM-PART Funktion. Die Graphik links zeigt die einfache Variante, die Graphik in der Mitte die gedrehte und die Graphik rechts die gedrehte und transformierte Variante.

Zur Untersuchung des Verhaltens allgemeiner EMO-Algorithmen wurden wiederum NSGA-II und SPEA2 heran gezogen. Zusätzlich wurde ein Algorithmus von Chan und Ray (genannt KP1) [21] verwendet, der die Verteilung der Population bereits im Algorithmus berücksichtigt. Dazu werden in jeder Generation zwei Maße für die Diversität der Population berechnet und angewendet. Ein Ansatz über die Größen der Nachbarschaft misst die Diversität im Suchraum, das dominierte Hypervolumen die Diversität im Lösungsraum. Beide Werte werden für die Selektion in KP1 verwendet [21].

Für eine ausführliche Untersuchung der Algorithmen auf den drei Probleminstanzen wurden Standardwerte für die Parameter der Verfahren gewählt. Als Variationsoperatoren wurden SBX ($\eta_c = 15$) und polynomielle Mutation ($\eta_m = 20$) mit einer Wahrscheinlichkeit von 1 eingesetzt. Das (100+100)-Selektionsschema lief über 300 Generationen, was in eine Anzahl von 30.000 Zielfunktionsauswertungen resultierte. Erste Ergebnisse mit 10.000 Zielfunktionsauswertungen deuteten bereits an, dass mit diesen Parametern ein hoher Wert für $cs(P, S)$ erzielt werden kann. Andere Mutationsoperatoren führten im Vergleich dazu zu wesentlich geringeren $cs(P, mS)$ Werten. Diese Beobachtung wurde aber nicht weiter untersucht.

Die Tabellen 2 und 3 geben die durchschnittlichen Ergebnisse der Algorithmen aus jeweils 30 Wiederholungen und die entsprechenden Standardabweichungen in Klammern wieder. Die Tabelle 2 enthält die erzielten Hypervolumenwerte der resultierenden Pareto-Fronten. Hier sind für die drei Algorithmen auf allen drei Probleminstanzen nur sehr geringe Unterschiede feststellbar. Dies ist auch einleuchtend, da lediglich die Verteilung der Lösungen im Lösungsraum betrachtet wird. NSGA-II und SPEA2 schneidet geringfügig besser ab. Dies liegt an der alleinigen Ausrichtung der Selektion auf die Verteilung in eben diesem Lösungsraum. KP1 zieht gleichzeitig die Verteilung im Suchraum mit in Betracht und erzielt daher nicht ganz so gute Lösungsmengen.

Tabelle 2: Test verschiedener mehrkriterieller evolutionärer Algorithmen auf den ausgewählten Instanzen des SYM-PART Problems. Die dargestellten Werte geben das durchschnittliche dominierte Hypervolumen nach 30 Läufen mit jeweils 30.000 Zielfunktionsauswertungen an (die zugehörigen Standardabweichungen stehen in Klammern).

algorithm	simple	rotated	rot.+trans.
NSGA-II	22.254 (0.00353)	22.255 (0.00305)	22.254 (0.00358)
SPEA2	22.257 (0.00237)	22.257 (0.00243)	22.255 (0.00278)
KP1	22.241 (0.00712)	22.231 (0.00689)	22.220 (0.00781)

Tabelle 3 enthält mit den erzielten $cs(P, S)$ Werten ein Maß für die Diversität der Lösungen im Suchraum. Hier schneidet KP1 wesentlich besser ab als die getesteten Standardverfahren. Für alle Probleminstanzen werden wesentlich mehr Kacheln entdeckt. Betrachtet man dazu die Verläufe der $cs(P, S)$ Werte geplottet über die Generationen (vgl. Abb. 5), stellt man allerdings fest, dass „entdeckt“ nicht der richtige Ausdruck ist. Vielmehr müsste von „behalten“ Gebrauch gemacht werden.

Tabelle 3: Test verschiedener mehrkriterieller evolutionärer Algorithmen auf den ausgewählten Instanzen des SYM-PART Problems. Die dargestellten Werte geben die $cs(P, S)$ nach 30 Läufen mit jeweils 30.000 Zielfunktionsauswertungen an (die zugehörigen Standardabweichungen stehen in Klammern).

algorithm	simple	rotated	rot.+trans.
NSGA-II	6.333 (1.446)	5.633 (1.450)	4.667 (1.124)
SPEA2	6.3 (1.022)	5.2 (1.157)	5 (1.364)
KP1	8.3 (1.290)	6.733 (1.818)	6.5 (0.9738)

Abbildung 5 zeigt durchschnittlichen $cs(P, S)$ Werte pro Generation gemittelt über 30 Läufe für alle 300 Generationen. Die Graphik für SPEA2 ist nahezu identisch mit der Graphik für den NSGA-II. Das Verhalten der Algorithmen ist nahezu identisch (vgl. [8]). Am Anfang sind alle Kacheln mit Lösungen in der Nähe der Pareto-Mengen besetzt. Dies ist auf den kleinen Suchraum und die zufällige Verteilung der Startlösungen zurück zu führen. Im Verlauf des Algorithmus verlieren NSGA-II wie auch SPEA2 immer weiter Kacheln. Nach 30.000 Zielfunktionsauswertungen bleiben im Schnitt nur noch fünf bis sechs Kacheln besetzt. Im Gegensatz dazu erzielt das Hypervolumen aber bereits nach 2.000 Zielfunktionswerten recht hohe Werte über 22.

Der KP1 zeigt im Gegensatz zu den Standardverfahren ein bemerkenswertes Verhalten. Zwar verliert auch er zu Beginn der Optimierung zuerst Kacheln, nach ca. 2.000 Auswertungen ändert sich aber das Verhalten des Algorithmus und es gelingt ihm, die Anzahl

der besetzten Kacheln wieder zu vergrößern (vgl. Abb. 5). Der Verlauf des Hypervolumens zeigt im Vergleich dazu keine Auffälligkeiten. Das Verhalten gleicht dem der Standardverfahren nahezu. So konnte ein signifikant anderes Verhalten für KP1 im Vergleich zu NSGA-II und SPEA2 nachgewiesen werden. Dieser Algorithmus ist in der Lage wesentlich mehr Kacheln zu halten als Standardverfahren mit vergleichbaren Parametrisierungen.

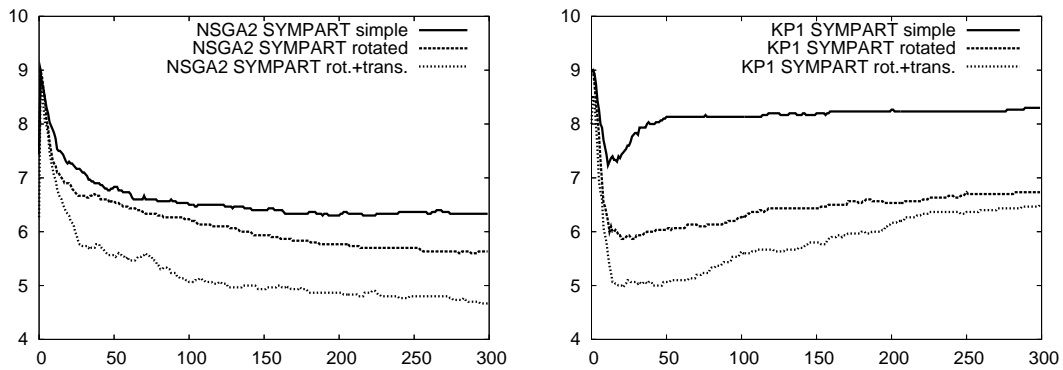


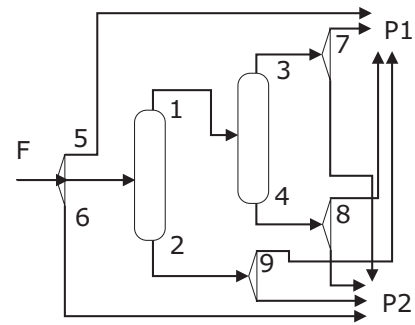
Bild 5: Vergleich von NSGA-II (links) und KP1 (rechts) mit Hilfe des $cs(P, S)$ Maßes. Dargestellt sind die Mittelwerte pro Generation aus 30 Läufen der jeweiligen Algorithmen. Die Populationsgröße betrug 100 Individuen und es wurden 300 Generationen berechnet.

Das Ziel wirklich alle Teilmengen der Pareto-Menge zuverlässig zu entdecken erzielte ein einkriterieller Multistart Ansatz. Die Idee dabei ist zuerst eine Reihe von einkriteriellen evolutionären Algorithmen für jedes Zielkriterium zu starten. Durch die zufällige Initialisierung gelingt es so verschiedene Bereiche des Suchraum zu explorieren und somit nicht-dominierte Lösungen in den jeweiligen Bereichen zu entdecken. Diese dienen dann als Startpunkt für folgende monokriterielle Optimierungen mit einem gewichteten Tchebycheff-Ansatz (vgl. Steuer [22]). Dabei ändert sich die Wahl der Gewichte in jedem neuen Lauf und die Endpunkte des Laufs zuvor werden als Startpunkte für die Läufe mit den neuen Gewichten verwendet. So gelingt es immer neue Pareto-optimale Punkte in einer Teilmenge der Pareto-Front zu ermitteln. Da dieser Ansatz für jeden Startpunkt auf der Phase zuvor durchgeführt wird, werden so für alle Startpunkte die entsprechenden Pareto-Teilmengen durch Lösungen approximiert. Die Anzahl der zu erwartenden Lösungen pro Teilmenge entspricht der Anzahl Läufe und Kombinationen der Gewichte im gewichteten Tchebycheff-Ansatz. In [8] wurden 10 Lösungen gewählt. Kombiniert mit den weiteren Parametereinstellungen aus [8] können für SYM-PART zuverlässig alle 9 Teilmengen mit weniger als 20.000 Zielfunktionsauswertungen entdeckt werden.

2.3 Ein praktisches Beispiel: Auslegung von Destillationsanlagen

Auch wenn die in Abbildung 3 vorgestellte Klassifizierung von mehrkriteriellen Problemen eine Idealisierung darstellt — schließlich wird die Unterscheidung zwischen Typ III und Typ IV durch gleiche bzw. verschiedene Zielfunktionswerte (mehrere Fronten) in den betreffenden Optima hervorgerufen — so ist sie doch nützlich, um zu erkennen, welcherart Probleme bisher kaum untersucht wurden. Da wir postuliert haben, dass viele praktische Probleme aufgrund der enthaltenen Multimodalitäten in den Teilfunktionen den Typen III und IV zuzuordnen sind, scheint es geboten, ein wirklichkeitsnahes Beispiel dafür zu präsentieren.

Bild 6: Schaltbild der zu optimierenden Destillationsanlage, die Ziffern bezeichnen die einzelnen Variablen. Jeweils 2 Variablen stehen für die light Key/heavy Key Splitfraktionen der beiden Kolonnen, die weiteren Variablen geben die relativen Anteile der verschiedenen Ausgänge der Stromteiler an. F bezeichnet den Zulauf (Feed), und die beiden Produkte werden bei $P1$ und $P2$ entnommen.

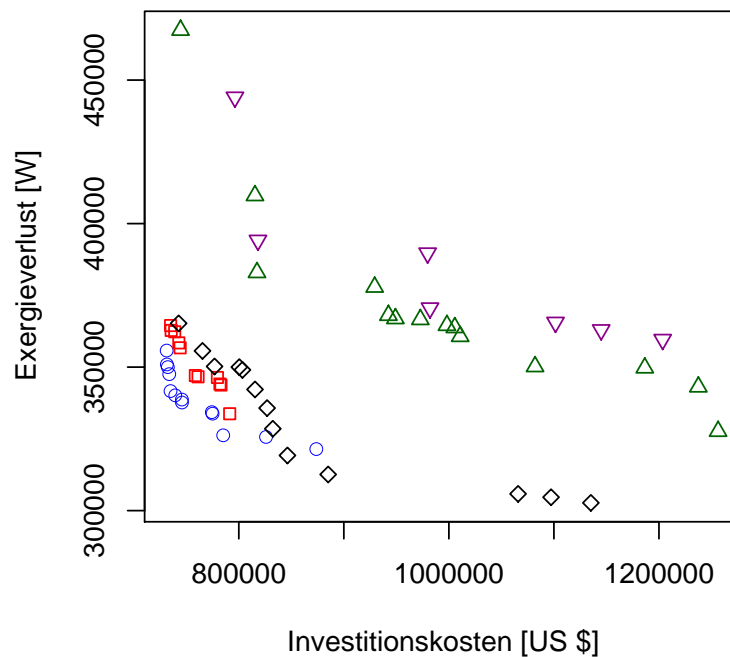


Unser Beispiel kommt aus der Thermodynamik, genauer aus der Auslegung von Destillationsanlagen. Eine einfache Destillationsanlage besteht hier aus mehreren hintereinandergeschalteten Destillationskolonnen, die das zugeleitete Gemisch anhand der unterschiedlichen thermischen Eigenschaften der enthaltenen Stoffe in verschiedene Ströme aufteilt (Abbildung 6 zeigt ein Schaltbild mit den enthaltenen Variablen). Dabei lassen sich in der Realität zumeist keine Reinstoffe erzeugen, denn dazu wären scharfe Trennungen nötig, die nur mit enormem Aufwand näherungsweise erreicht werden können. Dies ist aber auch nicht immer notwendig, da in der chemischen Industrie oft bestimmte Gemische gebraucht werden, die sonst erst wieder durch Rückvermischung zustande kommen würden. Allerdings müssen bestimmte Produktqualitäten eingehalten werden, um die erhaltenen Gemische kommerziell nutzbar zu machen. Es handelt sich also um ein Problem mit Restriktionen, das im Gegensatz zur scharfen Trennung in Reinstoffe nicht mit Hilfe eines einfachen mathematischen Modells gelöst werden kann. Daher verwenden wir mit ASPEN einen kommerziellen Simulator, der für eine vorgegebene Auslegung sowohl die physikalische Gültigkeit als auch die zu erwartenden Kosten und den Energiebedarf berechnet.

Diese beiden Kriterien werden nun anhand des konkreten Problems betrachtet, ein Gemisch von Propan, iso-Butan und n-Butan in 2 Produkte (Gemische) mit verschiedenen Mischungsverhältnissen der 3 Stoffe aufgeteilt werden. Dieses Problem wurde in [23] schon für den einkriteriellen Fall untersucht, und es hatte sich eine Zielfunktionsstopologie mit zahlreichen Inseln gültiger Lösungen ergeben, die von großen Bereichen ungültiger Lösungen getrennt werden. Mit der Einbeziehung der Restriktionen in die Zielfunktionen über Strafterme erhält man also eine multimodale Topologie. Daher kann für die 2-kriterielle Betrachtung ein Problem vom Typ III oder IV erwartet werden.

Tatsächlich zeigt sich im mehrkriteriellen Fall sowohl das eine als auch das andere Verhalten. Da wir hier nicht an der Bewertung von Algorithmen, sondern nur an einem besseren Verständnis des vorliegenden Problems interessiert sind, greifen wir die besten 5 aus einer Reihe von Optimierläufen (30) heraus und analysieren die Lage der Pareto-Fronten und Pareto-Mengen. Die Abbildung 7 zeigt die erzielten Fronten. Bei der besten (untersten) Front zeigt sich ein Typ IV-Effekt, da scheinbar mehrere, unabhängige Teile der Pareto-Front existieren (Kreise und Rauten). Die mit Quadraten bezeichnete Lösungsmenge ist der der Kreise in der Qualität sehr ähnlich und liegt auch im Entscheidungsraum in unmittelbarer Nähe. Hier ist entweder die Optimierung nicht weit genug fortgeschritten, um zu offenbaren, dass diese Lösungen tatsächlich zu Front und Menge der Kreise gehören, oder es handelt sich um eine alternative Pareto-Menge, die auf die gleiche Front führt (Typ III). Im Falle der schlechteren, oberen Front (beide Arten von Dreiecken) liegt ebenfalls ein Verhalten vom Typ III vor. Die beiden Lösungsmengen sind im Entscheidungsraum deutlich voneinander verschieden, aber im Zielraum sehr ähnlich.

Bild 7: Pareto-Fronten der 5 erfolgreichsten aus 30 Läufen, siehe auch die zugehörigen Pareto-Mengen in Abbildung 8. Die beste erhaltene Front wird von den mit Kreisen und Rauten bezeichneten Lösungen gebildet, die im Entscheidungsraum allerdings weit auseinander liegen. Die beiden mit Dreiecken versehenen Lösungsmengen bilden eine weitere, schlechtere Front, auch sie liegen im Entscheidungsraum weit auseinander.



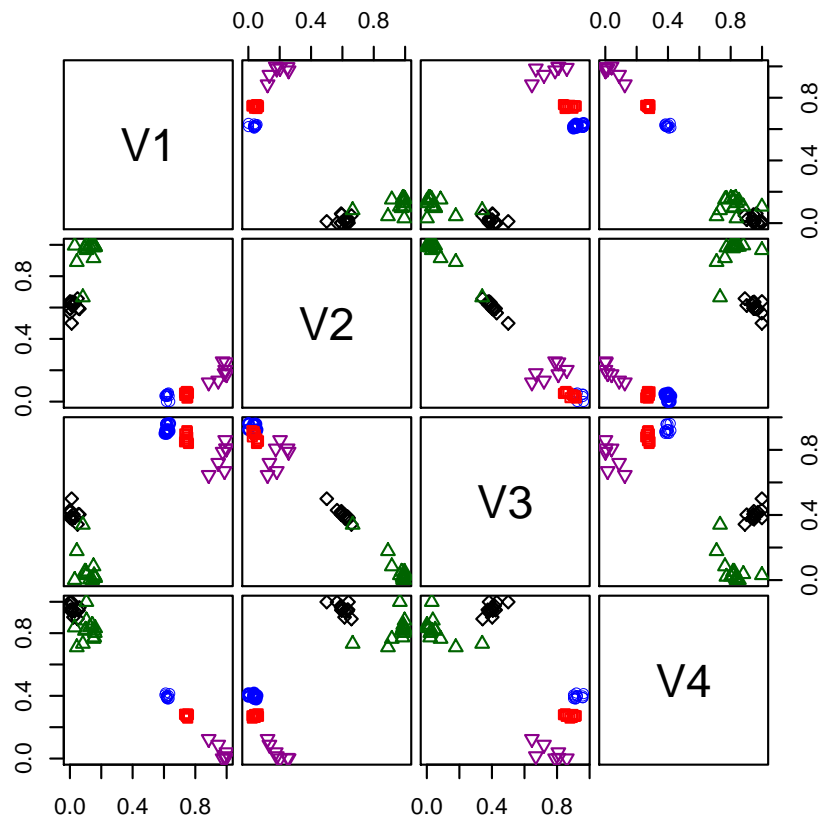
Es drängt sich der Schluss auf, dass sich die Topologie der Zielfunktion ähnlich wie im einkriteriellen Fall als eine Menge von durch ungünstige Bereiche getrennten Inseln darstellt. Die vom verwendeten SMS-EMOA generierten Lösungsmengen stammen in jedem Lauf fast ausschliesslich aus einem im Entscheidungsraum relativ kleinen Bereich, es ist also zu vermuten, dass der Optimieralgorithmus sich relativ schnell auf eine gültige Insel konzentriert und die Annäherung der Pareto-Front dann innerhalb dieser durchführt. Je nach Qualität der in einer Insel erreichbaren Lösungen in den einzelnen Zielfunktionen ergibt sich entweder ein neuer, bisher nicht bekannter Teil der Pareto-Front, oder eine alternative Pareto-Menge für einen schon vorhandenen Teil der Pareto-Front. Es handelt sich also um eine vermischtes Typ III/IV Problem.

3 Abschließende Bemerkungen

Die Intention dieses Beitrags ist es, Aufmerksamkeit für ein bisher wenig beachtetes Phänomen zu wecken: die multi-Modalität in mehr-kriteriellen Problemstellungen. Dazu wurde zuerst eine einfache Funktion betrachtet, die in der einfachsten denkbaren Weise eine uni-modale und eine multi-modale Funktion zu einer mehr-kriteriellen Aufgabenstellung verbindet.

Bereits hier wurde deutlich, dass dadurch weitaus komplexere Situationen zustande kommen, als bisher angenommen. Insbesondere treten Fällen von nicht-zusammenhängenden Pareto-Mengen auf, deren Vorhandensein Standardalgorithmen zur mehr-kriteriellen Optimierung nicht geeignet berücksichtigen. In diesen Algorithmen wird lediglich die Diversität der Population im Lösungsraum berücksichtigt, nicht aber die Diversität der Genotypen im Suchraum. Entsprechend wurde diese Situation bisher kaum untersucht. Ein Auftreten dieser Situationen ist aber recht wahrscheinlich, da davon ausgegangen werden kann, dass eine Erhöhung der Komplexität der Einzelzielfunktionen auch zu einer erhöhten Komplexität der zusammengesetzten mehr-kriteriellen Funktion führt.

Bild 8: Paardiagramme der ersten vier Variablen des Auslegungsproblems (siehe Abbildung 6, dies sind jeweils die heavy Key bzw. light Key Splitfraktionen der beiden Kolonnen), die die 5 Pareto-Mengen zu den in Abbildung 7 dargestellten Fronten enthalten. Die verschiedenen Pareto-Mengen sind fast vollständig disjunkt, und lassen sich jeweils auf sehr enge Bereiche eingrenzen.



Im Anschluss wurde eine neue Zielfunktion konstruiert, welche derartige Situationen nicht-zusammenhängender Pareto-Mengen mit einer einzigen Pareto-Front direkt für Untersuchungen zugänglich macht. Hier zeigte sich ganz konkret das Versagen von Standard-Methoden. Zudem wurden zwei spezielle Verfahren vorgestellt, die die Problematik wesentlich besser beherrschen. Insbesondere ein einfacher Ansatz mit wiederholter einkriterieller Optimierung konnte mit sehr guten Ergebnissen überzeugen.

Die hier untersuchte Zielfunktion wurde zusammen mit der vorgestellten Metrik zum Zählen der erzielten Kacheln zu einem Wettbewerb von evolutionären mehr-kriteriellen Optimierverfahren auf einer international angesehenen Konferenz (Congress on Evolutionary Computation, CEC 2007, Singapur) eingereicht. Auch hier eroberten die zum Wettbewerb angetretenen Optimierverfahren in der Regel nur ein oder zwei Kacheln. Mehr als zwei Kacheln wurden von keinem Optimierverfahren nach einer gewissen Anzahl von Zielfunktionsauswertungen noch gehalten.

Dass es sich zudem nicht nur um ein theoretisches Problem handelt, dass nur bei der Betrachtung von mathematischen Funktionen als Zielfunktion auftritt, konnte im anschließend betrachteten Anwendungsfall der Auslegung von Destillationsanlagen eindrucksvoll gezeigt werden. Unserer Meinung nach bergen eine Vielzahl von realen Aufgabenstellungen ähnliche Strukturen, die von herkömmlichen Algorithmen nicht beachtet werden. Es ist daher an der Zeit, neue Algorithmen zu entwerfen, die aufgrund der Berücksichtigung der Strukturen im Such- **und** im Lösungsraum in der Lage sind, neben guten Zielfunktionswerten allein auch eine Palette von alternativen Lösungen zu finden.

Literatur

- [1] Deb, K.; Agrawal, S.; Pratab, A.; Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In: *Proceedings of the Parallel Problem Solving from Nature VI Conference* (Schoenauer, M.; Deb, K.; Rudolph, G.; Yao, X.; Lutton, E.; Merelo, J. J.; Schwefel, H.-P., Hg.), S. 849–858. Paris, France: Springer. Lecture Notes in Computer Science No. 1917. 2000.
- [2] Deb, K.; Agrawal, S.; Pratab, A.; Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. KanGAL report 200001, Indian Institute of Technology, Kanpur, India. 2000.
- [3] Zitzler, E.; Laumanns, M.; Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In: *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems* (Giannakoglou, K.; Tsahalis, D.; Periaux, J.; Papailou, P.; Fogarty, T., Hg.), S. 95–100. Athens, Greece. 2002.
- [4] Zitzler, E.; Laumanns, M.; Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Techn. Ber. 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland. 2001.
- [5] Emmerich, M.; Beume, N.; Naujoks, B.: An EMO algorithm using the hypervolume measure as selection criterion. In: *Evolutionary Multi-Criterion Optimization (EMO 2005)* (Coello, C. A. C.; et al., Hg.), S. 62–76. Springer, Berlin. 2005.
- [6] Beume, N.; Naujoks, B.; Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research* 181 (2007) 3, S. 1653–1669.
- [7] Preuss, M.; Naujoks, B.; Rudolph, G.: Pareto Set and EMOA Behavior for Simple Multimodal Multiobjective Functions. In: *Parallel Problem Solving from Nature (PPSN IX)* (Runarsson, T. P.; et al., Hg.), Bd. 4193 von *Lecture Notes in Computer Science*, S. 513–522. Springer, Berlin. 2006.
- [8] Rudolph, G.; Naujoks, B.; Preuß, M.: Capabilities of EMOA to Detect and Preserve Equivalent Pareto Subsets. In: *Evolutionary Multi-Criterion Optimization (EMO 2007)* (Obayashi, S.; et al., Hg.), Bd. 4403 von *Lecture Notes in Computer Science*, S. 36–50. Springer, Berlin. 2007.
- [9] De Jong, K. A.: *An analysis of the behavior of a class of genetic adaptive systems..* Dissertation, University of Michigan. 1975.
- [10] Goldberg, D. E.; Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, S. 41–49. Mahwah, NJ, USA: Lawrence Erlbaum Associates, Inc. ISBN 0-8058-0158-8. 1987.

- [11] Beasley, D.; Bull, D. R.; Martin, R. R.: A Sequential Niche Technique for Multimodal Function Optimization. *Evolutionary Computation* 1 (1993) 2, S. 101–125. URL citeseer.ist.psu.edu/beasley93sequential.html.
- [12] Pétrowski, A.: A clearing procedure as a niching method for genetic algorithms. In: *Proceedings of 1996 IEEE International Conference on Evolutionary Computation (ICEC '96), Nagoya* (Fukuda, T.; Furuhashi, T.; Fogel, D. B., Hg.), S. 798–803. Piscataway NJ: IEEE Press. 1996.
- [13] Jelasity, M.: UEGO, an Abstract Niching Technique for Global Optimization. In: *PPSN* (Eiben, A. E.; Bäck, T.; Schoenauer, M.; Schwefel, H.-P., Hg.), Bd. 1498 von *Lecture Notes in Computer Science*, S. 378–387. Springer. ISBN 3-540-65078-4. 1998.
- [14] Wineberg, M.: *Improving the Behavior of the Genetic Algorithm in a Dynamic Environment*. Dissertation, Carleton University, Ottawa, Canada. 2000.
- [15] Li, J.-P.; Balazs, M. E.; Parks, G. T.; Clarkson, P. J.: A species conserving genetic algorithm for multimodal function optimization. *Evolutionary Computation* 10 (2002) 3, S. 207–234.
- [16] Streichert, F.; Stein, G.; Ulmer, H.; Zell, A.: A Clustering Based Niching Method for Evolutionary Algorithms. In: *Genetic and Evolutionary Computation – GECCO-2003* (Cantú-Paz, E., Hg.), S. 644–645. Berlin: Springer-Verlag. ISBN 3-540-40602-6. 2003.
- [17] Shir, O. M.: Niching in Evolution Strategies. In: *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation* (Beyer, H.-G., Hg.), S. 865–872. New York, NY, USA: ACM Press. ISBN 1-59593-010-8. 2005.
- [18] Zitzler, E.; Deb, K.; Thiele, L.: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation* 8 (2000) 2, S. 173–195.
- [19] Deb, K.; Thiele, L.; Laumanns, M.; Zitzler, E.: Scalable Multi-Objective Optimization Test Problems. In: *Congress on Evolutionary Computation (CEC 2002)*, S. 825–830. IEEE Press, Piscataway NJ. 2002.
- [20] Fisher, R. A.: *The Design of Experiments*. Edinburgh: Oliver and Boyd. 1935.
- [21] Chan, K. P.; Ray, T.: An Evolutionary Algorithm to Maintain Diversity in the Parametric and the Objective Space. In: *Proceedings of IEEE International Conference on Computational Robotics and Autonomous Systems (CIRAS 2005)*. Centre for Intelligent Control, National University of Singapore. ISSN: 0219-6131. 2005.
- [22] Steuer, R. E.: *Multiple Criteria Optimization: Theory, Computation, and Application*. Wiley series in probability and mathematical statistics. New York: Wiley. 1986.
- [23] Henrich, F.; Bouvy, C.; Kausch, C.; Lucas, K.; Preuss, M.; Rudolph, G.; Roosen, P.: Economic optimization of non-sharp separation sequences by means of evolutionary algorithms. *Computers chem. Engng.* accepted (2007).

Ein mehrkriterielles Evolutionsverfahren zur Bestimmung des Phasengleichgewichts von gemischten Flüssigkeiten

Günter Rudolph, Mike Preuss

Lehrstuhl für Algorithm Engineering
Universität Dortmund, 44221 Dortmund

Tel.: (0231) 755 - 77 01

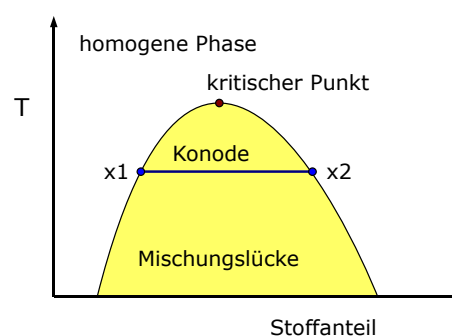
Fax: (0231) 755 - 77 40

E-Mail: {guenter.rudolph, mike.preuss}@uni-dortmund.de

1 Ein vermeintlich einfaches Problem aus der Thermodynamik

Das hier behandelte Problem mutet zunächst einfach an, da es in seiner einfachsten Form nur zwei zu bestimmende Variablen enthält. Es kommt aus dem Kontext der Flüssig-Flüssig-Phasengleichgewichte (englisch *Liquid-Liquid Equilibria* (LLE)), bei dem es vor allem darum geht, die Bereiche zu bestimmen, in denen industrielle Trennverfahren operieren können. Liegt ein Gemisch aus verschiedenen Flüssigkeiten vor, das in seine einzelnen Komponenten zerlegt werden soll, muss es in einen Zustand gebracht werden, in dem die Durchmischung möglichst schlecht ist. Bei günstiger Wahl der Umgebungsbedingungen wie z.B. Temperatur und Druck lässt sich oft die Entmischung erzwingen, so dass das Gemisch in einzelne Phasen zerfällt. Man versucht also, die sogenannte *Mischungslücke* zu finden, die diesen Bereich charakterisiert. In Abbildung 1 ist die Mischungslücke nach oben durch einen kritischen Punkt begrenzt, oberhalb dessen keine getrennten Phasen mehr vorliegen. Dieser Fall tritt beim hier betrachteten Stoffsystem Wasser/Pentanol ein, ist aber nicht der einzig mögliche. Andere Gemische können einen unteren kritischen Punkt haben, oder einen unteren und einen oberen, oder gar keinen.

Bild 1: Die Mischungslücke bezeichnet den Bereich, in dem keine homogene Mischung vorliegt, da das Gemisch in Phasen zerfällt. Für jede Temperatur (T) sind die Endpunkte der entsprechenden Konode gesucht, die die Grenzen zwischen homogener Phase und dem mehrphasigen Bereich darstellen. Am kritischen Punkt fallen die beiden Endpunkte zu einem zusammen.



Jedes durch Angabe von Temperatur, Druck, Dichten, Stoffkonzentrationen usw. vollständig bestimmte System lässt sich durch Lösen des entsprechenden Gleichungssystems relativ genau berechnen. Dadurch wird ermittelt, ob es sich um einen stabilen, d.h. in der Realität tatsächlich erreichbaren Zustand handelt. Ausgehend vom gewünschten chemischen Gleichgewicht lassen sich über Hilfskonzepte wie das der über thermodynamische Modelle berechenbaren Fugazität $\varphi_i^\alpha(T, p, x_i^\alpha)$ entsprechende (nichtlineare) Gleichungen aufstellen, wie z.B. für das einfache Zweistoffsystem Wasser/Pentanol:

$$x_i^\alpha \cdot \varphi_i^\alpha = x_i^\beta \cdot \varphi_i^\beta, \quad (i = 1, 2). \quad (1)$$

Dabei stehen die x_i für die beiden Stoffe, und die oberen Exponenten α und β für die verschiedenen Phasen. Sei x_1 Wasser und α die wasserreiche Phase (β die pentanolreiche Phase), so ergeben sich die Endpunkte der Konode für eine bestimmte Temperatur T unter konstantem Druck P aus der Bestimmung von x_1^α und x_1^β , der Konzentrationen von Wasser in beiden Phasen. Die beiden Variablen des entsprechenden Optimierungsproblems sind dann $x1 = x_i^\alpha$ und $x2 = x_i^\beta$. Nach der Berechnung der Fugazitätskoeffizienten erhält man aus den beiden Gleichungen (1) zwei Fehlerwerte e_1 und e_2 , die jeweils möglichst klein ($e_i \leq 10^{-10}$) werden sollen. Als Zielfunktion verwenden wir die Summe der quadratischen Fehler (2), die zu minimieren ist:

$$f(x1, x2) = e_1^2 + e_2^2. \quad (2)$$

Leider hat diese Modellierung des Problems den Nachteil, dass die Gleichungen (1) unerwünschte triviale Lösungen zulassen, z.B. bei $x_i^\alpha = x_i^\beta$, also gleicher Konzentration eines Stoffes in beiden Phasen. Da der Fehler hier 0 wird, sind alle diese Punkte sogar global optimal, während das gewünschte Optimum aufgrund numerischer Schwierigkeiten in der Fugazitätsmodellierung nur näherungsweise bestimmt werden kann und damit immer ein lokales Optimum bleibt. Die Gitterplots in Abbildung 2 zeigen zusätzlich, dass es ein sehr dominantes Attraktionsbassin gibt, das zu einer trivialen Lösung führt. Das Attraktionsgebiet des gewünschten Optimums ist dagegen so klein, dass es ohne Transformation kaum sichtbar ist. Für die Optimierung folgt also, daß man nach einem sehr schwer zu findenden lokalen Optimum sucht. Es versteht sich von selbst, dass herkömmliche gradientenbasierte Verfahren, wie sie oft in der Thermodynamik zum Einsatz kommen, auf einem solchen Problem völlig überfordert sind.

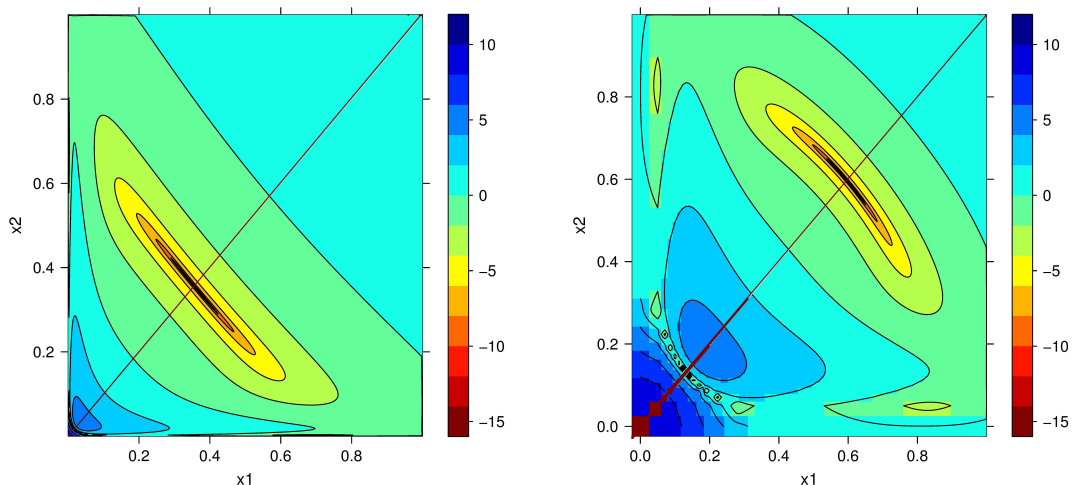


Bild 2: Links: Gitterplot über die Variablen $x1$ und $x2$. Rechts: mit einer Wurzelfunktion transformiert. Nur im rechten Bild lassen sich die gewünschten Optima bei ca. (0.650, 0.001) bzw. (0.001, 0.650) erkennen. Die Diagonale stellt die triviale Lösungsmenge $x1 = x2$ dar.

2 Lösungsansätze mit kanonischen evolutionären Algorithmen

Aus den beschriebenen Eigenschaften des behandelten Optimierungsproblems lässt sich schon vermuten, dass neben klassischen Verfahren auch kanonische evolutionäre Algorithmen Schwierigkeiten mit dem Auffinden des gewünschten lokalen Optimums haben werden. Allerdings können letztere leicht mit einem Multistart-Mechanismus ausgestattet werden, der die Suche bei Stagnation abbricht und an anderer Stelle fortsetzt. Außerdem ist zu hoffen, dass man die Eigenschaften der EAs über eine automatisierte Parametersuche so weit an das Optimierungsproblem anpassen kann, dass sich ihre Erfolgswahrscheinlichkeit deutlich steigern lässt. Wir setzen dazu die *Sequentielle Parameteroptimierung* (SPO) [1] ein, die auf Basis eines initialen, raumfüllenden Designs ein gemischtes Regressions- und Krigingmodell anpasst und anhand dessen sukzessive nach erfolgreicher Parameterkombinationen sucht. Dieser Ansatz hat den Vorteil, dass die Ergebnisse nicht durch ungeeignete Parametrisierung der untersuchten Optimierverfahren verfälscht werden. Die erhaltene Aussage betrifft also nicht nur die Algorithmen unter Einsatz von standardisierten oder manuell gesuchten Parameterwerten, sondern ihre parameterbasierte Anpassbarkeit an das Problem insgesamt.

Die betrachteten drei Optimierverfahren stehen prototypisch für verschiedene, relativ einfache Möglichkeiten, den Suchprozess in Zeit und/oder Raum zu parallelisieren. Dabei kommt bei dem behandelten Problem der Möglichkeit zum mehrfachen Neustart eine besondere Bedeutung zu, da das aufzufindende Attraktionsbassin sehr klein ist. Wird nicht wenigstens ein Suchpunkt in diesem Bassin plaziert, bleibt die Suche zwangsläufig erfolglos.

Multistart (1+1) Evolutionstrategie (ES): Aus einem Elternindividuum wird durch Mutation jeweils ein Nachkomme generiert, der nur akzeptiert wird, wenn er mindestens genauso gut ist wie sein Elter. Der Neustart erfolgt hier nach einer festgelegten Anzahl von Generationen ohne Verbesserung.

Multistart (μ, λ) Evolutionstrategie: Populationsbasierte Suche, λ Nachkommen werden aus μ Eltern durch Rekombination und Mutation erzeugt, nur die besten μ Nachkommen überleben und werden Eltern der nächsten Generation. Hier wird der Neustart ebenfalls nach einer festgelegten Stagnationszeit ausgelöst.

Multistart Small-World Netzwerk zellulärer EA (swn-CEA): Hier handelt es sich um eine Population einzelner (1+1) Strategien, die untereinander in einer Ringtopologie verbunden sind. Einige der Kanten im Ring werden nach dem Verfahren von Watts und Strogatz [2] zufällig mit anderen als ihren Nachbarknoten verbunden. Durch Setzen oder Löschen dieser abkürzenden Kanten kann die Gesamtkonnektivität des Systems drastisch erhöht oder verringert werden. Wenn für eine festgelegte Anzahl an Generationen keine „Zelle“ eine Verbesserung erzielt, wird der Algorithmus neu gestartet.

In Tabelle 1 sind die für SPO erlaubten Parameterintervalle angegeben. Insgesamt durfte die Parameteroptimierung für jeden EA jeweils 1000 Algorithmen-Läufe durchführen, wobei ein initiales Design der Größe 100 mit je 4 Wiederholungen benutzt wird.

Die Ergebnisse der Parameteroptimierung sind in Tabelle 2 zusammengefasst. Zwei Gesichtspunkte sind bei deren Betrachtung von besonderem Interesse:

- Wie weit kann die Performance für die einzelnen Varianten gesteigert werden?

Tabelle 1: Algorithmendesigns (Parameterintervalle) für SPO auf den drei EA Vergleichsverfahren: Populationsgröße, Selektionsdruck, Lernrate, initiale Mutationsstärke, tolerierte Stagnationszeit, Wahrscheinlichkeit für Kantenreueverbindung

EA	μ	$\frac{\lambda}{\mu}$	τ	σ_{init}	s	β
(1+1)-ES	(1)	(1)	0 : 1	0.01:0.5	10:100	—
(μ,λ) -ES	5:150	3:10	0 : 1	0.01:0.5	1:20	—
swn-CEA	5:150	(1)	0 : 1	0.01:0.5	1:20	0:0.2

- Lassen sich systematische Veränderungen an den Parametern erkennen, die als Anpassung in eine bestimmte Richtung interpretiert werden können?

Nimmt man die durchschnittliche Zahl der Treffer im gewünschten Attraktionsbassin als Richtwert für die Performance, so fällt zunächst auf, dass die (1+1)-ES sowohl vor als auch nach der Parameteroptimierung den anderen beiden EA deutlich überlegen ist. Der swn-CEA wird soweit an das Problem angepasst, dass er ebenfalls eine beachtliche Performance erreicht. Die (μ,λ) -ES ist allerdings offenbar für das Problem ungeeignet. Hier kommt allerdings möglicherweise auch ein Problem fehlender Rückkopplung in der Parameteroptimierung zum Tragen. Da die Performanceunterschiede innerhalb des initialen Designs bei dieser Strategie sehr klein sind, enthält das von SPO intern verwandte Modell keine Information darüber, wo bessere Konfigurationen gefunden werden können. Das heißt nicht notwendigerweise, daß diese nicht existieren, sie sind dann aber sehr schwer zu finden.

Tabelle 2: Mittlere und beste gefundene EA-Parametrisierungen (siehe auch Tabelle 1), $\varnothing h_i$ bezeichnet die durchschnittliche Zahl der Bassin-Treffer pro 10000 Auswertungen

μ	$\frac{\lambda}{\mu}$	τ	σ_{init}	s	β	$\varnothing h_i$
(1+1)-ES, oben: mittlere Startkonfiguration, unten: beste SPO-Konfiguration						
(1)	(1)	0.635	0.385	61	—	4.13
(1)	(1)	0.18	0.350	10	—	7.50
(μ,λ) -ES, oben: mittlere Startkonfiguration, unten: beste SPO-Konfiguration						
13	6.89	0.125	0.066	11	—	0.00
7	3.24	0.020	0.493	18	—	0.53
swn-CEA, oben: mittlere Startkonfiguration, unten: beste SPO-Konfiguration						
77	(1)	0.075	0.110	7	0.035	0.33
5	(1)	0.033	0.500	1	0.140	4.00

Vergleicht man die erhaltenen besten Parameterwerte aus Tabelle 2 miteinander, fällt schnell auf, dass die verbesserten EA, vor allem (1+1)-ES und swn-CEA, sich durch extrem kleine Toleranzen gegenüber Stagnation auszeichnen. Desweiteren ist die Populationsgröße bei letzterem stark reduziert worden. Scheinbar ahmt der angepasste swn-CEA das Verhalten einer (1+1)-ES nach. Bei der (μ,λ) -ES ist ebenfalls ein Trend zu kleineren Populationen zu beobachten, der sich allerdings im Ergebnis wesentlich schwächer auswirkt.

Zusammenfassend lässt sich sagen, dass einige kanonische EA sich durch eine Parameteroptimierung an das Problem soweit anpassen lassen, dass sie das gesuchte Attraktionsbassin von Zeit zu Zeit treffen. Trotz aufwändiger Parameteroptimierung ist die erreichte Performance aber noch immer kaum zufriedenstellend. Außerdem soll angemerkt werden, dass die präzise Approximation des lokalen Optimums selbst innerhalb des Bassins für die (1+1)-ES relativ schwierig ist. Fehlerwerte unterhalb von 10^{-6} werden nur selten in vertretbarer Zeit erreicht.

3 Mehrkriterieller Lösungsansatz und Hybridisierung

Aus den soeben geschilderten Erfahrungen lässt sich ableiten, dass das Bassin irgendwie „größer gemacht“ werden muss, um häufiger Treffer in der Nähe der gesuchten (nichttrivialen) Optima zu platzieren. Dazu muss problemspezifisches Wissen bei der *Problemformulierung* geeignet eingearbeitet werden: Wir wissen bereits, dass auf der Raumdiagonalen $x_1 = x_2$ die nicht gewünschten, trivialen Optima liegen. Also könnte man mit Hilfe von Straffunktionen die Nähe einer Lösung zur Raumdiagonalen bestrafen, um den Algorithmus von den trivialen Lösungen fortzuleiten. Problematisch sind hier allerdings die zu wählenden Parameter bei der Straffunktion. Deshalb lag es nahe, hier einen mehrkriteriellen Ansatz zu verfolgen: Die erste Zielfunktion $f_1(\cdot, \cdot)$ bleibt das zu minimierende Fehlermaß aus Gleichung (2), während die zweite Zielfunktion die Ähnlichkeit der beiden Komponenten x_1 und x_2 durch

$$f_2(x_1, x_2) = 1 - |x_1 - x_2| \quad \text{mit } x_1, x_2 \in [0, 1] \quad (3)$$

minimiert. Wendet man darauf einen mehrkriteriellen evolutionären Algorithmus an, so erhält man eine Approximation der Paretofront. Die zugehörigen Urbilder stellen dann die Bandbreite der effizienten Kompromisse aus niedrigem Fehlermaß *und* der Größe der Entfernung zu den trivialen Lösungen dar. Folglich darf man erwarten, dass zumindest *einige* Approximationspunkte des mehrkriteriellen Verfahrens im Attraktionsbassin der nichttrivialen Lösungen liegen.

Diese Hypothese haben wir mit dem SMS-EMOA [3, 4] getestet. Dieses mehrkriterielle evolutionäre Verfahren nutzt intern die **S**-Metrik zur **S**elektion (SMS) von Individuen und erzielt so eine gleichmäßige Approximation der Paretofront. Abbildung 3 zeigt eine typische Approximation des SMS-EMOA nach 2500 Funktionsauswertungen bei einer Populationsgröße von 20 Individuen.

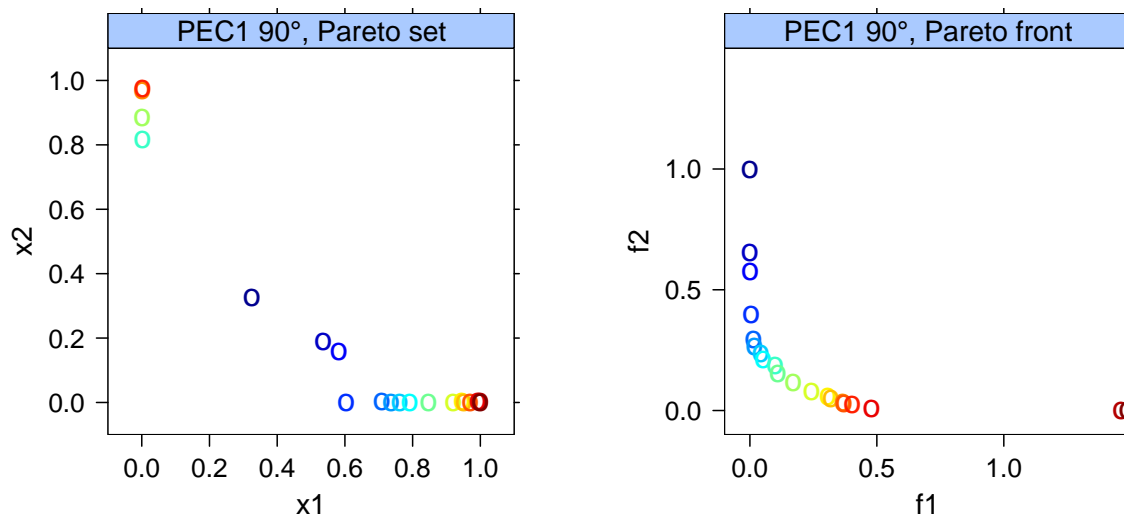


Bild 3: Links: Typische Approximation durch den SMS-EMOA. Links: Paretomenge. Rechts: Paretofront. Das interessierende Gebiet liegt entweder bei $x_1 = 0,65$ oder bei $x_2 = 0,65$.

Ganz offensichtlich ist es durch diesen Ansatz möglich, Lösungen in der Nähe der gesuchten Optima zu platzieren. Allerdings ist die Lokalisationsgenauigkeit dieser Optima noch nicht zufrieden stellend. Deshalb wird von jeder Lösung aus dem Lauf des SMS-EMOA eine lokale Suche gestartet, die die gefundenen Optima genauer lokalisieren soll. Für das

resultierende hybride Verfahren verwenden wir das direkte Suchverfahren von HOOKE & JEEVES [5], das ohne die Berechnung von Gradienten auskommt. Konkret setzen wir die C-Implementierung von Johnson [6] ein, welche auf dem ALGOL-Code in [7] sowie den numerischen Code-Verbesserungen in [8, 9] beruht.

Das lokale Verfahren wird abgebrochen, sobald $|x_1 - x_2| < 0,01$, da man dann von einem Lauf zu einer trivialen Lösung ausgehen kann. Ansonsten stoppt das Verfahren, sobald für die interne Schrittweite s die Bedingung $s < 10^{-10}$ gilt. Nach Beendigung der lokalen Suche gemäß letzterer Bedingung wurde die Suche als erfolgreich gewertet, wenn das Fehlermaß kleiner als 10^{-10} war.

4 Ergebnisse und Analyse

Die hier vorgestellten Ergebnisse erweitern die bereits in [10] beschriebenen Resultate. Das hybride Verfahren aus SMS-EMOA und direkter Suche nach HOOKE & JEEVES wurde 25 mal auf das Wasser/Pentanol-Problem jeweils für die Temperaturen $T = 40, 50, 60, 70, 80, 90$ Grad Celsius angewandt. Folglich wurden bei einer Populationsgröße von 20 Individuen nach Ende des SMS-EMOA genau 20 lokale Suchen mit je einem Individuum als initiale Lösung gestartet. Dabei haben wir *keinerlei* Parameter-tuning bei beiden Verfahren durchgeführt: Beim direkten Suchverfahren wurde der Schrittweitenkontraktionsfaktor auf $\rho = 0,5$ und die initiale Schrittweite auf $s = 0,001$ gesetzt. Der SMS-EMOA stoppte nach 2500 Funktionsauswertungen.

Tabelle 3 zeigt die aggregierten Leistungsdaten des hybriden Verfahrens. So waren etwa bei $T = 60^\circ\text{C}$ im Mittel 10 von 20 lokalen Suchen in *jedem* der 25 Läufe erfolgreich bei der Suche nach den nichttrivialen Lösungen. Selbst im schlechtesten Fall waren mindestens drei lokale Suchen erfolgreich, im besten Fall sogar 17.

Tabelle 3: Mittlere Erfolgsquoten für das hybride Verfahren mit jeweils 20 lokalen Suchen nach jeweils 25 Läufen je Temperatur.

Temperatur	Minimum	Maximum	Median	Mittelwert	Quote
40°C	2	14	7	$7,00 \pm 3,1$	35,0 %
50°C	4	14	7	$7,98 \pm 2,8$	39,9 %
60°C	3	17	10	$10,00 \pm 3,3$	50,0 %
70°C	7	16	11	$11,32 \pm 3,2$	56,6 %
80°C	7	18	13	$13,12 \pm 2,6$	65,6 %
90°C	9	19	15	$15,12 \pm 2,9$	75,6 %

Es gilt nun auszuschließen, dass dieses erfreuliche Verhalten maßgeblich durch das HOOKE & JEEVES Verfahren hervorgerufen wird. Deshalb haben wir $25 \times 20 = 500$ lokale Suchen von gleichverteilten Startpunkten in $[0, 1]^2$ gestartet und die gleichen Abbruchbedingungen wie beim hybriden Verfahren angewandt. Tabelle 4 zeigt deutlich, dass die hohen Erfolgsquoten beim hybriden Verfahren nicht durch das spezielle lokale Suchverfahren sondern durch die Generierung von Startlösungen mit Hilfe des mehrkriteriellen Ansatzes bedingt sind.

Zweifelsfrei stellt die Erhöhung der mittleren Erfolgsquote von höchstens 3% (bei 90°C) beim lokalen Multistartverfahren auf mindestens 35% (bei 40°C) beim hybriden Verfahren eine enorme Verbesserung dar. Zur genaueren Abschätzung der Verbesserung müssen

Tabelle 4: Mittlere Erfolgsquoten für HOOKE & JEEVES nach jeweils 500 Läufen bei gleichverteilten Startpunkten.

Temperatur	Läufe zur trivialen Lösung	Läufe zur gesuchten Lösung	Läufe mit vorzeitiger Konvergenz	mittlere Erfolgsquote
40°C	494	4	2	0,8 %
50°C	490	9	1	1,8 %
60°C	494	4	2	0,8 %
70°C	493	5	2	1,0 %
80°C	486	12	2	2,5 %
90°C	482	14	4	2,9 %

allerdings die 2500 Zielfunktionsauswertungen der mehrkriteriellen Phase hinzugerechnet werden. Tabelle 5 zeigt die mittlere Anzahl von Zielfunktionsaufrufen, die das lokale Verfahren benötigt, bis ein Abbruchkriterium greift. Zwei Dinge fallen auf:

Erstens: Für die Lokalisierung trivialer Lösungen wird eine Größenordnung weniger Auswertungen benötigt. Dies erklärt sich durch das effiziente Abbruchkriterium, das die offensichtlich erfolglose Suche frühzeitig beendet.

Tabelle 5: Mittlere Laufzeiten von HOOKE & JEEVES während des hybriden Verfahrens abhängig von der letztlich gefundenen Lösung.

Temperatur	Mittlere Anzahl von Zielfunktionsauswertungen für triviale Lösungen	Mittlere Anzahl von Zielfunktionsauswertungen für gesuchte Lösungen
40°C	198,4 ± 90,9	1680,3 ± 1706,8
50°C	209,1 ± 73,2	3073,4 ± 2129,3
60°C	182,1 ± 69,0	2246,3 ± 1553,7
70°C	173,0 ± 57,9	2475,3 ± 1750,8
80°C	149,9 ± 83,6	2413,9 ± 1372,2
90°C	130,4 ± 58,7	1241,4 ± 1100,0

Zweitens: Die Streuung bei der Anzahl der Auswertungen für die gesuchten, nichttrivialen Lösungen ist ziemlich hoch. Eine genauere Analyse zeigt, dass die Anzahl der benötigten Auswertungen signifikant davon abhängt, welches der beiden (zur Winkelhalbierenden symmetrischen) gesuchten Lösung letztlich lokalisiert wird. So zeigt sich etwa in Tabelle 6, dass Lösungen vom Typ 2 (mit $x_1 > 0,5$) unabhängig von der Temperatur mit einer gleichbleibend niedrigen Anzahl von rund 460 Auswertungen akkurat bestimmt werden können, während Lösungen von Typ 1 (mit $x_1 \leq 0,5$) eine temperaturabhängige, rund eine Größenordnung höhere Anzahl von Auswertungen benötigt. Da das Problem eigentlich symmetrisch sein sollte, sind die Gründe für diese unterschiedlichen Laufzeiten zur Zeit ungeklärt. Die Vermutung von numerischen Instabilitäten im Simulator liegt jedoch nahe.

Nach diesen vorbereitenden Analysen können wir uns nun der genaueren Beurteilung der Leistungsfähigkeit des hybriden Verfahrens widmen. Zunächst einmal halten wir fest, dass wir zuverlässig an einem hinreichend großen Wert von $|x_1 - x_2|$ bei einem Fehlermaß $< 10^{-10}$ erkennen können, ob wir eine gesuchte Lösung gefunden haben. In diesem Fall könnten wir das lokale Multistartverfahren oder das hybride Verfahren stoppen. Unter diesen Voraussetzungen und der Annahme, dass die Reihenfolge der Startlösungen im hybriden Verfahren zufällig permutiert wird, können wir aus unseren statistischen Daten

Tabelle 6: Mittlere Laufzeiten von HOOKE & JEEVES des hybriden Verfahrens abhängig von gefundener nichttrivialer Lösung.

Temperatur	Mittlere Anzahl von Zielfunktionsauswertungen für Lösungen Nr. 1	Mittlere Anzahl von Zielfunktionsauswertungen für Lösungen Nr. 2
40°C	4168,8 ± 1183,6	456,7 ± 30,3
50°C	5117,6 ± 1264,6	456,1 ± 20,8
60°C	3905,5 ± 465,4	459,9 ± 23,4
70°C	4254,6 ± 753,2	465,9 ± 31,7
80°C	3446,7 ± 472,4	470,8 ± 39,0
90°C	3300,2 ± 465,1	464,3 ± 24,7

eine erwartete Anzahl von Zielfunktionsauswertungen schätzen, die bis zum Auffinden einer gesuchten Lösung bei beiden Verfahren benötigt wird.

Seien s_L und s_H die mittleren Erfolgsquoten des lokalen Multistartverfahrens (L) bzw. des hybriden Verfahrens (H). Diese lassen sich aus der 6. Spalte von Tabelle 4 bzw. der 5. Spalte von Tabelle 3 ablesen. Weiter seien M_T und M_G die mittlere Anzahl von Zielfunktionsauswertungen, die das lokale Verfahren zur Lokalisierung der trivialen (T) bzw. gesuchten (G) Lösung benötigt. Diese Werte finden sich in der 2. bzw. 3. Spalte von Tabelle 5. Unter Verwendung dieser Kenngrößen benötigt der lokale Multistartansatz im Mittel

$$\frac{1}{s_L} \times M_T + M_G \quad (4)$$

und das hybride Verfahren

$$2500 + \frac{1}{s_H} \times M_T + M_G \quad (5)$$

Auswertungen bis zum Auffinden einer gesuchten Lösung. Setzt man die empirischen Werte aus den Tabellen in den Gleichungen (4) und (5) ein, so ergeben sich die in Tabelle 7 zusammengefassten Ergebnisse. Die Überlegenheit des hybriden Verfahrens, insbesondere bei den schwierigeren Problemen mit niedriger Temperatur, tritt deutlich hervor.

Tabelle 7: Mittlere Anzahl von Zielfunktionsauswertungen bis zum Auffinden einer nichttrivialen Lösung beim Multistartverfahren und beim hybriden Verfahren.

	40°C	50°C	60°C	70°C	80°C	90°C
Multistart	26480	14690	25009	19775	8410	5738
Hybrid	4747	6097	5111	5281	5142	3914

Ein weiterer Vorteil des hybriden Verfahrens liegt in der geringeren Varianz bei der Anzahl der Läufe bis zu einer gesuchten Lösung. Die Standardabweichung errechnet sich aus der Erfolgswahrscheinlichkeit s via $\sqrt{1-s}/s$, so dass bei etwa 40°C die Schwankung beim Multistartverfahren bei $\pm 124,5$ und beim hybriden Verfahren nur bei $\pm 2,3$ Läufe liegt.

5 Abschließende Bemerkungen

Das vermeintliche einfache, niedrigdimensionale Problem hat einige Schwächen von Standardversionen evolutionärer Algorithmen offenbart. Erst durch das Einbringen von pro-

blemspezifischem Wissen in die Problemformulierung samt dem mehrkriteriellen Ansatz konnte das Problem mit vertretbarem Aufwand gelöst werden.

Zur Zeit wird dieser Ansatz auch für mehr als zwei Phasen und mehr als zwei Komponenten weiter entwickelt. Dies erfordert eine Reformulierung des Optimierungsmodells und ggf. weitere Zielfunktionen. In diesem Rahmen werden auch die Parametrisierungen der einzelnen Algorithmen mit SPO [1] optimiert werden und alternative direkte Suchverfahren [11] zum Einsatz kommen.

Literatur

- [1] Bartz-Beielstein, T.: *Experimental Research in Evolutionary Computation – The New Experimentalism*. Natural Computing Series. Berlin: Springer. ISBN 3-540-32026-1. 2006.
- [2] Watts, D. J.; Strogatz, S. H.: Collective Dynamics of 'small-world' Networks. *Nature* 393 (1998), S. 440–442.
- [3] Emmerich, M.; Beume, N.; Naujoks, B.: An EMO algorithm using the hypervolume measure as selection criterion. In: *Evolutionary Multi-Criterion Optimization (EMO)* (Coello, C. A. C.; et al., Hg.), S. 62–76. Springer, Berlin. 2005.
- [4] Beume, N.; Naujoks, B.; Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research* 181 (2007) 3, S. 1653–1669.
- [5] Hooke, R.; Jeeves, T. A.: Direct Search Solution of Numerical and Statistical Problems. *Journal of the ACM* 8 (1961) 2, S. 212–229.
- [6] Johnson, M. G.: Implementation of HOOKE & JEEVES Direct Search Algorithm in C. Online: <http://www.netlib.org/opt/hooke.c>. Feb. 1994.
- [7] Kaup Jr., A. F.: Algorithm 178 Direct Search. *Communications of the ACM* 6 (1963) 6, S. 313–314.
- [8] Bell, M.; Pike, M. C.: Remark on Algorithm 178 [E4] Direct Search. *Communications of the ACM* 9 (1966) 9, S. 684–685.
- [9] Tomlin, F. K.; Smith, L. B.: Remark on Algorithm 178 [E4] Direct Search. *Communications of the ACM* 12 (1969) 11, S. 637–638.
- [10] Preuss, M.; Rudolph, G.; Tumakaka, F.: Solving Multimodal Problems via Multiobjective Techniques with Application to Phase Equilibrium Detection. In: *Proceedings of the International Congress on Evolutionary Computation (CEC2007)*. Piscataway (NJ): IEEE Press. Im Druck. 2007.
- [11] Kolda, T. G.; Lewis, R. M.; Torczon, V. J.: Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods. *SIAM Review* 45 (2003) 3, S. 385–482.

First Steps to Future Applications of Spinal Neural Circuit Models in Neuroprostheses and Humanoid Robots

Yang Chen^{1,2}, Christian Bauer¹, Ole Burmeister¹, Rüdiger Rupp³,
Ralf Mikut¹

¹Forschungszentrum Karlsruhe GmbH, Institute for Applied Computer Science

²Universität Karlsruhe (TH)

³Orthopädische Universitätsklinik Heidelberg

E-Mail: {yang.chen, christian.bauer, ole.burmeister, ralf.mikut}@iai.fzk.de,
Ruediger.Rupp@ok.uni-heidelberg.de

1 Introduction

Modeling how humans control and coordinate their locomotion in daily life, such as walking and grasping, is an important research topic with promising application perspectives in medical and rehabilitation engineering, humanoid robotics, and other biological-inspired systems.

In recent years, the concept of central pattern generator (CPG), which has the ability to generate self-sustained rhythmic motor patterns even in the absence of supraspinal inputs and sensory afferent feedbacks [1, 2], has drawn much attention from engineers by showing novel application perspectives in neuroprostheses and humanoid robots.

Although for a thorough understanding of the CPG network there is still a long way to go, much work has been done in the last decades, ranging from deciphering the anatomical structure [3, 4] and computational modeling study [1, 5] to implementation of prototypes for engineering applications [6, 7].

In this paper, we outline the relevant topics of neural control for locomotion (Sec. 2), focusing on the functional significance of the CPG network in this context (Sec. 3). By employing a computational modeling approach (Sec. 4), we develop a simulation model (Sec. 5) and outline possible comparison strategies with human data (Sec. 6), aiming to gain further insights of spinal neural circuit models and evaluate their potentials in future neuroprosthetic and humanoid robotics applications.

2 Function and structure of motor control systems

The human neural control system for posture and movement is a complex, nonlinear system with a hierarchical structure (Fig. 1). At the top level, the volitional command is transmitted via the corticospinal tract, which directly connects the motor cortex to the spinal cord with cross over parts in the brainstem. The sensorimotor cortex is responsible for the strategic decision making, such as trajectory planning and grasp mode selection in dexterous grasping, as well as the high level sensory integration of visual and audio feedback. The cerebellum plays an important role in the precise coordination in time and space of locomotion. It is the processing center of the proprioceptive information (e.g. hip joint position) conveyed by the spinocerebellar tract [8, 9].

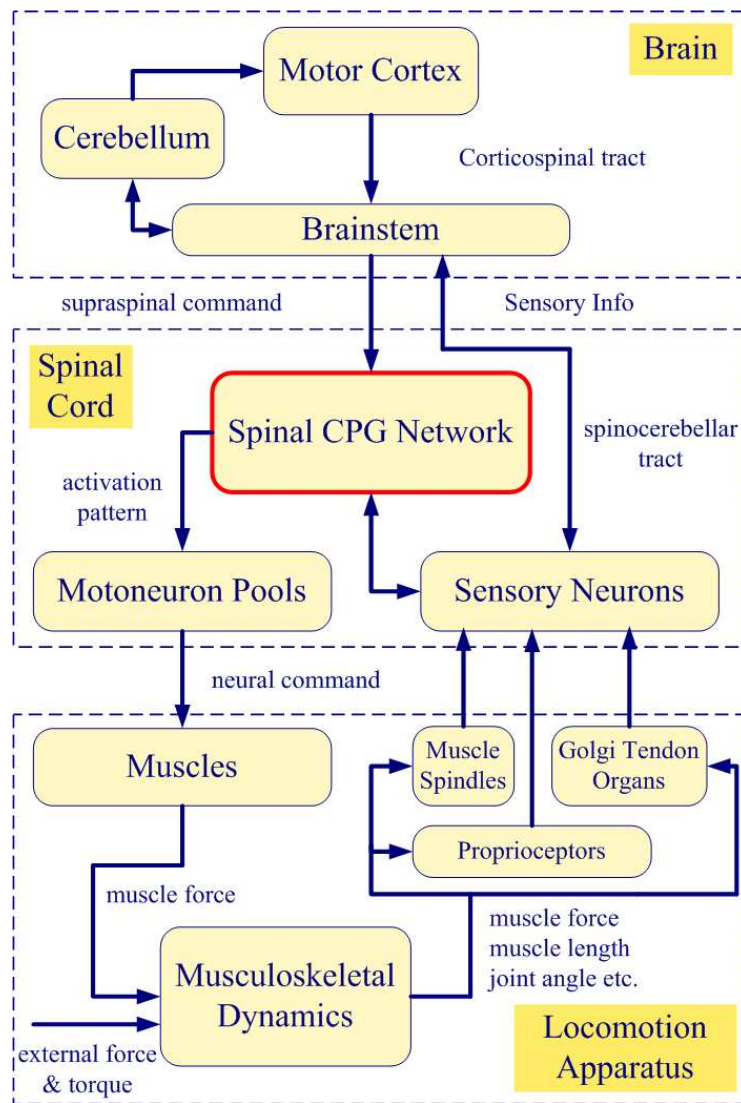


Figure 1: Block diagram of motor control system

The spinal cord forms the intermediate level and serves as coordinator and bridging element between the high level intelligence, the effectors and proprioceptors at the lower end of the system.

Opposing the traditional view, which believes that the neural reflexes only take up very primitive functions in a passive way, accumulating scientific evidences have proved that a significant level of control is mediated in the spinal cord and finally led to the concept of CPG.

Motoneurons in the ventral root of the spinal cord comprise the final common pathway of the neural commands. They interface with the musculoskeletal system via the neuromuscular junctions. The neuromuscular transformation, that is, how neural command is translated into muscle force, presents another modeling challenge to researchers. This process involves complex interactions between motoneurons and muscle fibers in the motor units, and the forward dynamics of the musculoskeletal system which converts the muscle force into movements of the locomotion apparatus. Unlike many common seen actuators in robots, biological muscles will actively adapt their stiffness with respect to

the muscle length measured by the muscle spindles and the force sensed by the golgi tendon organ.

A salient feature of the biological locomotor control system is the prevalence of bidirectional interaction at all levels. The CPG spinal circuit is by no means just an obedient servant who passively executes the commands sent by the motor cortex. Conversely, it actively shapes the supraspinal signal to integrate it into the context of ongoing locomotion. The biological evidence of this feature and its implication on robotic design is discussed in [10].

3 CPG network

3.1 Functional structure of the CPG network

The term CPG refers to a functional network consisting of a set of spinal neurons, which could generate rhythmic activation patterns resembling those observed in actual locomotion even in the absence of supraspinal controls. It is assumed that there is at least one such CPG for each limb. But opinions diverge as whether these CPGs are localized or distributed along the spinal cord of mammalian and primates including human [11]. The existence of such spinal reflex circuits are unequivocally proved in cats through the fictive locomotion observed in decerebrated unanesthetized preparations. Up till now a similar functional neural network in humans can only be demonstrated in an indirect way. However, sufficiently plenty of evidences have been obtained since recent years to make a convincing hypothesis of a spinal CPG network in human [12–14].

Deciphering the structure of the CPG network is traditionally carried out with help of electrophysiological analysis of the target species, and more recently with genetic methods [3]. In our work we take the model presented in [1, 5], which, despite its simple scheme, is able to intuitively outline the underlying neural structure of some properties observed in the fictive locomotion of cat. The involved neurons and their interconnection are depicted in Fig. 2.

Based on the half-center concept, the rhythmic pattern of the CPG network is generated by two populations of tightly coupled pace-maker neurons (RG-E and RG-F populations). The reciprocal inhibition mediated by the interneurons IN RG-E and IN RG-F between the two half-centers is responsible for the synchronized alternating transition between flexion and extension phases. In addition to the rhythm generation network, a pattern formation network consisting of the populations PF-E and PF-F, which are similar to those in the RG network but with a weaker bursting property, is incorporated here to allow a separate control of amplitude and timing of flexor and extensor motoneuron activities. This is independent of the frequency and phase of the locomotor oscillations set by the rhythm generation network. The output of the pattern formation network drives the collateral motoneuron pools to produce the motor activation patterns. The Renshaw cells, which are a set of interneurons providing recurrent inhibition to the motoneuron, form a local negative feedback loop to protect the muscle from over excitation, known as Renshaw inhibition.

Although the CPG network is able to produce rhythmic output independently, sensory inputs play an important role in sculpting the motor activation patterns. Observations in

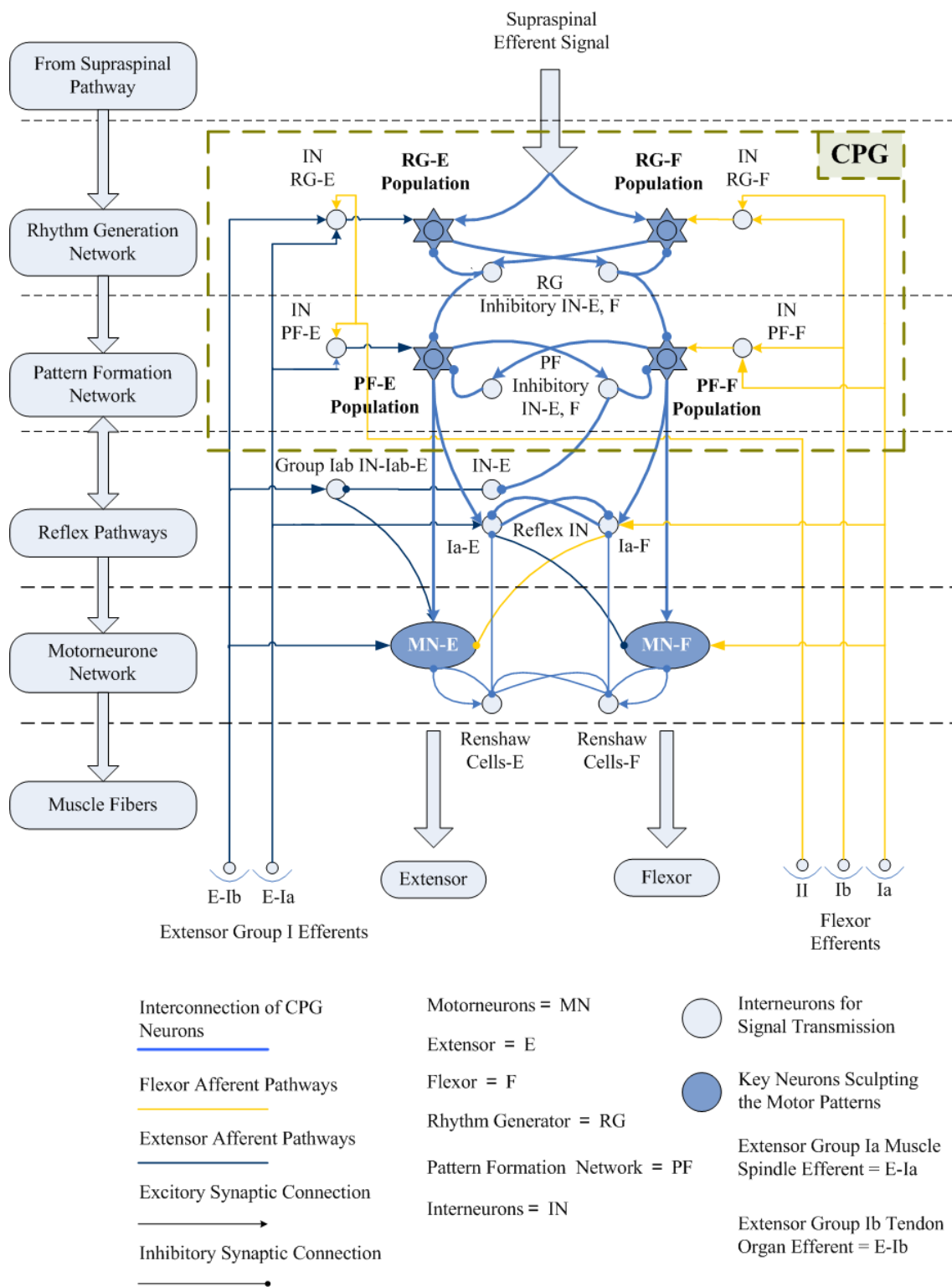


Figure 2: Structure of CPG network and the embedded reflex pathways [5, 15, 16]

healthy subjects and individuals with spinal cord injury (SCI) showed that load receptor inputs and hip joint afferents essentially contribute to the activation patterns of leg muscles during human locomotion [12, 16]. For a thorough review about the sensorimotor interaction in locomotion, see [15]. Some of the well understood reflex pathways are shown

in Fig. 2 to illustrate how the CPG and reflex circuits interact with each other to modulate the motoneuron activation pattern. The group I extensor afferents (Ia muscle spindles, Ib golgi tendon organ) contribute to the body-weight support during stance and control the stance-swing-transition by providing positive feedback loops in different levels of the network. These signals not only run through a disinaptic pathway via the interneuron IN-Iab-E, they are also fed to the CPG via the interneurons IN RG-E(F) and IN PF-E(F) for further processing. With this two level architecture of the CPG network, the group I extensor afferents can either control the proportion of extension phase via the PF network without changing the timing of the following cycles, or reset the locomotor rhythm by affecting the RG network, depending on the amplitude of stimulation. As suggested by the model in [5], the effect of Ia afferents on the RG network is weaker than their effect on the PF network. The flexor group I afferents have an analogue effect on the flexion phase, but the flexor group II is assumed to be excitatory to the extensor parts and hence forms a competing sensory signal to the flexor group I afferents. This could explain the spontaneous reversing reflex action of these afferents [5].

4 Mathematical models of single neurons

4.1 Model of pace-maker neurons

The Hodgkin-Huxley (HH) formalism enables a biophysically accurate mathematical description of single neurons. The general form of the single neuron model is given as follows:

$$C_m \cdot \frac{dV_m}{dt} = \sum_i^{n_{Ion}} I_{Ion,i} + \sum_i^{n_{Syn}} I_{Syn,i} + \sum_i^{n_{Ext}} I_{Ext,i} \quad (1)$$

where $C_m = 1\mu F/cm^2$ is the membrane capacity, and V_m the membrane potential.

Each external drive $I_{Ext,i}$ can be a constant current or a simple function like sine or ramp. The synaptic input I_{Syn} is the linear combination of all the event-triggered currents delivered by the connected n_{Syn} source neurons. The non-N-Methyl-D-Aspartic Acid (NMDA) glutamatergic synaptic current can be described as

$$I_{Syn,i} = \bar{g}_{Syn,i} \cdot w_i \cdot \exp\left(\frac{-t - t_{ls,i}}{\tau_{Syn,i}}\right) \cdot (V_m - E_{Syn,i}) \quad (2)$$

where $\bar{g}_{Syn,i} = 0.05mS/cm^2$ is the maximum conductance of synapses, w_i the connection weight, $t_{ls,i}$ is the last spike time of the i th neuron and $\tau_{Syn,i} = 5ms$ the time constant of the synaptic current. $E_{Syn,i}$, which is $-10V$ for excitatory synapses and $-40V$ for inhibitory synapses, is the reversal potential of synapses. For details of other more complicated models such as the alpha function form and the dual exponential form, see [2].

The characteristic behavior of a neuron is largely determined by the particular combination of ionic conductances $I_{Ion,i}$ found in its cell membrane and the interaction between them. When the basic Hodgkin-Huxley model is equipped with additional ion currents, which are linear superposition to the three ion channels of the classical model (sodium: I_{Na} , potassium: I_K , leakage: I_L), a much broader repertoire of behavior, such as the bursting property of pace-maker neurons in the CPG network, can be obtained. While the intrinsic mechanisms defining the rhythmogenic properties of CPG neurons remain

unknown, there is indirect evidence for the role of persistent sodium current I_{NaP} in rhythmogenesis [17, 18]. Together with the sodium, potassium, and leakage current in the original HH model, the ionic current in the pace-maker neuron consists of the following parts:

$$I_{Ion,Bursting} = I_{Na} + I_K + I_L + I_{NaP} \quad (3)$$

$$I_{Na} = g_{Na,max} \cdot m_{Na}^3 \cdot h_{Na} \cdot (V_m - E_{Na}) \quad (4)$$

$$I_K = g_{K,max} \cdot n_K^4 \cdot (V_m - E_K) \quad (5)$$

$$I_L = g_L(V_m - E_L) \quad (6)$$

$$I_{NaP} = g_{NaP,max} \cdot m_{NaP} \cdot h_{NaP} \cdot (V_m - E_{Na}) \quad (7)$$

where the constants E_x ($E_{Na} = 55mV$; $E_K = -85mV$; $E_L = -55mV$) are the reversal potentials of the ion currents, and g_x ($g_{Na} = 120mS/cm^2$; $g_K = 48mS/cm^2$; $g_{NaP} = 13mS/cm^2$; $g_L = 3mS/cm^2$) the maximum ionic conductances. The dynamics of the activation variables m_{Na} , n_K , m_{NaP} and inactivation variables h_{Na} , h_{NaP} all share a similar form derived from the original Hodgkin-Huxley formalism. Here the simplification is done by considering the fact that the activation variable m_{NaP} occurs on a much slower time scale than that of m_{Na} , so that the dynamics of (in-)activation variables represented generally by x is simplified to

$$\frac{dx}{dt} = \frac{x_\infty(V) - x}{\tau_x(V)}, x \in \{m_{Na}, h_{Na}, n_K, m_{NaP}, h_{NaP}\} \quad (8)$$

$$x_\infty(V) = \frac{1}{1 + \exp\left(\frac{V_m - \theta_x}{\sigma_x}\right)} \quad (9)$$

$$\tau_x(V) = \frac{\bar{\tau}_x}{\exp\left(\frac{V_m - \lambda_x}{\alpha_x}\right) + \exp\left(-\frac{V_m - \lambda_x}{\beta_x}\right)} \quad (10)$$

where $x_\infty(V)$ is the end value of the (in-)activation variables and $\tau_x(V)$ are the voltage dependent time constants. The other terms $\alpha_x, \beta_x, \lambda_x, \sigma_x, \bar{\tau}_x, \theta_x$ are constants selected to produce time curves of the ionic conductances similar to experimental observations, see [19].

In this model, the fast sodium current I_{Na} with its small time constant is in charge of the stiff rising edge of action potential, while the delayed-rectified potassium current I_K brings the membrane voltage back to the resting status shortly after the firing. The leakage current I_L , with g_L being a constant, is a static current accounting for the effect of other unspecified ion types. The maximum conductance of the persistent sodium channel $g_{NaP,max}$ set its weight on the total neuron behavior. If it remains under a certain threshold value, bursting and beating properties can no longer be obtained. The activation variable m_{NaP} works in a similar way to that of the fast sodium current. They are both responsible for the triggering of the action potential and the onset of the bursting activities. On the other hand, the inactivation h_{NaP} with its very large time constant $\tau_{h_{NaP}}(V)$ is in charge of the termination of the bursting pattern and thus determines the bursting period. By altering the leakage reversal potential E_L , typically from $-64mV$ to about $-50mV$, the neuron undergoes a transition from the quiescent state, through burster, to the beater status. Therefore, bursting is actually a gradual transitional state lying between these two ends and we can adjust the bursting ratio of the neuron by altering E_L . For a detailed discussion of the properties and interaction of the individual ionic currents, see [20].

4.2 Compartmental model of motoneurons

With help of experimental studies using ion channel blockers and neurotransmitters, complex firing patterns have been found in the vertebrate motoneurons. The bistable firing, which results from the interplay between several kinds of Ca^{2+} dependent currents, enables the motoneurons to convert short-lasting synaptic inputs into long-lasting motor output. Here we accept and use the model presented in [19].

The ion currents in soma include

$$I_{Ion,Moto,Soma} = I_{Na} + I_K + I_L + I_{CaN} + I_{K(Ca)} \quad (11)$$

$$I_{CaN} = g_{CaN,max} \cdot m_{CaN}^2 \cdot h_{CaN} \cdot (V_m - E_{Ca}) \quad (12)$$

$$I_{K(Ca)} = g_{K(Ca),max} \cdot m_{K(Ca)} \cdot (V_m - E_K) \quad (13)$$

$$g_{K(Ca),max} = \frac{Ca}{Ca + K_d} \quad (14)$$

$$\frac{dCa}{dt} = f(-\alpha \cdot I_{Ca} - k_{Ca} \cdot Ca) \quad (15)$$

and the dendrite is modeled with

$$I_{Ion,Moto,Dend} = I_{CaN} + I_{K(Ca)} + I_{CaL} \quad (16)$$

$$I_{CaL} = g_{CaL,max} \cdot m_{CaL} \cdot (V_m - E_{Ca}) \quad (17)$$

This two-compartment model was developed to reproduce some of the biophysical mechanisms underlying the Ca^{2+} -dependent regenerative responses observed in turtle spinal motoneurons under the injection of ion blockers. The constants g_L ($0.51mS/cm^2$), g_{CaL} ($0.33mS/cm^2$), g_{CaN} ($14mS/cm^2$ in soma, $0.3mS/cm^2$ in dendrite) and $g_{K(Ca),max}$ ($1.1mS/cm^2$ in soma, $5mS/cm^2$ in dendrite) are the newly added ion conductances of the L-like calcium conductance I_{CaL} , N-like calcium conductance I_{CaN} , and the calcium-dependent potassium current $I_{K(Ca)}$. Some of the parameters in Sec. 4.1 are changed correspondingly: $E_L = -65mV$, $g_K = 100mS/cm^2$, $g_L = 0.51mS/cm^2$. The activation and inactivation variables m_{CaN} , h_{CaN} , m_{CaL} follow the dynamics in accordance with the general form given in (8) to (10). The high threshold, inactivating current I_{CaN} allows calcium influx during action potentials and generates Ca^{2+} -based spikes in response to depolarizing current steps. The calcium-dependent potassium current $I_{K(Ca)}$ contributes to the slow after-hyperpolarization (AHP) following a spike. Its activation $m_{K(Ca)}$ is determined by the intracellular calcium concentration with (14), where $K_d = 0.2\mu M$ defines the half saturation level of this conductance. The intracellular calcium concentration Ca follows the dynamics in (15), where $f = 0.01$ describes the proportion of free Ca^{2+} , $\alpha = 0.0009mol \cdot \mu m/C$ converts the calcium current I_{Ca} into Ca^{2+} concentration and $k_{Ca} = 2ms^{-1}$ represents the Ca^{2+} removal rates.

4.3 Simplified model of spiking neurons

In order to overcome the prohibitive computational effort in large population simulations, the Hodgkin-Huxley type neural models can be reduced to a two-dimensional system of ordinary differential equations by using bifurcation methodologies. Izhikevich presented

a model which is mathematically elegant and yet retains some of the salient features of the biological realistic firing patterns in the form:

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I \quad (18)$$

$$\frac{du}{dt} = a(bv - u) \quad (19)$$

with a resetting paradigm of

$$\text{if } v \geq 30mV, \text{ then } \begin{cases} v = c \\ u = u + d \end{cases} \quad (20)$$

v and u are dimensionless variables representing the membrane potential and the membrane recovery rate originating from the activation of potassium current and inactivation of the sodium current in the Hodgkin-Huxley model. By varying the parameters a , b , c and d , a rich repertoire of firing patterns can be obtained with this model. The normal spiking state of interneurons corresponds to $a = 0.02$; $b = 0.2$; $c = -65$; $d = 2$; $I = 0$. For a detailed discussion on parametrization and the resulting firing behaviors, refer to [21].

5 Simulative evaluation

5.1 Modeling strategy for large-scale network simulation

The Hodgkin-Huxley model has the advantage of being able to replicate rich firing patterns as observed in the biological world and having a structure that can be directly derived from their biological counterparts, while the simplified model for spiking neurons is superior in terms of computational efficiency. To combine the strength of both approaches, we employ a hybrid model in our simulation. Since interneurons are only responsible for signal transmission, we use the Izhikevich model to describe them, so that a larger population with randomly distributed parameters and connection weights can be implemented. As to the pace-maker neurons and motoneurons, the HH model is utilized to produce a more biological realistic firing pattern.

5.2 Simulation results

We focus on the interaction of coupled pace-maker neurons in the CPG network and the resulting rhythmogenic properties of this topology. The low-level reflex circuits shown in Fig. 2 are not included at this stage. The model is implemented in software package NEURON [22]. Differential equations of the HH-style pace-maker neurons and motoneurons are solved using the *cnexp* method provided by NEURON. Other interneurons modeled with the simplified model for spiking neurons are solved with the *derivimplicit* method due to their nonlinear property.

The pace-maker neurons and motoneurons are modeled as concentrated elements to represent the corresponding neuron populations. Interneurons within the CPG network have a population of ten neurons. Heterogeneity is introduced by adding randomly distributed parameters for synaptic connection weights and delays. Simulation parameters are based on [1, 19, 21, 23] and adjusted to produce reasonable results.

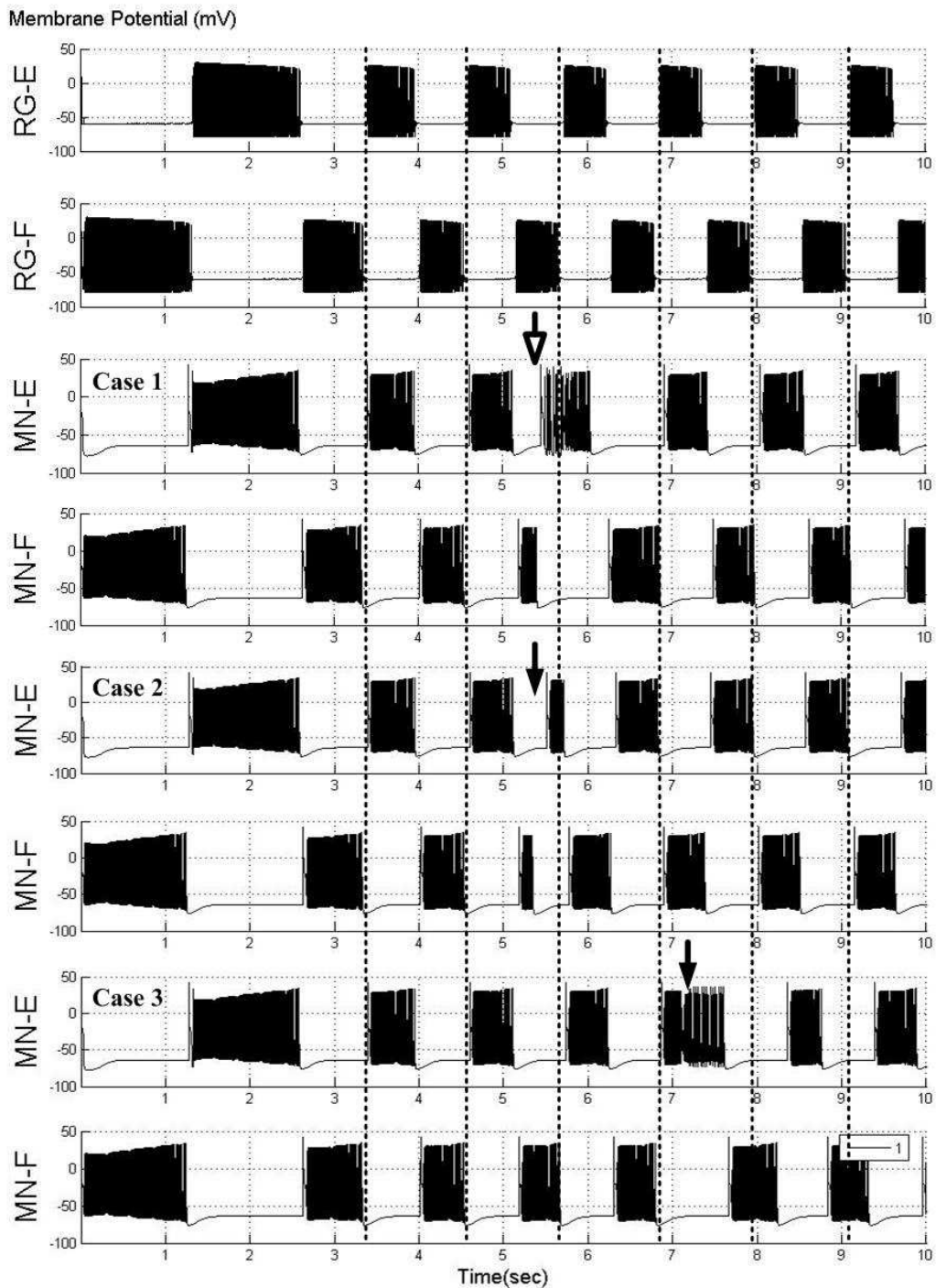


Figure 3: Modeling the effects of group Ia extensor afferent stimulation (see Fig. 2 for the different neuron types)

A short pulse with duration of 10ms is delivered to either side of the rhythm generation network to avoid the resonant state of the network resulting from a symmetrical initial state. The whole network is able to synchronize automatically without external inputs and finally settle down to the bursting frequency set by the rhythm generation neurons RG-E and RG-F. The alternating firing pattern of the flexor and extensor motoneurons

can represent the synergic movements of the corresponding muscles during stance and swing phase. The simulation result corresponds to a step cycle period of about 1.1s, see the output of pace-maker neurons in rhythm generation network (RG-E, RG-F) at top of Fig. 3.

To study how the rhythm generation network and the pattern formation network contribute to the sculpting of the output activation patterns under group Ia afferent inputs, we apply stimulations to different levels of the CPG with different timings. At the instants indicated by the arrows in Fig. 3, an afferent stimulus with a duration of 500ms is applied to the CPG. The hollow arrow indicates a stimulus to the PF-level via the interneuron PF-Ia-E, while solid arrows represent a strong stimulus influencing both the RG and PF levels. The output of the network in the extensor and flexor motoneurons MN-E and MN-F are shown in Fig. 3, aligned to the intrinsic rhythmic pattern set by the RG network. In Case 1, the Ia afferent causes a premature flexion without changing the timing of the following cycles. Conversely, a strong extensor Ia afferent can reset the locomotion rhythm by affecting the rhythm generation network via the sensory neuron RG-Ia-E. Notice the reversed phase in Case 2 after the external stimulation. On the other hand, Ia afferent can induce a prolongation of the extension phase, when the extensor motoneuron is activated (Case 3). This simulates the scenario that the transition from stance to swing phase can be delayed in prolonged steady load conditions. By this, the Ia afferent pathway and the CPG network actively contribute to the body-weight support during gait.

5.3 Discussion

This computational model of CPG network consisting of biological realistic neurons is able to generate coordinated rhythmic neural activation patterns through intrinsic mechanisms, and sculpts the outputs under the modulation of sensory information. We consider this model as a basis for subsequent modeling studies of reflex pathways and correlations between muscle activation patterns and locomotion movements.

A further development of this model is confronted with manifold challenges. The computational capacities of current simulation tools would soon reach their limit if network level behavior is under investigation (the current simulation needs about 30 min. to simulate 10 sec. of network behavior). Since this model is sensitive to subtle parameter changes and possess a high dimensional space of strong interrelated parameters, without analytical methods for the analysis of complex spiking neuron networks, application of optimization algorithms in parameter tuning would be a difficult issue.

6 Work in progress: modeling and evaluation of CPG structures

Future application perspectives of our model include: (1) as neural controller and locomotion pattern generator for bipedal-walking of humanoid robots; (2) as internal model for analysis of afferent and efferent modulations on human locomotion patterns; (3) based on the preceding item, as human-computer-interface in neuroprosthetic applications. One of the milestones on the way to the successful implementation of our model is the understanding of the synergetic interactions between the individual parts in the motor control system (Sec. 2). Or in other words, the linkage between motoneuron firing patterns and muscle contraction in rhythmic behaviors.

6.1 Software solutions

For the fulfillment of our research goal, we propose a simulation paradigm for closed-loop systems of locomotor control (Fig. 4). With respect to the simulation of biological realistic spiking neurons and their networks (see [24] for a comprehensive review), which serves as controller and coordinator in the whole system, software specialized for this application (e.g. NEURON) outperforms the general purpose simulation programs. They provide tailor-made numerical integration methods and prefabricated functions and classes describing various neural mechanisms and entities. SIMM is a commercially available software package for simulation and visualization of musculoskeletal models. MMS (Musculoskeletal Modeling in Simulink) [25] can translate the models developed in SIMM into Simulink blocks. The transformation from neural activation signal to muscle forces can be done by Virtual Muscle [26]. For visualization and analysis of neural spike trains and muscle EMG signals serves the Gait-CAD toolbox [27]. Human data can be incorporated into the closed loop system for parameter training and evaluation of simulation results. A related data set is discussed in the next section.

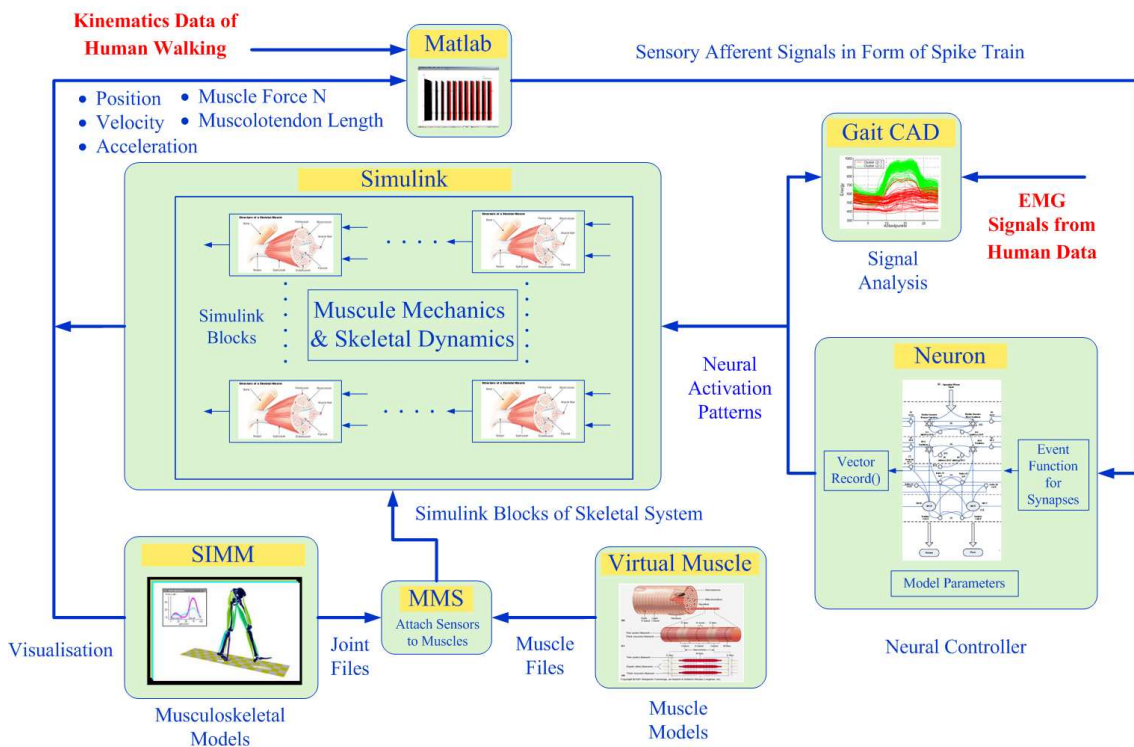


Figure 4: Possible simulation strategy for closed-loop systems of locomotor control

6.2 Human data

Data of human walking has been recorded from a 25-year-old female subject who walks on a treadmill with a speed of 1.1 m/s in the locomotion lab at the Orthopädische Universitätsklinik Heidelberg. Kinematic data which includes the joint angles of foot, knee, hip, pelvis, trunk, shoulder, and elbow in the sagittal, transversal and frontal planes is recorded with a commercially available 3D motion analysis system developed by Motion Analysis

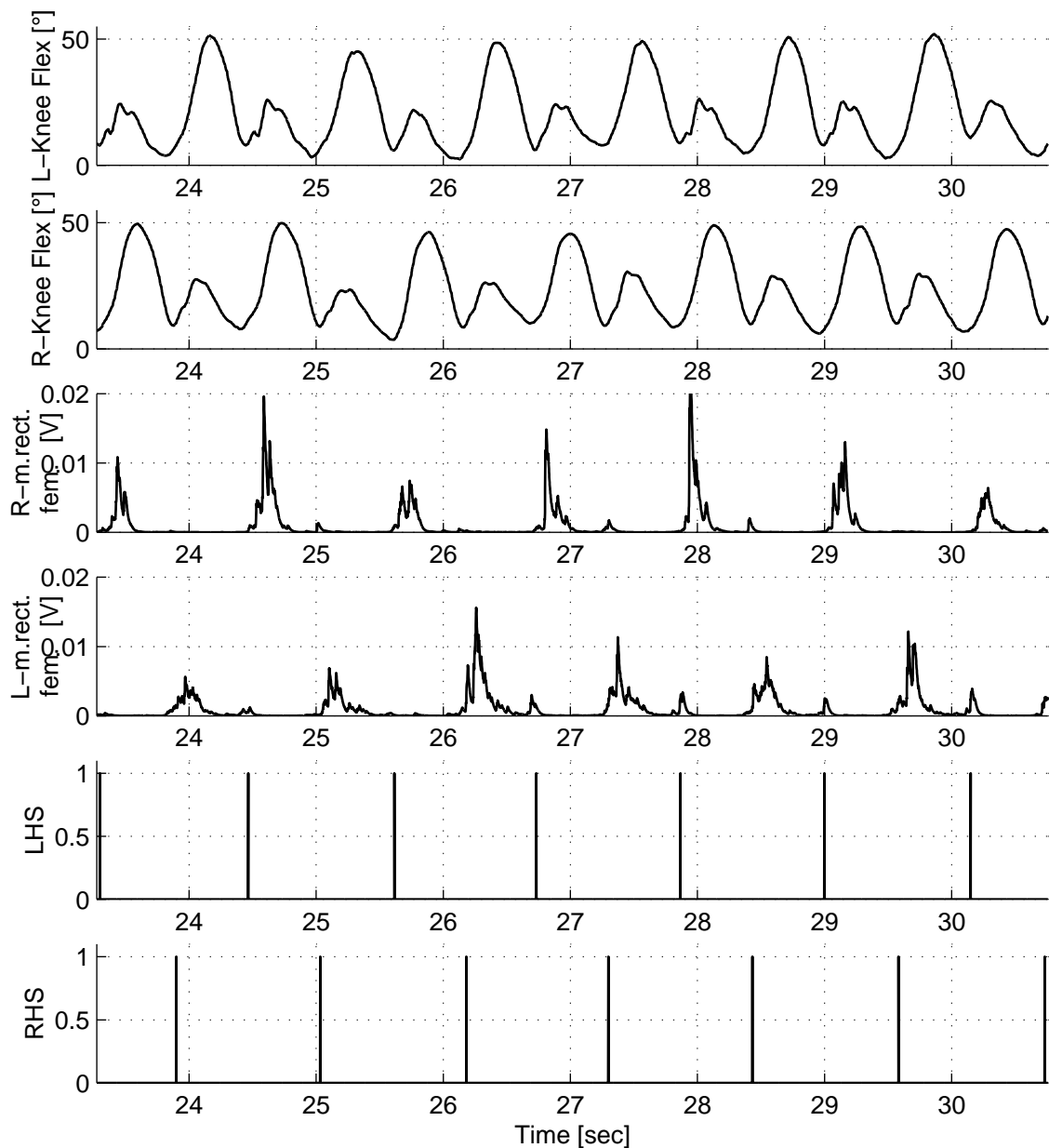


Figure 5: Examples of gait data: joint angles of the left and right knee in the sagittal plane, electromyographic signal of the left and right m. rectus femoris, and detected gait events for the left and right heel strike (top down)

Corp., USA with a sampling rate of 60 Hz. The hardware platform is described in [28]. In parallel to the kinematic data, EMG signals from the following muscles of both legs are recorded with a sampling rate of 960 Hz: m. gastrocnemius, m. tibialis anterior, m. biceps femoris, m. rectus femoris.

The following data preprocessing procedures are carried out:

- Gait events like heel strike (LHS, RHS) and toe off (LTO, RTO) at left and right side are detected.
- Joint angles are resampled at 960 Hz and low-pass filtered using a 1st order Infinite Input Response (IIR) filter with $a = 0.9$.
- Electromyographic signals are band-pass filtered using a 5th order Butterworth filter

with edge frequencies of 10 Hz and 350 Hz. The signals are further rectified and low-pass filtered by a 1st order IIR filter with $a = 0.95$.

Some selected results are presented in Fig. 5.

This human walking data provides information about gait events and triggering signals, which can be used to tune the parameters of the CPG network for the generation of more realistic muscle activation patterns. Furthermore, systematic analysis of differences between the recorded and the estimated muscle activations can be carried out using the methods from [29]. We also plan to obtain data of different walking speeds from the same subject and use them to analyze the influence of walking speed and temporal speed gradient on muscle activation patterns.

7 Conclusion

In this paper we gave an overview about the research done to identify the neurological structures which are responsible for the generation of rhythmic activation patterns for locomotion. Then the structure and components of these central pattern generators were elucidated and a simulation model, which has been implemented using NEURON, was introduced. This model will be used for further evaluations of human EMG data and locomotion, based on which it will be adapted for further uses in research for neuroprostheses and in the field of humanoid robotics.

Acknowledgements: This work has been performed within the framework of the German Humanoid Robotics Program (SFB 588) funded by the German Research Foundation (*DFG: Deutsche Forschungsgemeinschaft*). In particular, we would like to thank Christian Schuld for the preparation of the EMG data.

References

- [1] Rybak, I.; Shevtsova, N.; Lafreniere-Roula, M.; McCrea, D.: Modelling Spinal Circuitry Involved in Locomotor Pattern Generation: Insights from Deletions During Fictive Locomotion. *Journal of Physiology* (2006).
- [2] Bower, J.; Beeman, D.: *The Book of GENESIS*. Free Internet Edition. 2003.
- [3] Gordon, I.; Whelan, P.: Deciphering the Organization and Modulation of Spinal Locomotor Central Pattern Generators. *The Journal of Experimental Biology* 209 (2007), pp. 2007–2014.
- [4] Burke, R.; Degtyarenko, A.; Simon, E.: Patterns of Locomotor Drive to Motoneurons and Last-Order Interneurons: Clues to the Structure of the CPG. *Journal of Neurophysiology* 86(1) (2001), pp. 447–462.
- [5] Rybak, I.; Stecina, K.; Shevtsova, N.; McCrea, D.: Modelling Spinal Circuitry Involved in Locomotor Pattern Generation: Insights from the Effects of Afferent Stimulation. *Journal of Physiology* 577.2 (2006), pp. 641–658.
- [6] Ogihara, N.; Yamazaki, N.: Generation of Human Bipedal Locomotion by a Bio-Mimetic Neuro-Musculo-Skeletal Model. *Biological Cybernetics* 84 (2001), pp. 1 – 11.
- [7] Righetti, L.; Ijspeert, A.: Design methodologies for central pattern generators: an application to crawling humanoids. In: *Proceedings of Robotics: Science and Systems*, pp. 191–198. Philadelphia, USA. 2006.
- [8] Sherwood, L.: *Human Physiology: From Cells to Systems*. Thomson Books/Cole. 2004.
- [9] Schmidt, R.; Thews, G.; Lang, F.: *Physiologie des Menschen*. Berlin: Springer. 2000.
- [10] Cohen, A.; Boothe, D.: Sensorimotor Interactions During Locomotion: Principles Derived from Biological Systems. *Autonomous Robots* 7 (1999), pp. 239–245.

- [11] Ivanenko, Y.; Poppele, R.; Lacquaniti, F.: Spinal Cord Maps of Spatiotemporal Alpha-Motoneuron Activation in Humans Walking at Different Speeds. *Journal of Neurophysiology* 95 (2006), pp. 602–618.
- [12] Dietz, V.; Harkema, S. J.: Locomotor Activity in Spinal Cord-Injured Persons. *Application Physiology* 96 (2004), pp. 1954–1960.
- [13] Hultborn, H.; Nielsen, J.: Spinal Control of Locomotion - from Cat to Man. *Acta Physiology* 189 (2007), pp. 111–121.
- [14] Duysens, J.; de Crommert, H. W. A. A. V.: Neural Control of Locomotion; Part 1: The Central Pattern Generator from Cats to Humans. *Gait & Posture* 7(2) (1998), pp. 131–141.
- [15] Rossignol, S.; Dubuc, R.; Gossard, J.: Dynamic Sensorimotor Interactions in Locomotion. *Physiological Reviews* 86 (2006), pp. 89–154.
- [16] McCrea, D.: Spinal Circuitry of Sensorimotor Control of Locomotion. *Journal of Physiology* .
- [17] Crill, W.: Persistent Sodium Current in Mammalian Central Neurons. *Annual Reviews of Physiology* 58 (1996), pp. 349–362.
- [18] Del Negro, C.; Koshiya, N.; Butera, R.; Smith, J.: Persistent Sodium Current, Membrane Properties and Bursting Behavior of Pre-Boetzinger Complex Inspiratory Neurons in Vitro. *Journal of Physiology* 88 (2002), pp. 2242 – 2250.
- [19] Booth, V.; Rinzal, J.; Kiehn, O.: Compartmental Model of Vertebrate Motoneurons for Ca²⁺-Dependent Spiking and Plateau Potentials under Pharmacological Treatment. *Journal of Neurophysiology* 78 (1997), pp. 3371 – 3385.
- [20] Malmivuo, J.; Plonsey, R.: *Bioelectromagnetism - Principles and Applications of Bioelectric and Biomagnetic Fields*. Oxford University Press. 1995.
- [21] Izhikevich, E.: Simple Model of Spiking Neurons. *IEEE Transaction on Neural Networks* 14 (2003) 6, pp. 1569 – 1572.
- [22] Carnevale, N.; Hines, M.: *The NEURON Book*. Cambridge University Press. 2006.
- [23] Butera, R.; Rinzal, J.; Smith, J.: Models of Respiratory Rhythm Generation in Pre-Boetzinger Complex. I. Bursting Pacemaker Neurons. *Journal of Physiology* 82 (1999), pp. 382 – 397.
- [24] Brette, R.; Rudolph, M.; Carnevale, T.; Hines, M.; Beeman, D.; Bower, J.; Diesmann, M.; Goodman, P.; Harris, F.; Jr. Zirpe, M.; Natschlaeger, Pecevski, D.; Ermeuntrout, B.; Djurfeldt, M.; Lansner, A.; Rochel, O.; Vieville, T.; Muller, E.; Davison, A.; Boustani, S.; Destexhe, A.: Simulation of Networks of Spiking Neuron: A Review of Tools and Strategies. *Journal of Computational Neuroscience* in press (2007), pp. 1 – 66.
- [25] Davoodi, R.: *User's Guide to MMS: Musculoskeletal Modeling in Simulink*. Alfred E. Mann Institute, University of Southern California. 2002.
- [26] Cheng, E.; Brown, I.; Jerry, L.: *Virtual Muscle 3.1.5 User's Manual*. Alfred E. Mann Institute, University of Southern California. 2001.
- [27] Mikut, R.; Burmeister, O.; Reischl, M.; Loose, T.: Die MATLAB-Toolbox Gait-CAD. In: *Proc., 16. Workshop Computational Intelligence*, pp. 114–124. Universitätsverlag Karlsruhe. 2006.
- [28] Schablowski-Trautmann, M.: *Konzept zur Analyse der Lokomotion auf dem Laufband bei inkompletter Querschnittlähmung mit Verfahren der nichtlinearen Dynamik*. Ph.D. thesis, Universität Karlsruhe, Universitätsverlag Karlsruhe. 2006.
- [29] Wolf, S.; Loose, T.; Schablowski, M.; Döderlein, L.; Rupp, R.; Gerner, H. J.; Bretthauer, G.; Mikut, R.: Automated Feature Assessment in Instrumented Gait Analysis. *Gait & Posture* 23(3) (2006), pp. 331–338.

Multikriterielle Optimierung mit modellgestützten Evolutionsstrategien

Frank Hoffmann, Jan Braun, Torsten Bertram

Lehrstuhl für Regelungssystemtechnik, Universität Dortmund
E-Mail: {frank.hoffmann}{jan.braun}{torsten.bertram}@uni-dortmund.de

Sebastian Hölemann

Institut für Regelungstechnik, RWTH Aachen
E-Mail: s.hoelemann@irt.rwth-aachen.de

1 Einleitung

Evolutionäre Algorithmen finden bei komplexen, hochdimensionalen Optimierungsproblemen hinreichend gute Lösungen, benötigen dafür jedoch wegen der partiell zufälligen Erkundung des Optimierungsraums durch eine Population im Allgemeinen eine große Anzahl von Fitnessbewertungen. Evolutionsstrategien werden unter anderem bei der automatisierten Hardware-in-the-Loop Optimierung von Regelungssystemen eingesetzt. Dabei kann eine einzige Fitnessbewertung „mehrere Sekunden“ dauern. Dieser Zeitaufwand schränkt die praktische Anwendbarkeit von evolutionären Algorithmen zur Optimierung realer Systeme ein.

1.1 Motivation

Da die evolutionäre Optimierung hochdimensionaler Probleme eine Vielzahl von Berechnungen der Fitnessfunktion erfordert, sind zahlreiche Untersuchungen zum Einsatz von Modellen der Fitnessfunktion unternommen worden (vergl. [1]). Auswertungen von Fitnessmodellen anstelle der wahren Fitnessfunktion selbst können den Optimierungsprozess signifikant beschleunigen, sofern das Modell deutlich schneller berechnet werden kann als die Fitnessfunktion. Fitnessmodelle werden auf der Basis bereits real ausgewerteter Punkte (Datenpunkte) gebildet.

Es ist jedoch nahezu unmöglich, ein Modell der wahren Fitnessfunktion zu erhalten, das im gesamten Suchraum gute Approximationen liefert. Aufgrund der hohen Dimensionalität und der damit verbundenen Komplexität vieler Optimierungsprobleme ist die Genauigkeit begrenzt, mit der ein aus einer begrenzten Anzahl von Trainingsdaten gebildetes Modell die Fitnessfunktion approximieren kann. Fitnessmodelle haben zumeist hohe Approximationsfehler und können im Modell irreführende Minima erzeugen, an Stellen an denen die reale Funktion selber kein Minimum aufweist. Wird im Evolutionsprozess ausschließlich ein Modell anstelle von Fitnessbewertungen verwendet, konvergiert der Algorithmus mit einer sehr hohen Wahrscheinlichkeit gegen ein falsches Minimum, wie in [2] am Beispiel von zwei Testfunktionen und einem neuronalen Netz als Approximationsmodell gezeigt wird.

Die Ergebnisse in [2] machen deutlich, dass auf Berechnungen der wahren Fitnessfunktion innerhalb des Evolutionsprozesses nicht verzichtet werden kann, um eine falsche Konvergenz der Evolutionsstrategie zu vermeiden. Daher wird ein Modellmanagement

zur Steuerung erforderlich, das bestimmt welche der im Verlauf der Evolution erzeugten Individuen tatsächlich bewertet werden müssen und bei welchen eine Fitnessschätzung ausreicht. Im folgenden Abschnitt werden verschiedene Strategien aus der Literatur vorgestellt.

1.2 Modellmanagement

Beim Modellmanagement werden zwei Ansätze unterschieden (vergl. [1], [2]), um alle oder nur einen Teil der Individuen in allen oder einzelnen Generationen mit der realen Fitnessfunktion zu bewerten. Bei generationsbasierten Strategien wird jeweils über g_1 Generationen ausschließlich das Modell zur Bewertung aller Individuen verwendet. In den darauf folgenden g_2 Generationen werden nun alle Individuen mittels der realen Fitnessfunktion bewertet. Mit Hilfe der in diesen „kontrollierten Generationen“ aus den realen Funktionsauswertungen gewonnenen neuen Daten wird das Fitnessmodell trainiert. Es wird empfohlen, die Frequenz der kontrollierten Generationen adaptiv mit einer ermittelten Modellgüte zu steuern, um in den nicht kontrollierten Generationen die Gefahr der falschen Konvergenz im Falle eines schlechten Modells zu verringern (siehe [2]).

Im Gegensatz zur generationsbasierten Methode werden bei individuenbasierten Ansätzen in jeder Generation eine bestimmte Anzahl von Individuen mit der Fitnessfunktion bewertet. Für die Auswahl dieser so genannten „kontrollierten Individuen“ werden in [3] die folgenden Strategien vorgestellt und miteinander verglichen:

- **Best Strategy**

Bei dieser Strategie werden λ Nachkommenindividuen von den μ Eltern abgeleitet und ihre Fitness mit einem Approximationsmodell geschätzt. Anschließend werden die $\lambda^* < \lambda$ Individuen mit der besten geschätzten Fitness durch die wahre Fitnessfunktion bewertet. Unter Ausnutzung der durch die realen Fitnessbewertungen zusätzlich erhaltenen Trainingsdaten wird die Fitness der $\lambda - \lambda^*$ übrigen Individuen erneut geschätzt. Die Strategie wählt nun aus der Menge aller λ Individuen gemäß ihrer realen bzw. geschätzten Fitness die μ besten Individuen als Eltern der folgenden Generation aus.

- **Clustering Technique**

Es werden λ Nachkommen erzeugt und in $\lambda^* < \lambda$ Cluster eingeteilt. In jedem Cluster wird dasjenige Individuum tatsächlich bewertet, das am nächsten zum Zentrum des jeweiligen Clusters liegt. Entsprechend werden λ^* reale Fitnessauswertungen durchgeführt. Die übrigen $\lambda - \lambda^*$ nicht real bewerteten Individuen werden durch ein Fitnessmodell geschätzt und aus der Menge aller λ Individuen werden die μ besten ausgewählt.

- **Clustering Technique with Best Strategy**

Diese Methode ist eine Erweiterung der zuvor genannten Vorgehensweise. Eine Menge von λ Nachkommen wird erzeugt und mittels eines Fitnessmodells bewertet. Alle λ Individuen werden in $\lambda^* < \lambda$ Cluster eingeteilt. Im Gegensatz zur vorher beschriebenen Cluster Methode wird hier jeweils das gemäß der geschätzten Fitness beste Individuum eines Clusters mit der Fitnessfunktion bewertet.

- **Pre-Selection**

Im Gegensatz zu den bisher genannten Strategien findet bei der Methode der *Vorselektion* die Selektion der μ Individuen für die kommende Elterngeneration ausschließlich aus einer Menge von real bewerteten Nachkommen statt. Eine Konvergenz gegen falsche Minima des Modells ist hier also nicht möglich. Da nach Vergleich aller genannten individuenbasierten Verfahren in [3] die Vorselektionsmethode als die aussichtsreichste herausgestellt wird, bildet sie auch die Grundlage der nachfolgenden Untersuchungen und wird in Abschnitt 2.1 eingehend erläutert.

2 Fitnessmodellierung

2.1 Evolutionsstrategie mit Vorselektion

Inspiziert durch die biologische Evolution, arbeitet die Evolutionsstrategie mit einer Population von *Individuen*. Jedes Individuum repräsentiert mit seinem reellwertigen Parametersatz \mathbf{x} eine potentielle Lösung des globalen Optimierungsproblems. Beginnend mit einer Startpopulation, bestehend aus μ Individuen, werden in jeder Generation mit Hilfe eines *Rekombinationsmechanismus* zunächst λ Nachkommen aus der Population abgeleitet. Die Nachkommen werden anschließend einem *Mutationsprozess* unterworfen. Im Falle einer Standard Evolutionsstrategie (ES) wird nun für alle λ Nachkommen die Zielfunktion an der entsprechenden Stelle \mathbf{x} berechnet und ihr Funktionswert $f(\mathbf{x})$ dem jeweiligen Individuum zugeordnet. Im Anschluss wählt ein *Selektionsverfahren* aus der Menge der Nachkommenindividuen ((μ, λ) -Strategie) bzw. aus der Menge der Eltern- und Nachkommenindividuen ($(\mu + \lambda)$ -Strategie) die gemäß ihrer Fitnessbewertung μ besten Nachkommen aus, um die Population für die folgende Generation zu bilden. Mögliche Abbruchkriterien der Evolutionsschleife sind beispielsweise das Erreichen einer maximalen Anzahl an Funktionsauswertungen oder einer vorgegebenen Mindestgüte. Der Algorithmus der Evolutionsstrategie ist schematisch im linken Teil von Bild 1 skizziert. Ausführliche Erläuterungen und Konvergenzbetrachtungen sind in [4] und [5] zu finden.

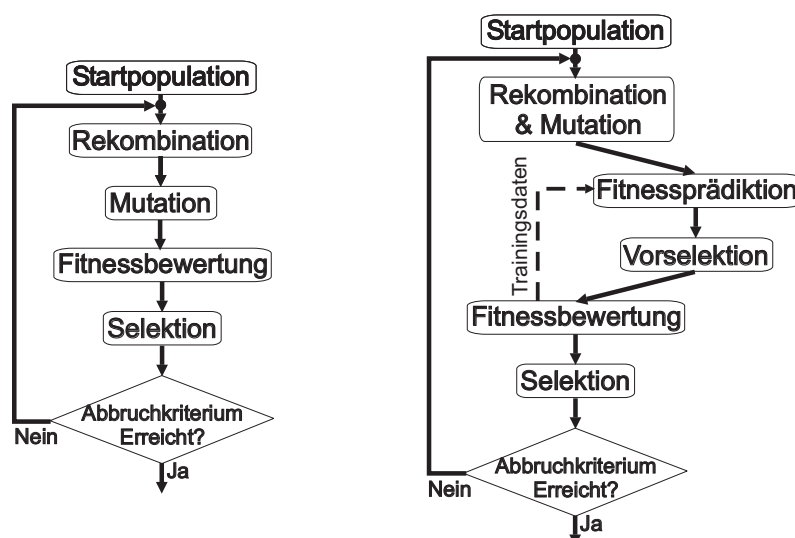


Bild 1: Evolutionsschleife der Standard ES und der MAES

Eine modellgestützte Evolutionsstrategie (MAES) entsprechend der in Abschnitt 1.2 erwähnten Methode der Vorselektion kann leicht in die Schleife der Standard ES integriert werden. Das Prinzip ist im rechten Teil von Bild 1 dargestellt. Im Gegensatz zur Standard ES werden hier nicht λ sondern eine Anzahl von $\lambda_p > \lambda$ Nachkommen mittels Rekombination und Mutation aus der Elternpopulation gebildet. Diese erhalten dann mittels eines Fitnessmodells einen Schätzwert für ihre tatsächliche Fitness. Die reale Fitnessfunktion soll nun nur für die vielversprechendsten Individuen berechnet werden. Daher wählt ein *Vorselektionsverfahren* aus den λ_p Nachkommen die gemäß ihrer Fitnessschätzung λ besten Individuen aus, um diese mit der tatsächlichen Fitnessfunktion zu bewerten. Aus den λ real bewerteten Individuen werden anschließend wiederum durch Selektion die μ Eltern der folgenden Generation bestimmt.

Wie bei der Standard ES werden auch bei der MAES in jeder Generation λ tatsächliche Funktionsauswertungen durchgeführt. Allerdings steigt mit der höheren Anzahl an erzeugten Nachkommen in jeder Generation die Wahrscheinlichkeit für das Auffinden besserer Lösungen. Der erhoffte Erfolg der MAES liegt also darin, dass die erwartete Verbesserung der Fitness des besten Individuums einer Population von einer Generation zur nachfolgenden größer ist als bei der Standard ES. Sofern also aufgrund der Modellschätzung die gemäß ihrer Fitness besten Individuen vorselektiert und damit auch selektiert werden, kann die Konvergenz der MAES gegenüber der Standard ES erheblich beschleunigt werden. Die Ergebnisse in Abschnitt 4 bestätigen diese Zusammenhänge.

2.2 Fitnessmodellierung mit lokaler linear gewichteter Regression

Die Fitnessmodellierung führt auf das Problem der Approximation der (unbekannten) realen Zielfunktion $f(\mathbf{x})$ mittels einer, im Verhältnis zur hohen Dimension n des Parameterraums, geringen Anzahl von Datenpunkten $\mathbf{x}^{(i)} \in D, i = 1, 2, \dots, k$. Um für einen Punkt $\mathbf{q} \notin D$ den Funktionswert $f(\mathbf{q})$ zu schätzen, wird in [7] das Verfahren der *lokal gewichteten Regression* (LWR) verwendet. Im Folgenden wird $f(\mathbf{x})$ in einer Umgebung von \mathbf{q} lokal durch eine lineare Funktion $\hat{f}_{\mathbf{q}}(\mathbf{x}) = \boldsymbol{\beta}^T \cdot [1 \ \mathbf{x}^T]^T$ angenähert. Die Koeffizienten der linearen Funktion werden nach [7] entsprechend Gleichung 1 berechnet. Darin hängt die Matrix \mathbf{H} nur von den Datenpunkten $\mathbf{x}^{(i)}$ ab, wobei \mathbf{y} der Vektor der realen Funktionswerte der Stützstellen und \mathbf{W} eine diagonale Gewichtungsmatrix ist.

$$\boldsymbol{\beta} = (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} \mathbf{y} \quad (1)$$

$$\mathbf{H} = \begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(k)} & x_2^{(k)} & \cdots & x_n^{(k)} \end{pmatrix} \quad (2)$$

Durch die Gewichtungsmatrix $\mathbf{W} = \text{diag}[w_1, w_2, \dots, w_k]$ in Gleichung 1 erhält das lineare Approximationsmodell einen lokalen Charakter. Die Diagonalelemente w_i sind eine Funktion des Abstandes $\Delta(\mathbf{x}^{(i)}, \mathbf{q}) = \|\mathbf{M}(\mathbf{x}^{(i)} - \mathbf{q})\|_2$ vom i -ten Datenpunkt $\mathbf{x}^{(i)}$ zum schätzenden Punkt \mathbf{q} . Dabei enthält die Diagonalmatrix \mathbf{M} Faktoren zur Skalierung jeder Dimension von \mathbf{x} . Als Gewichtungsfunktion wird im Folgenden $w(\Delta) = e^{-\Delta^2}$ verwendet. Durch die Gewichtung haben Datenpunkte mit einem kleinen Abstand zu \mathbf{q} einen

größeren Einfluss auf die Approximationsfunktion als Punkte mit einem größeren Abstand. Damit kann sich auch eine lineare Funktion in einer kleinen Umgebung von \mathbf{q} gut an den Verlauf von $f(\mathbf{x})$ anpassen und gute Approximationen liefern.

Eine spezielle Form der lokal gewichteten Regression ist der *abstandsgewichtete Mittelwert* entsprechend Gleichung 3, bei dem die Approximationsfunktion eine Konstante darstellt. Die Gewichtung der Datenpunkte und die Abstandsberechnung werden wie zuvor beschrieben durchgeführt.

$$\hat{f}(\mathbf{q}) = \frac{\sum_{i=1}^k f(\mathbf{x}^{(i)}) \cdot w(\Delta(\mathbf{x}^{(i)}, \mathbf{q}))}{\sum_{i=1}^k w(\Delta(\mathbf{x}^{(i)}, \mathbf{q}))} \quad (3)$$

Die lokale lineare Regression erweist sich bei der Vorhersage der Fitness gegenüber dem abstandsgewichteten Mittelwert aufgrund ihrer Fähigkeit zur Extrapolation der Daten als überlegen. Die Fähigkeit zur Extrapolation ist gerade für die Fitnessmodellierung von Bedeutung, da die neu generierten zu schätzenden Nachkommen oftmals am Rande der bisherigen Elternpopulationen liegen.

Den Einfluss des Fitnessmodells auf den Fortschritt der Evolution zeigt Bild 2 für die Modellierung durch die Modelle: nächster Nachbar, abstandsgewichteter Mittelwert, lineare gewichtete Regression und Gauss Prozesse. Als Benchmark für die MAES dienen vier künstliche Optimierungsprobleme: die Kugelfunktion,

$$f(\mathbf{x}) = \sum_{l=1}^n x_l^2 \quad (4)$$

die Rosenbrockfunktion,

$$f(\mathbf{x}) = \sum_{l=1}^{n-1} \left(10(x_l^2 - x_{l+1})^2 + (x_l - 1)^2 \right) \quad (5)$$

die Griewankfunktion,

$$f(\mathbf{x}) = 1 + \frac{1}{100} \sum_{l=1}^n x_l^2 - \prod_{l=1}^n \cos\left(\frac{x_l}{\sqrt{l}}\right) \quad (6)$$

und die Ackleyfunktion.

$$f(\mathbf{x}) = -20 e^{\left(-0.2 \frac{1}{n} \sqrt{\sum_{l=1}^n x_l^2}\right)} - e^{\left(\frac{1}{n} \sum_{l=1}^n \cos(2\pi x_l)\right)} + 20 + e \quad (7)$$

In allen Fällen basieren die in Bild 2 dargestellten Verläufe der Fitness für eine MAES mit konstantem $\lambda = \lambda_p/2$ auf einer Mittelung von hundert unabhängigen Läufen bei einer jeweiligen Dimension des Optimierungsraumes von $n = 20$. Von allen Modellen weisen nur die linear gewichtete Regression und Gauss-Prozesse ein über alle Probleme gleich gutes Konvergenzverhalten auf. Demgegenüber neigt der abstands-basierte Mittelwert zu einer vorzeitigen Konvergenz der evolutionären Optimierung gerade bei der Kugelfunktion.

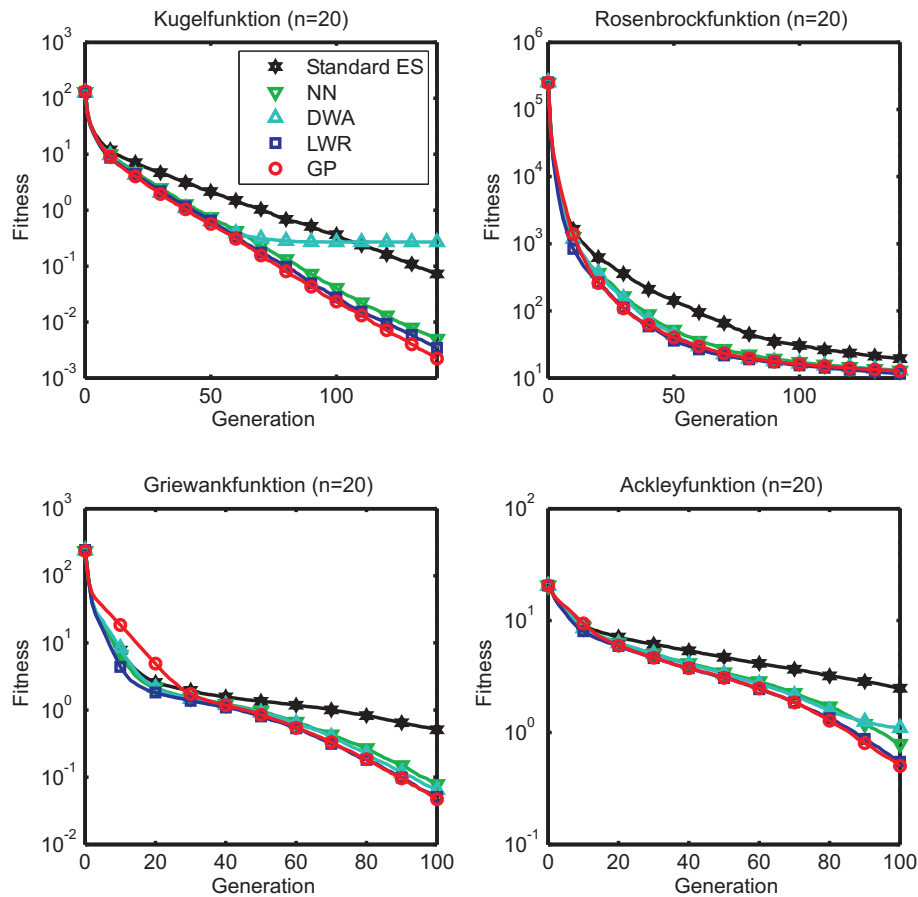


Bild 2: Entwicklung der Fitness ausgewählter Benchmarkprobleme für eine Standard ES und eine MAES, mit folgenden Modellen: nächster Nachbar (NN), abstandsgewichteter Mittelwert (DWA), linear gewichtete Regression (LWR) und Gauss-Prozesse (GP).

2.3 Modellgüte

Ein wichtiger Aspekt des Modellmanagements besteht in der Überwachung der Modellgüte. Dabei orientiert sich die Modellgüte im Gegensatz zu herkömmlicher Datenmodellierung nicht primär am mittleren Fehler zwischen tatsächlicher und vorhergesagter Fitness, sondern bemisst sich vielmehr an der Fähigkeit des Modells korrekte Selektionsentscheidungen zu treffen. Die Modellqualität bestimmt sich aus der Fähigkeit des Modells anhand der geschätzten Fitnesswerte die tatsächlich μ besten Eltern aus den λ Nachkommen zu selektieren. Dazu werden die λ Nachkommen anhand ihrer tatsächlich beobachteten Fitness absteigend sortiert und dem i -ten Individuum ein Gewicht $(\lambda - i)$ zugeordnet. Dieses Gewicht reflektiert das Ziel, die tatsächlich besten Lösungen zu finden. Die Güte des Modells ergibt sich aus der mit dem jeweiligen Rang gewichteten Anzahl anhand der Schätzung korrekt selektierter Eltern.

$$Q = \sum_{i=1}^{\mu} g_i(\lambda - i) \quad (8)$$

Wobei $g_i = 1$ falls das i -te Individuum aufgrund der Schätzung durch das Model den μ besten Lösungen zugeordnet wird, und $g_i = 0$ falls die Schätzung dieses eigentlich zu den Eltern gehörende Individuum nicht ausgewählt hat. Dieses Kriterium berücksichtigt die tatsächliche Rangfolge insofern als es die Modellgüte stärker reduziert, wenn das de facto

beste Individuum verfehlt wird, als beispielsweise das Zweitbeste. Im Idealfall, dass das Modell alle μ tatsächlich besten Lösungen findet, ergibt sich die maximale Modellgüte.

$$Q^{max} = \sum_{i=1}^{\mu} (\lambda - i) = \mu\lambda - \frac{\mu(\mu + 1)}{2} \quad (9)$$

Im Falle einer rein zufälligen Selektion ergibt sich als Erwartungswert der Modellgüte.

$$Q^{rand} = \frac{\mu^2}{\lambda} \cdot \frac{2\lambda - \mu - 1}{2} \quad (10)$$

2.4 λ -Regelung

Wie die Ergebnisse in Abschnitt 4 zeigen werden, werden bei der in Abschnitt 2.1 erläuterten MAES unnötig viele reale Funktionsauswertungen durchgeführt, die dem Evolutionsprozess keinen zusätzlichen Gewinn verschaffen. Daher verfolgt die kontrollierte modellgestützte Evolutionsstrategie (λ -CMAES) das Ziel, die Anzahl der tatsächlich bewerteten Nachkommen in einem Bereich $\lambda \in [\mu, \lambda_p]$ an die aktuelle Modellqualität anzupassen. Entscheidend ist dabei nicht die Qualität der Funktionsapproximation, sondern die Qualität des Vorselektionsprozesses auf Grundlage des Modells. Ein Modell ist dann gut, wenn dieselben Individuen auch bei einer Bewertung mit der tatsächlichen Fitnessfunktion vorselektiert würden.

Um die Qualität des Vorselektionsprozesses bewerten zu können, müssten eigentlich die tatsächlichen Fitnesswerte aller λ_p Nachkommen bekannt sein. Es wird daher stattdessen die Qualität des Selektionsprozesses von $\tilde{\mu}$ aus den λ Individuen bestimmt, für die sowohl die geschätzten Fitnesswerte als auch die tatsächlichen bekannt sind, und angenommen, dass dieses Qualitätsmaß mit dem der Vorselektion übereinstimmt. Damit auch bei einer μ aus $\lambda = \mu$ Selektion eine schlechte Modellqualität noch erkannt werden kann wird $\tilde{\mu}$ entsprechend Gleichung 11 gebildet. Damit stehen immer mindestens $\mu/2$ Stichproben zur Bestimmung der Modellqualität zur Verfügung.

$$\tilde{\mu} = \begin{cases} \mu, & \text{für } \lambda \geq 2\mu \\ \frac{\lambda}{2}, & \text{für } \lambda < 2\mu \end{cases} \quad (11)$$

Die Berechnung der Modellqualität nach [6] beruht auf der Anzahl der durch das Modell korrekt selektierten Individuen. Ein Individuum gilt dann als korrekt selektiert, wenn es ebenfalls durch einen Selektionsprozess basierend auf reellen Fitnessbewertungen selektiert worden wäre. Sortiert man alle λ Individuen aufsteigend nach ihrer tatsächlichen Fitness, lässt sich gemäß der Rangfolge dem i -ten Individuum eine Wertigkeit von $(\lambda - i)$ zuordnen. Die Qualität Q des Modells wird nun durch Summation der Wertigkeiten aller korrekt selektierten Individuen berechnet. Dieses Qualitätsmaß berücksichtigt nicht nur die Anzahl der korrekt selektierten Individuen, sondern gleichzeitig auch die Rangfolge der Individuen, die fälschlicherweise nicht selektiert worden sind.

Wird eine Zufallsauswahl von λ aus den λ_p Nachkommen getroffen, entspricht das Konvergenzverhalten der MAES dem einer Standard ES. Für die Zufallsauswahl wird der Erwartungswert der Modellgüte $\langle Q^{(rand)} \rangle$ [6] angegeben. Erreicht die Vorselektion eine bessere Qualität als eine Zufallsauswahl, werden also bevorzugt die gemäß ihrer tatsächlichen Fitness besten Individuen vorselektiert, wird das Konvergenzverhalten der MAES

gegenüber dem der Standard ES beschleunigt. Dagegen wird es verlangsamt, wenn die Vorselektion schlechter ist als eine Zufallsauswahl. Für die λ -CMAES wird daher das folgende Regelkriterium verwendet: Ist die aktuelle Güte der Vorselektion besser als eine Zufallsauswahl, dann wird λ entsprechend den Gleichungen 12 und 13 für die folgende Generation verringert. Umgekehrt wird λ mit erhöht, wenn die Vorselektion schlechter als eine Zufallsauswahl ist, um den Einfluss des Modells auf den Evolutionsprozess zu verringern.

$$\tilde{\lambda}^{(g+1)} = \lambda^{(g)} \cdot \left(\frac{1}{2}\right)^{\frac{Q^{(g)} - Q^{(rand)}}{Q^{(max)} - Q^{(rand)}} \cdot \delta_\lambda} \quad (12)$$

$$\lambda^{(g+1)} = \begin{cases} \lambda_p & \text{für } \tilde{\lambda}^{(g+1)} > \lambda_p \\ \mu & \text{für } \tilde{\lambda}^{(g+1)} < \mu \\ \tilde{\lambda}^{(g+1)} & \text{sonst} \end{cases} \quad (13)$$

Die λ -CMAES regelt eine Qualität der Vorselektion an, die der einer Zufallsauswahl entspricht. Damit müsste die Strategie über den Generationen aufgetragen nur ungefähr das Konvergenzverhalten einer Standard ES erzielen. Allerdings wird der gleiche Fortschritt der Optimierung nun mit deutlich weniger realen Funktionsauswertungen erreicht. Trägt man also die Fitness gegenüber den Funktionsauswertungen auf, ergibt sich für die λ -CMAES eine deutlich beschleunigte Konvergenz.

3 Modellgestützte multiobjektive evolutionäre Optimierung

3.1 Multiobjektive evolutionäre Optimierung

In der Praxis enthält die Mehrzahl von Optimierungsproblemen mehrere häufig zueinander in Konflikt stehende Kriterien f . In diesem Falle lässt sich ein einzelnes Gütekriterium nur auf Kosten der gleichzeitigen Verschlechterung anderer Kriterien verbessern. Ein Ansatz zur Lösung multiobjektiver Probleme besteht darin, alle Kriterien in einer skalaren Gütefunktion zusammenzufassen und mit einem herkömmlichen skalaren Optimierungsverfahren die entsprechend beste Lösung zu finden. Diese Vorgehensweise ist insofern problematisch, als dass es für den Anwender schwierig ist den Zielkonflikt oder die Gewichtung einzelner Kriterien zueinander a priori qualitativ zu beschreiben, insbesondere ohne Kenntnis der überhaupt realisierbaren Lösungen.

Daher zielt die multiobjektive evolutionäre Optimierung darauf ab, zunächst eine möglichst diverse Menge alternativer Lösungen zu generieren, aus denen dann im Nachhinein die beste Lösung ausgewählt wird. Die Mechanismen zur Mutation und Rekombination bei der multiobjektiven evolutionären Optimierung sind identisch zum skalaren Pendant. Bei einem multiobjektiven Problem lassen sich Lösungen nicht mehr anhand ihrer Güte in eindeutiger Reihenfolge anordnen, wodurch die klassische (μ, λ) -Selektion nicht weiter anwendbar ist. Stattdessen basiert die Selektion auf dem Konzept der Dominanz. Eine Lösung dominiert eine andere Lösung genau dann, wenn sie sich in mindestens einem Kriterium als besser und in keinem der übrigen Kriterien als schlechter erweist. Die Selektion

bevorzugt diejenigen Lösungen, welche von keinem anderen Individuum der Population dominiert werden. Der Rang eines Individuums ergibt sich zu

$$r_i = 1 + n_i \quad (14)$$

wobei n_i die Anzahl der übrigen Lösungen angibt, von denen das i -te Individuum dominiert wird. Alle nicht dominierten Lösungen erhalten den Rang eins und damit die höchste Fitness.

Darüberhinaus bestraft die Selektion zu ähnliche Lösungen mit dem Ziel, die Diversität der pareto optimalen Population aufrecht zu erhalten. Es werden zunächst alle Individuen mit optimalen Rang eins ausgewählt. Übersteigt deren Anzahl die der Eltern μ wird diese Menge sukzessive ausgedünnt, indem von dem jeweils nächstbenachbarten Paar eine Lösung entfernt wird, solange bis genau μ Eltern übrigbleiben.

Nicht dominierte Lösungen werden in einer fortlaufend aktualisierten Archivpopulation gespeichert, welche letztendlich die pareto optimale Lösungsmenge bildet. Eine Übersicht über die gängigen Methoden der multiobjektiven evolutionären Optimierung findet sich bei Coello [11] und Deb [10].

3.2 Fitnessmodellierung

Die Fitnessmodellierung bei multiobjektiven Problemen erfolgt analog zum skalaren Fall, wobei für jedes Kriterium ein separates Gütemodell erlernt wird. Als schwieriger erweist sich eine adäquate Definition der Modellgüte, da die Dominanzeigenschaft von allen Kriterien beeinflusst wird. Damit hängt die Fähigkeit der Fitnessmodellierung im Sinne der multiobjektiven Optimierung korrekte Selektionsentscheidungen zu treffen wesentlich von der Qualität des schlechtesten Einzelmodells ab. Im Folgenden schlagen wir drei alternative Ansätze zur Modellgüte im multiobjektiven Fall vor.

Das einfachste Kriterium berechnet die Modellgüte für einzelne Kriterien analog zum skalaren Fall in Gleichung 8. An die Stelle der Anzahl der Eltern μ wird hier die Summe jeweils bis zum halben Wert von λ gebildet. Die Güte des multikriteriellen Gesamtmodells ergibt sich dann aus der schlechtesten Einzelmodellgüte.

Die zweite Variante beruht auf der sogenannten S-Metrik [8, 9]. Die S-Metrik beschreibt das durch eine Menge nicht dominierter Lösungen dominierte Volumen des Lösungsraumes. Anschaulich wächst die S-Metrik mit der Annäherung der Population an die Paretofront an. Die auf der S-Metrik basierende Selektion bevorzugt diejenigen Lösungen welche den größten Zuwachs zur Verbesserung der S-Metrik beitragen. Wird die Fitness eines Individuums durch seinen Zugewinn an S-Metrik-Volumen definiert, dann lassen sich wiederum der anhand der Fitnessmodelle vorhergesagte Zugewinn mit dem tatsächlich realisierten Zugewinn vergleichen. Die nicht dominierten Lösungen lassen sich wie im skalaren Fall in eine eindeutige Rangfolge bringen. So lässt sich die Modellgüte durch Vergleich mit der geschätzten Rangfolge auf die ursprüngliche Definition in Gleichung 8 zurückführen.

Die dritte Definition der Modellgüte untersucht den Einfluss der Modellfehler auf die Dominanzrelation zwischen Paaren von Lösungen. Für zwei Lösungen A und B unterscheiden sich drei Fälle, A dominiert B , B dominiert A oder es liegt keine Dominanz

vor. Das Fitnessmodell ist umso besser, je häufiger es die tatsächliche Dominanzrelation korrekt vorherzusagen imstande ist. Die Modellgüte ergibt sich zu

$$Q_g = \sum_{i=1}^{\lambda-1} \sum_{j=i+1}^{\lambda} \delta(r(\mathbf{f}_i, \mathbf{f}_j), r(\hat{\mathbf{f}}_i, \hat{\mathbf{f}}_j)) \quad (15)$$

wobei $r(\mathbf{f}_i, \mathbf{f}_j)$ die tatsächliche Dominanzrelation und $r(\hat{\mathbf{f}}_i, \hat{\mathbf{f}}_j)$ die sich aus der geschätzten Fitness ergebende Dominanzrelation wiedergibt. Die δ -Funktion ist eins, wenn beide Relationen gleich sind.

Die λ -Regelung erfolgt in Analogie zur skalaren Fitnessmodellierung, in dem die Modellgüte mit derjenigen einer Zufallsauswahl verglichen wird, und bei guten Modellen sich der Wert von λ reduziert, wohingegen bei Modellen die lediglich rein zufällige Vorhersagen treffen, λ erhöht wird.

4 Ergebnisse

4.1 λ -geregelter Fitnessmodellierung für skalare Optimierungsprobleme

Die Analyse des Konvergenzverhaltens der λ -CMAES beruht auf dem in Abschnitt 2 vorgestellten Satz von Benchmarkoptimierungsproblemen. Die Ergebnisse basieren wiederum auf jeweils hundert gemittelten Läufen.

Die Graphen in Bild 3 vergleichen die Entwicklung der Fitness zwischen einer geregelten λ -CMAES, einer Standard ES und einer MAES. Unter der Annahme, dass der Aufwand einer Schätzung gegenüber einer realen Bewertung vernachlässigbar ist, und die Anzahl der Auswertungen pro Generation sich bei den drei Varianten unterscheidet, beruht ein fairer Vergleich auf der tatsächlichen Anzahl realer Fitnessbewertungen. Der Verlauf der Fitness in Bild 3 verdeutlicht die Überlegenheit der λ -CMAES, welche bei gleicher Anzahl von Auswertungen eine um Größenordnungen bessere Fitness erreicht. Umgekehrt betrachtet, erreicht die λ -CMAES den gleichen Gütewert wie eine herkömmliche Evolutionsstrategie mit etwa einem Drittel des Aufwands an Fitnessauswertungen. Für die MAES wird die Anzahl der zu schätzenden Nachkommen λ_p zuvor in unabhängigen Läufen problemspezifisch optimiert, es ergeben sich je nach Optimierungsproblem Werte $\lambda_p \in [300 \ 800]$. Die Anzahl der vorselektierten Individuen in der MAES bleibt konstant bei einem Wert $\lambda = 35$ für die beiden unimodalen und $\lambda = 50$ für die beiden multimodalen Probleme.

Weiterhin stellt sich heraus, dass die jeweils optimale Anzahl an Funktionsauswertungen λ sowohl problemspezifisch als auch abhängig von der Konvergenz der Population ist und sich im Laufe der Evolution ändert. Diese Tatsache verdeutlicht Bild 4 welche für die 10- und 30-dimensionale Griewank-Funktion die Entwicklung der Fitness über die Anzahl der Funktionsauswertungen zeigt. Die beiden unteren Graphen zeigen den Verlauf von λ bei der geregelten CMAES. Zum Vergleich dienen zwei MAES mit einem niedrigen Wert ($\lambda = \mu$, $\lambda = 2\mu$). Die Ergebnisse zeigen, dass ohne dazugehöriges Modellmanagement, bei konstant niedrigen Werten von λ die Optimierung Gefahr läuft vorzeitig zu konvergieren, wohingegen sich die geregelte λ -CMAES als robust erweist.

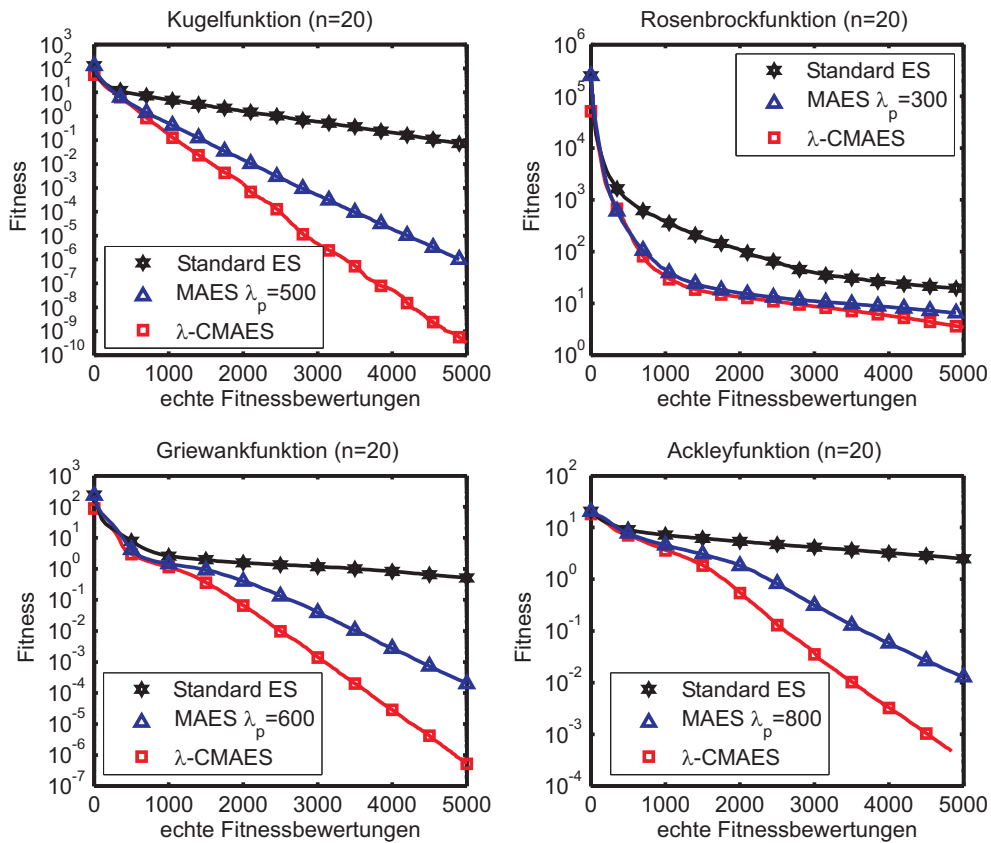


Bild 3: Verlauf der Güte gegenüber der Anzahl der tatsächlichen Fitnessbewertungen für eine Standard ES, eine MAES mit konstantem λ und geregelten λ -CMAES.

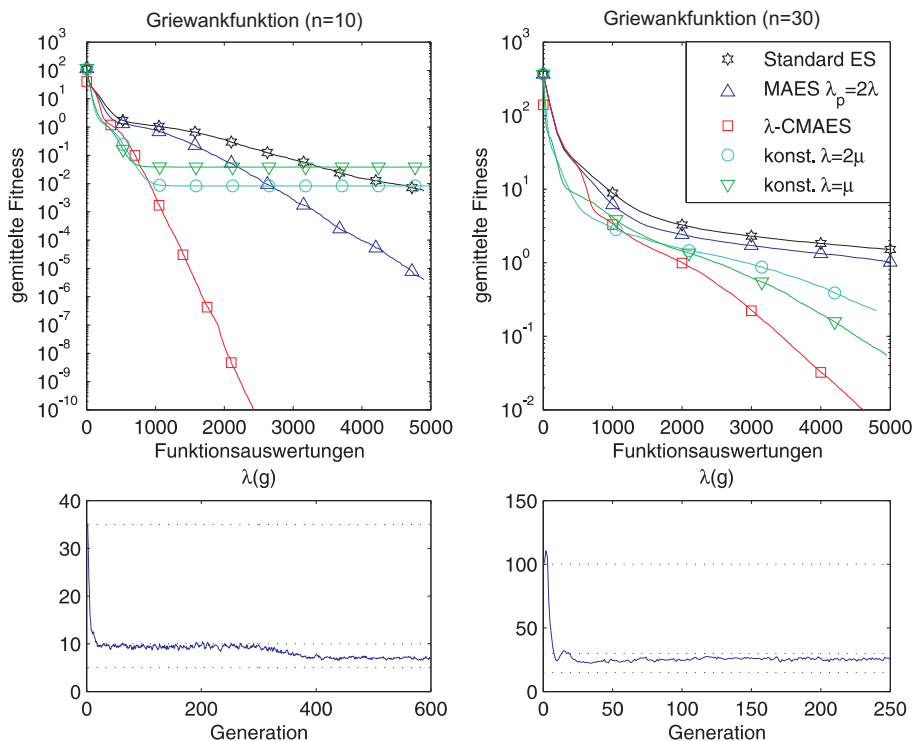


Bild 4: Verlauf der Güte gegenüber der Anzahl der tatsächlichen Fitnessbewertungen bei der Griewank-Funktion für eine Standard ES, drei MAES mit unterschiedlichem aber konstantem λ und einer geregelten λ -CMAES. Untere Graphen: Evolution von λ bei der geregelten λ -CMAES.

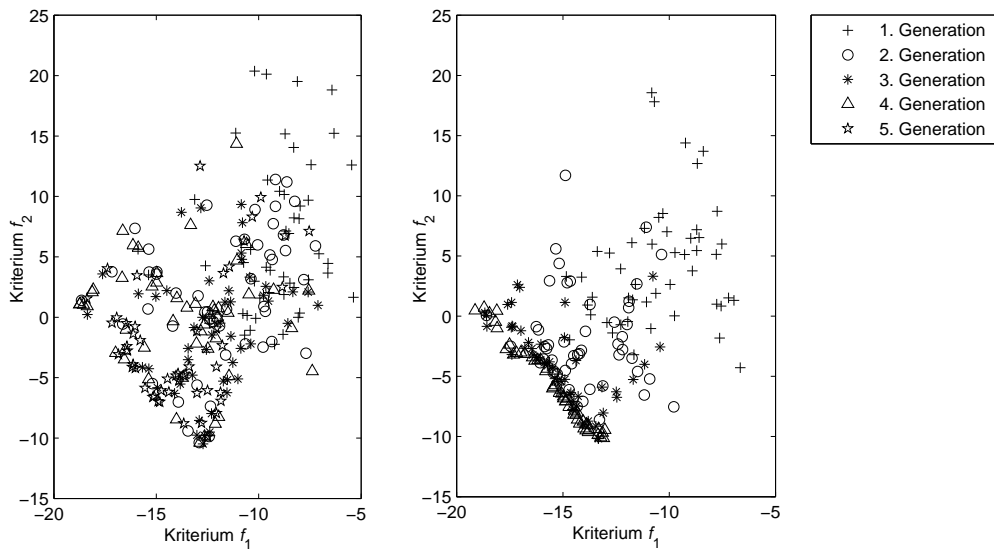


Bild 5: Entwicklung der Paretofront der standard (links) und einer modellgestützten (rechts) multiobjektiven Evolutionsstrategie über fünf Generationen

4.2 Fitnessmodellierung für multiobjektive Optimierungsprobleme

Als Benchmarkproblem für die multiobjektive Optimierung dient die Kursawefunktion

$$\begin{aligned}
 f_1(\mathbf{x}) &= \sum_{i=1}^{n-1} \left(-10 e^{(-0.2 \sqrt{x_i^2 + x_{i+1}^2})} \right) \\
 f_2(\mathbf{x}) &= \sum_{i=1}^n (|x_i|^{0.8} + 5 \sin(x_i)^3)
 \end{aligned} \tag{16}$$

mit den beiden in Konflikt stehenden Zielfunktionen f_1 und f_2 . Die durch die Kursawefunktion definierte Paretofront setzt sich aus wechselseitig konvexen und konkaven Segmenten zusammen. In den nachfolgend beschriebenen Optimierungsergebnissen beträgt die Dimension des Parameterraumes $n = 3$.

Bild 5 illustriert die Entwicklung der paretooptimalen Lösungen im Laufe von fünf Generationen für eine Standard (10+50)-ES (links) und eine (10+50)-MAES (rechts), bei welcher $\lambda_p = 200$ Nachkommen generiert und geschätzt werden, aus denen die $\lambda = 50$ besten Individuen der realen Bewertung unterzogen werden. Der Vergleich für einen prototypischen Lauf verdeutlicht, dass die modellgestützte Strategie wesentlich schneller gegen die Paretofront der Kursawefunktion konvergiert und diese nach fünf Generationen bereits dicht besiedelt.

Um die Robustheit und den Fortschritt der Optimierung qualitativ zu beurteilen, wird die S-Metrik herangezogen. Es wird die Differenz zwischen dem optimalen Wert der S-Metrik bei vollständiger Nachbildung der Paretofront und dem von der aktuellen Elitopopulation erzielten S-Metrik dargestellt. Die in Bild 6 dargestellte Entwicklung der residualen S-Metrik beruht auf der Mittelung von 20 Läufen über jeweils 10000 reale Fitnessbewertungen. Zum Vergleich stehen eine Standard (30+200)-ES, und zwei modellgestützte Strategien mit λ_p geschätzten Nachkommen, aus denen die jeweils $\lambda = 200$ bestgeschätzten real bewertet werden. Die modellgestützten Strategien unterscheiden sich im Fitnessmodell, linear gewichtete Regression gegenüber dem Nächster Nachbar Modell.

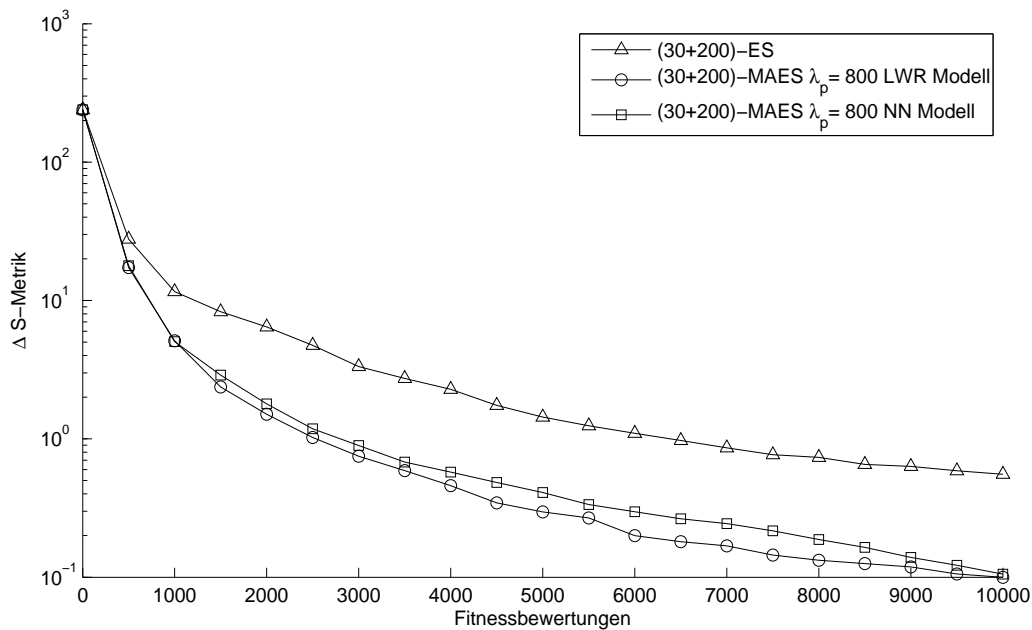


Bild 6: Entwicklung der residualen S-Metrik im Vergleich einer standard und zweier modellgestützten (LWR, NN) multiobjektiven Evolutionsstrategien über 10000 reale Fitnessbewertungen

Der Verlauf verdeutlicht die durch ein Fitnessmodell für die evolutionäre multiobjektive Optimierung erzielbare Einsparung an Auswertungen. Eine gleich gute Approximation der Paretofront erreicht die modellgestützte Variante mit etwa 4000 Auswertungen gegenüber 10000 Bewertungen einer Standard ES. Bei einer gleichen Anzahl von 10000 Auswertungen ist die verbleibende nicht durch die Elite abgedeckte Fläche um einen Faktor 10 kleiner.

Erste Experimente mit einer λ -geregelten modellgestützten Evolutionsstrategie brachten zunächst keine Verbesserung gegenüber der Fitnessmodellierung mit konstant gehaltenem λ . Dies beruht zum Teil darauf, dass bis zu 50 Prozent aller Individuen einer Population nicht-dominiert sind und daher einen Beitrag zur Reduktion der S-Metrik liefern. Im Gegensatz zur skalaren Optimierung, bei welcher nur die μ besten Lösungen zum Fortschritt der Evolution beitragen, sind bei der multiobjektiven Optimierung alle nicht dominierten Lösungen von zukünftigem Nutzen. Für eine effiziente Regelung von λ bedarf es daher eines anderen adäquaten Maßes zur Beurteilung der Modellgüte. Anstelle der rangbasierten Analyse der Modellgüte in Gleichung 8 beruht diese für die multiobjektive Optimierung auf der Fähigkeit die tatsächliche Dominanzrelation der Individuen der Population untereinander korrekt zu prognostizieren. Alternativ lässt sich die Modellgüte aus der Differenz einer geschätzten Reduktion der S-Metrik gegenüber der tatsächlich erzielten Verbesserung durch den Beitrag eines Individuums herleiten. Die Identifikation geeigneter Modellgüten und die λ -Regelung im Falle der multiobjektiven Optimierung sind Gegenstand aktueller Forschung.

5 Zusammenfassung

Eine modellgestützte Evolutionsstrategie eliminiert mit Hilfe eines erlernten Fitnessmodells im vorhinein wenig aussichtsreich erscheinende Individuen und unterzieht nur die potentiell besten Lösungen einer realen Bewertung. Die Approximation der Fitnessfunktion ermöglicht hinreichend zuverlässige Selektionsentscheidungen ohne den mit einer realen Fitnessbewertung verbundenen Aufwand. Zur Approximation der Fitnessfunktion und des zu erwartenden Schätzfehlers erweist sich die lokale lineare gewichtete Regression als instanzbasierte Methode aufgrund ihrer Online-Lernfähigkeit und der Fähigkeit zur Extrapolation als am Besten geeignet für die datenbasierte Fitnessmodellierung.

Aus Untersuchungen der Selektionsrate in Abhängigkeit der ausgeführten realen Fitnessauswertungen geht hervor, dass bei Optimierungsdurchläufen mit dem Fitnessmodell unnötig viele Funktionsauswertungen durchgeführt werden. Dieser Vergeudung von Auswertungen wird durch eine Strategie zur Regelung der Anzahl der realen Auswertungen (λ -CMAES) begegnet. Das Modellmanagement überwacht die Güte des Fitnessmodells hinsichtlich der Zuverlässigkeit der auf ihm beruhenden Selektionsentscheidung. Die Modellgüte bestimmt sich aus dem Vergleich der Selektionsentscheidung bei modellgestützter Bewertung mit der Entscheidung auf Basis einer optimalen Auswahl bei vollständiger Kenntnis der Fitnesswerte und einer reinen Zufallsauswahl. In der geregelten modellgestützten Evolutionsstrategie wird daher die Anzahl der tatsächlich bewerteten Individuen gerade soweit reduziert, bis die Qualität der Selektionsentscheidung gerade oberhalb der von einer rein zufälligen Auswahl zu erwartenden Qualität liegt. Mit anderen Worten, eine tatsächliche Fitnessbewertung erfolgt nur für die vermeintlich besten Kandidaten deren Fitness durch das datenbasierte Modell nicht unterscheidbar ist.

Beim Vergleich des Konvergenzverhalten der λ -CMAES Strategie an ausgewählten skalaren Benchmarkoptimierungsaufgaben erweist sich diese gegenüber sowohl einer Standard ES als auch einer MAES als überlegen hinsichtlich der Konvergenzrate und Robustheit der Optimierung.

Dieser Beitrag wendet erstmalig das Konzept der Fitnessmodellierung auf die evolutionäre Optimierung multiobjektiver Probleme an. Wiederum führt die auf dem Modell basierende Vorselektion zu einer deutlichen Reduktion benötigter Güteauswertungen beziehungsweise zu einer schnelleren Konvergenz der Lösung gegen die Paretomenge. Die im skalaren Fall zu beobachtenden Vorteile einer λ -geregelten gegenüber einer statischen MAES lassen sich bei dem untersuchten einfachen niedrigdimensionalen Benchmarkproblem nicht nachweisen. Eine genauere Analyse des unterschiedlichen Konvergenzverhaltens mit und ohne λ -Regelung im multiobjektiven bleibt zukünftigen Untersuchungen vorbehalten.

Literatur

- [1] Jin, Y.: *A Comprehensive Survey of Fitness Approximation in Evolutionary Computation*. In: *Soft Computing Journal*, 9(1) S. 3–12. 2005.
- [2] Jin, Y.; Olhofer M.; Sendhoff, B.: *A Framework for Evolutionary Optimization with Approximate Fitness Functions*. In: *IEEE Transactions on Evolutionary Computation*, 6(5) S. 481–494. 2002.
- [3] Gräning, L.; Jin, Y.; Sendhoff, B.: *Efficient Evolutionary Optimization Using Individual-based Evolution Control and Neural Networks: A Comparative Study*. In: *European Symposium on Artificial Neural Networks (ESANN'2005)*, S. 273–278. 2005.
- [4] Schwefel, H. P.: *Evolution and Optimum Seeking*. New York: Wiley Interscience. 1995.
- [5] Schöneburg, E.; Heinzmann, F.; Feddersen S.: *Genetische Algorithmen und Evolutionsstrategien. Eine Einführung in Theorie und Praxis der simulierten Evolution*. Bonn, Paris: Addison-Wesley. 1994.
- [6] Ulmer, H.; Streichert, F.; Zell, A.: *Evolution Strategies with Controlled Model Assistance*. In: *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, S. 1569–1576. 2004.
- [7] Atkeson, C.; Moore, A.; Schaal, S.: *Locally Weighted Learning*. In: *AI Review* S. 11-73. 1997.
- [8] Beume, N.; Naujoks, B.; Rudolph G.: *Mehrkriterielle Optimierung durch evolutionäre Algorithmen mit S-Metrik-Selektion*, In: *16th GMA Workshop on Computational Intelligence*, 2006.
- [9] Zitzler, E.; Thiele, L.: *Multiobjective Optimization Using Evolutionary Algorithm: A Comparative Case Study*. In: *Parallel Problem Solving from Nature (PPSN V)*, LNCS 1498, S. 292-301. Springer, Berlin 1998.
- [10] Deb, K.: *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, 2001.
- [11] Coello Coello, C.A. *Evolutionary Multi-Objective Optimization: A Historical View of the Field*, In: *IEEE Computational Intelligence Magazine*, S. 28-36. 2006.

Evolutionäre Optimierung des Betriebs von rohrlosen Chemieanlagen

Sabine Piana und Sebastian Engell

Lehrstuhl für Anlagensteuerungstechnik
Universität Dortmund, 44221 Dortmund

Tel.: (0231) 755 7378

Fax: (0231) 755 5129

E-Mail: Sabine.Piana@bci.uni-dortmund.de

Zusammenfassung

Rohrlose Chemieanlagen ermöglichen durch den Transport der Substanzen in mobilen Behältern eine sehr flexible Herstellung mehrerer Chargen auch unterschiedlicher Rezeptur. Durch die parallele Herstellung entsteht ein kombinatorisches Optimierungsproblem. Es müssen Entscheidungen zum zeitlichen Ablauf der Produktion, zur Auswahl der Apparate und zum Routing der Behälter getroffen werden. Ein Modellierungs- und Simulationsprogramm zur Unterstützung des Betriebs von rohrlosen Anlagen wurde um eine evolutionäre Optimierung ergänzt. Im Folgenden werden die Implementierung und erste Ergebnisse vorgestellt. Die Optimierung wird auf ein Beispiel aus der Literatur sowie eine industrielle Fallstudie angewendet.

1 Einleitung

Das Konzept der rohrlosen Anlage ist noch relativ jung. Im Unterschied zu traditionellen, rezepturgesteuerten Mehrproduktanlagen werden die Substanzen nicht durch feste Rohrleitungen zu den einzelnen Verarbeitungsstationen gebracht, sondern in mobilen Behältern mittels eines fahrerlosen Transportsystems (automated guided vehicles, AGVs) durch die Anlage befördert. Das Prinzip ist in Bild 1 dargestellt. Fest installierte Stationen können jeweils unterschiedliche verfahrenstechnische Operationen ausführen. Die Reihenfolge, in der diese Stationen angefahren werden, ist durch die Rezeptur der Charge vorgegeben. Die Behälter erfüllen dabei nicht nur den Zweck eines Transportmediums, sondern können auch als Reaktoren oder Zwischenlager dienen. Die Reinigung der Behälter wird durch separate Reinigungsstationen ausgeführt, während die Stationen durch den Anschluss von Spüleleitungen vor Ort gereinigt werden (cleaning in place, CIP).

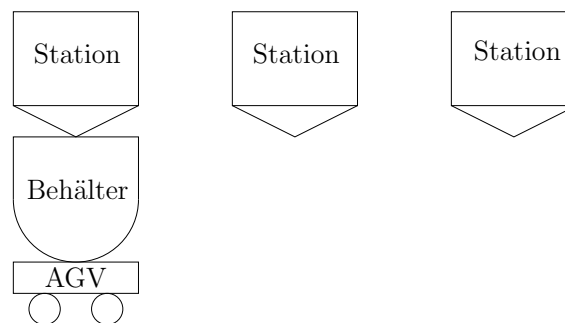


Bild 1: Das Prinzip einer rohrlosen Anlage

Eine rohrlose Anlage hat mehrere Vorteile. So können die Stationen mit den AGVs variabel angefahren werden, wodurch die Flexibilität im Bereich der Feinplanung erhöht wird. Des Weiteren können dringende Chargen vorgezogen oder eine defekte Station umfahren werden. Durch die Beförderung der Produkte in unterschiedlichen Behältern ist die Möglichkeit einer Querkontamination fast ausgeschlossen. Die Reinigungs- und Sterilisationszeit bei Produktwechseln verringert sich durch den geringen Anteil an fester Verrohrung. Im Gegensatz zu traditionellen Mehrproduktanlagen ist es nicht nötig, miteinander verschaltete Teile der Anlage während der Reinigung vollständig stillzulegen, da die Behälter separat gereinigt werden. Dadurch erreicht die rohrlose Anlage eine höhere Auslastung. Die erhöhte Flexibilität erfolgt jedoch auf Kosten einer erhöhten Zahl an Freiheitsgraden. Die kombinatorische Komplexität macht einen computergestützten Betrieb der rohrlosen Anlage unumgänglich.

Der Einsatz von rohrlosen Anlagen ist derzeit noch eingeschränkt. Alle Operationen einer Charge müssen in einem Behälter durchführbar sein. Die Apparate können bislang nur unter Umgebungsdruck betrieben werden und daher können nur Produkte mit geringer verfahrenstechnischer Komplexität hergestellt werden. Die möglichen Operationen beschränken sich auf Dosieren, Mischen, Heizen, Kühlen, Entleeren und Reinigen. Beispielsweise können Lacke, Klebstoffe und Life-Science Produkte in einer rohrlosen Anlage hergestellt werden. Ein umfassender Überblick über die technischen Anforderungen an eine rohrlose Anlage ist in [1] zu finden.

Rohrlose Anlagen wurden in der Literatur bislang relativ wenig betrachtet. Die erste umfassende Veröffentlichung ist [2], welche das Design, Layout und Scheduling rohrloser Anlagen simultan bearbeitet. Die Bestimmung des optimalen Anlagenlayouts wurde häufiger betrachtet [3, 4]. Zur Auswahl stehen das sogenannte Herringbone, kreisförmige oder lineare Layout. Zur Optimierung des Betriebs von rohrlosen Anlagen wurden hauptsächlich gemischt-ganzzahlige lineare oder nichtlineare Programme verwendet. Diese wurden mit Heuristiken [5] oder einem Queueing Ansatz [6] kombiniert. Auch ein Ansatz mit Hilfe von Constraint Satisfaction Methoden existiert [3]. Die Besonderheit der hier vorgestellten und weiterentwickelten Software PPSiM [7] ist, dass alle Aspekte der rohrlosen Anlage, welche im folgenden Abschnitt näher erläutert werden, betrachtet werden.

2 Problemstellung

Es wird angenommen, dass die Produktionsnachfrage, die Anzahl und Größe der Chargen, alle Rezepturen und das Anlagenlayout bekannt sind. Es soll ein zulässiger Produktionsplan mit minimaler Bearbeitungszeit (Makespan) gefunden werden. Die Herausforderung liegt dabei darin, dass die Berechnung des Makespan zeitaufwendig ist, da die genaue Bearbeitungszeit an den Stationen durch Simulation errechnet werden muss. Des Weiteren müssen drei sich gegenseitig beeinflussende Probleme während des Betriebs einer rohrlosen Anlage gelöst werden:

1. Die zeitliche Planung der Operationen der Produktionsaufträge (Scheduling)

Mehrere Chargen unterschiedlicher Produkte werden parallel hergestellt. Die Reihenfolge der Operationen muss bestimmt werden und sie muss bestimmten Bedingungen, welche durch die Rezepturen definiert sind, gerecht werden.

2. Die Zuteilung von Apparaten (Assignment)

Jeder Operation wird ein Behälter, eine Stationen und ein AGV zugeteilt. Die gewählten Apparate müssen über die technischen Funktionen verfügen, die zur Ausführung der Operation nötig sind.

3. Die Wegeberechnung der AGVs (Routing)

Nach der Apparatezuordnung muss das AGV den Behälter aufnehmen und zur Station transportieren. Dabei darf es nicht mit anderen AGVs kollidieren.

Aufgrund dieser unterlagerten Probleme ist es ohne große Vereinfachungen nicht möglich, eine geschlossene mathematische Formulierung für die Optimierung einer rohrlosen Anlage zu finden.

3 Simulation von rohrlosen Anlagen

Am Lehrstuhl für Anlagensteuerungstechnik wurde die Software PPSiM zur Modellierung, Simulation und Produktionsplanung von rohrlosen Anlagen entwickelt. In einer industriellen Fallstudie zeigte sich, dass eine rohrlose Anlage gegenüber einer traditionellen Mehrproduktanlage eine signifikant geringere Produktionsdauer aufweist [7].

Über einen Anlageneditor kann der Benutzer das Anlagenlayout, d.h. die Anzahl der AGVs, Behälter und Stationen sowie deren verfahrenstechnische Funktionen, Eigenschaften und Position in der Anlage, spezifizieren. Die Rezepturen der geforderten Produkte werden in einem Rezepteditor beschrieben und es gibt einen Produktionsplanungeditor, in welchem die gewünschte Anzahl der Chargen und deren Produkte eingegeben werden. Diese Editoren stellen die Modellierungsebene von PPSiM dar. Das Hauptprogramm besteht aus einem Modul zum Scheduling der Operationen, welches eine sogenannte as-soon-as-possible (ASAP) Heuristik verwendet. Die Operationen werden dabei chronologisch nach ihrer frühestmöglichen Startzeit eingeplant. Dazu wird der Operation ein Behälter, ein AGV und eine Station zugewiesen. Diese Auswahl erfolgt ebenfalls nach einer Heuristik, die den Zeitpunkt, zu dem der Apparat verfügbar ist, nutzt. Alternativ kann der Benutzer die Apparatezuweisung manuell vorgeben. Zur Einplanung der Operation ruft das Scheduling Modul einen unterlagerten diskreten Simulator für die Berechnung der Dauern der verfahrenstechnischen Operationen und ein Modul zur Berechnung von konfliktfreien kürzesten AGV Wegen auf. Das Routing wird von einem A* Algorithmus berechnet und der dadurch gefundene Weg im Falle einer Kollision durch Warten oder Ausweichmanöver zulässig gemacht. Des Weiteren besitzt PPSiM ein Visualisierungstool zur Anzeige von Gantt Charts der Belegungen oder Zeitverläufen beispielsweise der Temperaturen des Behälterinhaltes.

PPSiM bietet zu den in Abschnitt 2 beschriebenen Problemen Entscheidungshilfen. Neben der Möglichkeit die rohrlose Anlage detailliert zu modellieren, können für unterschiedliche Entscheidungen Simulationsstudien durchgeführt werden. Eine Optimierung der Entscheidungsvariablen führt das Programm jedoch nicht durch.

4 Einsatz eines evolutionären Algorithmus

Die in PPSiM verwendeten einfachen Heuristiken zur Einplanung der Operationen wurden um eine Optimierung mit Hilfe eines evolutionären Algorithmus (EA) erweitert. Der EA ersetzt die ASAP Heuristik zum Scheduling der Operationen. Statt der chronologischen Reihenfolge anhand der frühesten Startzeit werden viele mögliche Reihenfolgen der Operationen ausgewertet. Die Hauptschwierigkeit besteht darin, zulässige Individuen zu generieren, da die Auswertung einer Lösung komplex und zeitintensiv ist. PPSiM evaluiert die Produktionsdauer der Lösungen, indem es die Startzeiten und die Dauern der Operationen, die Apparatebelegungen und das Routing der Behälter simuliert. Diese Auswertung erfolgt mit Hilfe der heuristischen Apparatzuweisung, der Simulation der Prozessdauern und der Wegeberechnung mit dem A* Algorithmus. Es wird ein $(\mu + \lambda)$ Selektionsschema verwendet, in dem jeweils λ Kinder zur Population hinzugefügt werden und dann unter allen $\mu + \lambda$ Individuen die λ mit der schlechtesten Fitness entfernt werden.

Die Anfangspopulation des EA beinhaltet neben der heuristischen ASAP Lösung weitere zufällig generierte Lösungen. Darunter sind auch sehr einfache Lösungen, in denen die Chargen nacheinander abgearbeitet werden, so dass im Prinzip nur ein AGV und ein Behälter erforderlich sind. Jede Lösung wird durch eine Reihenfolge der Operationen aller Chargen dargestellt. Das Chromosom ist eine Sequenz von Operationen und stellt eine Prioritätenliste dar. Die Operationen werden nicht anhand ihrer eigenen ID, sondern anhand der Chargennummer identifiziert. Dies stellt sicher, dass Reihenfolgebeziehungen innerhalb der Rezepte nicht durch Rekombinations- und Mutationsoperatoren verletzt werden. Die Idee ist in Bild 2 anhand eines Beispiels dargestellt. Es gibt zwei Chargen (A und B), welche jeweils drei Operationen haben. Ein Chromosom auf Basis der Operationsbezeichnungen (1-6) enthält Reihenfolgebeziehungen, welche im Bild als Pfeile dargestellt sind. Eine solche Codierung würde die Variationsoperatoren stark einschränken. Die Darstellung des Chromosoms anhand der Chargenbezeichnung vereinfacht das Reihenfolgeproblem. Werden in diesem Chromosom Einträge beispielsweise vertauscht, bleibt die Reihenfolge innerhalb eines Rezeptes, also einer Charge, unberührt. Dies ist der Fall, da ein Eintrag des Chromosoms so interpretiert wird, dass die erste noch nicht eingeplante Operation der Charge eingeplant werden soll.

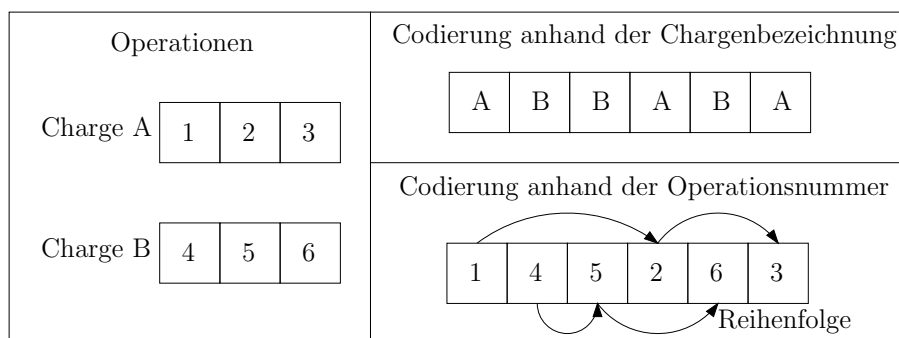


Bild 2: Codierung eines Chromosoms

Da jede Chargenbezeichnung so oft im Chromosom vorkommen muss, wie sie Operationen beinhaltet, wurde ein Rekombinationsoperator, welcher Permutationen beibehält,

implementiert. Dies ist der Order Crossover Operator. Die Eltern werden durch eine Tournament Selektion ausgewählt. Dann werden zwei Crossover Punkte bestimmt und das Segment dazwischen vom ersten Elternteil ins Chromosom des Kindes kopiert. Die restlichen Einträge werden, angefangen am zweiten Crossover Punkt, vom zweiten Elternteil aufgefüllt, wobei doppelte Einträge übersprungen werden. Durch Austausch der Eltern kann ein zweites Kind generiert werden. Das Prinzip ist in Bild 3 dargestellt. In unserem Fall werden Einträge nicht übersprungen wenn sie doppelt vorkommen, sondern wenn die Chargenbezeichnung schon genügend oft (Anzahl der Operationen) im Chromosom vorkommt. Die Mutation erfolgt durch einen einfachen Swap.

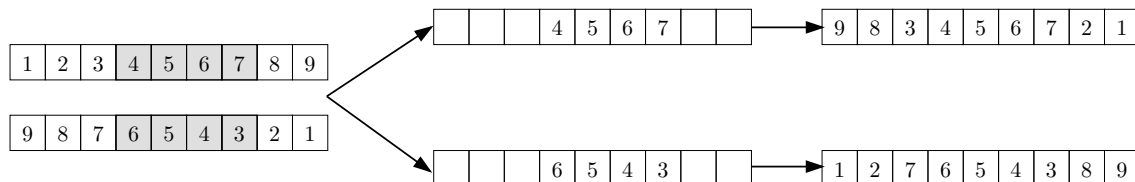


Bild 3: Order Crossover

Für die Evaluierung der Lösungen wird eine Verbesserung der Einplanung gegenüber der Originalversion von PPSiM realisiert. Die Zeitpunkte, an denen der diskrete Simulator versucht, Operationen einzuplanen, werden nicht nur nach Beendigung einer Operation sondern auch nachdem ein AGV einen Weg beendet und nach dem CIP einer Station gesetzt. Dadurch können Lücken im Belegungsplan, welche durch unnötiges Warten entstehen, geschlossen werden. Die Auswertung eines Individuums für eine vorgegebene Reihenfolge liefert dadurch einen etwas geringeren Makespan. Die Auswertung der Individuen in einer Generation des EA ist schneller als die Durchführung einer Planung mit PPSiM, da die Reihenfolge der Operationen bereits vorgegeben ist und nicht bestimmt werden muss.

5 Probleme mit der Apparateauswahl

Durch die Änderung der Einplanungsreihenfolge ergibt sich das Problem, dass nicht mehr alle Apparate, die eigentlich zum Zeitpunkt der Einplanung einer Operation verfügbar sind, ausgewählt werden können. Im Gegensatz zur ASAP Heuristik, in der chronologisch geplant wurde, springt der Algorithmus in der Zeit. Die Operationen werden jeweils so früh wie möglich eingeplant. Das kann auch vor bereits eingeplanten Operationen stattfinden, sofern diese dadurch nicht verzögert werden. Genau dort liegt jedoch das Problem. Durch die Einplanung auf einem Apparat, der zu einem späteren Zeitpunkt bereits verplant ist, ist nicht mehr sichergestellt, dass die späteren Operationen wie geplant ausgeführt werden können. Bild 4 zeigt das Problem an einem Beispiel. Zu sehen ist die Belegung von zwei AGVs. Angenommen, die ersten beiden Operationen des Chromosoms sind bereits zu unterschiedlichen Zeitpunkten eingeplant worden. Die dritte Operation sollte nun so früh wie möglich eingeplant werden. Auf AGV 2 ist dies jedoch nicht möglich. Zwar würde die Operation zeitlich in die Lücke passen, das AGV befände sich aber nach Ausführung der Operation für die kommende Operation (hier Operation 1) an einer anderen Position in der Anlage. Das bedeutet, dass das Routing für Operation 1 neu berechnet werden müsste. Die Transferzeiten sind also reihenfolgeabhängig. Eine komplette Neuberechnung des Routings erscheint zu zeitaufwendig. Alternativ könnte eine Fahrt des

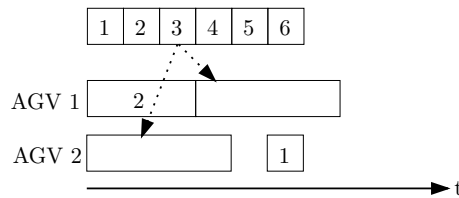


Bild 4: Das Problem des Lückenfüllens

AGVs zur korrekten Position eingefügt werden. Dies würde die Makespan allerdings etwas verlängern. Daher wird das Problem dadurch umgangen, dass nur Apparate, welche zu einem späteren Zeitpunkt nicht mehr genutzt werden, ausgewählt werden dürfen.

6 Vermeidung unzulässiger Individuen

Unzulässigkeiten, die in den zufällig generierten Anfangslösungen und durch Rekombination und Mutation auftreten können, werden durch einfache Algorithmen repariert. Sie verwenden problemspezifisches Wissen, welches aus der ASAP Lösung gewonnen wird. Das Nutzen von problemspezifischem Wissen im EA ist unumgänglich, da sonst der Großteil der generierten Lösungen unzulässig wäre. Die folgenden Daten werden verwendet:

1. Die Chargenrezepte, d.h. die Operationen und ihre Reihenfolge,
2. die Anzahl der Behälter in der Anlage und
3. die Anzahl der verfügbaren Stationen für jede Operation.

Die unterschiedlichen Fälle, in denen ein Individuum unzulässig ist, können in drei Gruppen eingeteilt werden und sind in Tabelle 1 zusammengefasst. Der einfachste Grund für eine Unzulässigkeit, nämlich die Verletzung der Reihenfolge innerhalb einer Rezeptur, wurde bereits durch die in Abschnitt 4 beschriebene Codierung im Chromosom vermieden. Unzulässigkeiten der Gruppe II können vor der Evaluierung erkannt und durch Reparaturalgorithmen behoben werden. Die zur letzten Gruppe gehörenden Unzulässigkeiten werden erst während der Evaluierung von PPSiM erkannt und können dann lediglich noch durch Strafterme vermieden werden. Beispielsweise kann der EA vor der Auswertung nicht erkennen, ob Fertigstellungstermine für die Chargen eingehalten werden. Es kann auch passieren, dass ein benötigtes AGV oder ein Behälter von anderen Apparaten

	Grund der Unzulässigkeit	Behebung der Unzulässigkeit
Gruppe I	Unzulässige Operationsreihenfolge	Codierung im Chromosom
Gruppe II	Nicht verfügbare Station Nicht ausreichend viele Behälter	Reparatur
Gruppe III	Verletzung zeitlicher Bedingungen Nicht vorhandener AGV Weg	PPSiM Strafterm

Tabelle 1: Behebung der unterschiedlichen Unzulässigkeiten

blockiert sind und deshalb kein Weg für das AGV gefunden werden kann. Der Strafterm ist umso höher, je mehr Operationen nicht mehr eingeplant werden können. Dabei wird angenommen, dass ein Individuum, welches bis auf einige wenige Operationen zulässig ist, besser ist als ein Individuum, in dem die Unzulässigkeit bereits relativ weit am Anfang auftritt.

Im Folgenden soll auf die Reparaturalgorithmen zur Behebung der Unzulässigkeiten der Gruppe II näher eingegangen werden. Ein Reparaturalgorithmus ist erforderlich, damit möglichst nur zulässige Individuen an PPSiM zur Auswertung gegeben werden. Es wurden zwei Reparaturalgorithmen implementiert zur Vermeidung von belegten Stationen und Behältern. Beide Reparaturen halten sich an die Interpretation des Chromosoms, dass dieses eine Prioritätenliste darstellt. Später im Chromosom stehende Operationen dürfen also frühere Operationen nicht verzögern. Wenn allerdings eine Operation an ihrer Position nicht einplanbar ist, darf diese Einplanung durch Vorziehen einer späteren Operation ermöglicht werden.

6.1 Freigabe von belegten Stationen

Eine häufig auftretende Unzulässigkeit tritt auf, wenn eine Operation an einer Station ausgeführt werden soll, die jedoch zu diesem Zeitpunkt noch von einem anderen Behälter belegt ist, der auf seinen Transport zur nächsten Station wartet. Der Reparaturalgorithmus überprüft für jeden Eintrag im Chromosom, ob eine Station für die benötigte verfahrenstechnische Aufgabe verfügbar ist. Sollten alle Stationen, die die Operation ausführen können, belegt sein, wird die nächste Operation im Chromosom, welche zu einer Charge gehört, die aktuell die erforderliche Station blockiert, vor der unzulässigen Operation eingefügt. Damit wird die benötigte Station freigegeben, da die Charge nun zur Station für die nächste Operation fährt. Die eingefügte Operation muss jedoch ebenfalls auf Zulässigkeit überprüft werden. Die Idee ist in Bild 5 mit Hilfe eines Beispielchromosoms verdeutlicht. Angenommen, es befinden sich zwei Dosierstationen in der Anlage. Nach der vierten Operation sind diese beiden Stationen belegt, nämlich durch Charge 2 und Charge 3. Daher ist der folgende Eintrag im Chromosom, die Dosierung von Charge 4, unzulässig aufgrund der Belegung aller infrage kommenden Stationen. Eine der beiden Dosierstationen muss freigegeben werden. Dies kann durch Vorziehen der siebten Operation, Aufheizen von Charge 3, erreicht werden. Durch diesen Schritt wird eine Dosierstation verfügbar, da Charge 3 von der Dosierstation zur Heizstation fährt. Jedoch muss nun überprüft werden, ob eine Heizstation für diese Operation zur Verfügung steht. Gegebenenfalls ist eine weitere Reparatur notwendig.

2 verfügbare Dosierstationen

$n :=$ Anzahl der belegten Dosierstationen

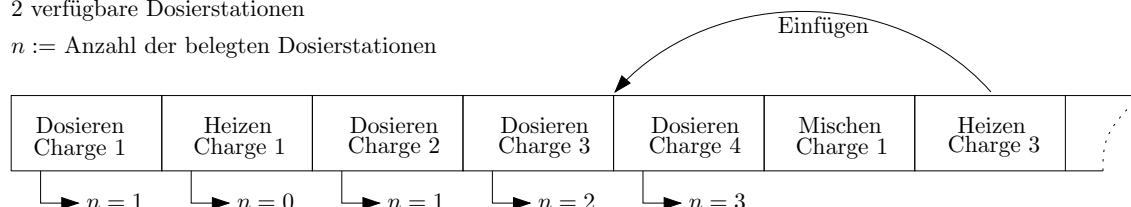


Bild 5: Reparatur einer belegten Station

Ein Spezialfall liegt vor, wenn zwei Behälter auf eine Station warten, die der jeweils andere Behälter gerade belegt. In diesem Fall ist es nicht möglich, die Unzulässigkeit durch

Einfügen einer Operation zu beheben. Daher werden in diesem Fall alle Einträge einer der beiden wartenden Chargen ans Ende des Chromosoms geschoben. Alternativ könnten Parkplätze in der Anlage eingerichtet werden, zu denen die Behälter nach Abschluss der Operation fahren. Diese Option wurde nicht implementiert, da dadurch unnötige Transporte stattfinden, die sich negativ auf den Makespan auswirken.

6.2 Vermeidung eines Mangels an Behältern

Falls weniger Behälter in der Anlage verfügbar sind als Chargen produziert werden sollen, kann es passieren, dass für die erste Operation einer Charge kein freier Behälter mehr zur Verfügung steht. Das liegt daran, dass innerhalb einer Charge kein Behälterwechsel stattfindet. Der Reparaturalgorithmus prüft für jeden Eintrag im Chromosom, wie viele Chargen aktuell produziert werden. Sollte diese Zahl größer sein als die Anzahl der Behälter, werden die nicht einplanbaren Operationen der Charge an der Position im Chromosom eingefügt, an der eine andere Charge beendet ist und somit einen Behälter freigibt. Dabei werden nur die Operationen eingefügt, die im Chromosom vor dieser Position stehen, d. h. Operationen werden nicht vorgezogen.

Dieser Reparaturalgorithmus wird solange abwechselnd mit dem in Abschnitt 6.1 beschriebenen Algorithmus ausgeführt, bis keine Unzulässigkeit mehr vorliegt. Es kann jedoch passieren, dass die Algorithmen gegensätzliche Änderungen am Chromosom vornehmen, so dass die Zulässigkeit niemals erreicht wird. Sollte nach einer bestimmten Anzahl an Iterationen keine Zulässigkeit vorliegen, so wird der Algorithmus zur Reparatur von fehlenden Behältern wie folgt abgeändert: Die nicht einplanbaren Chargen werden statt hinter eine fertig gestellte Charge komplett ans Ende des Chromosoms geschoben. Dadurch wird niemals eine neue Unzulässigkeit aufgrund einer belegten Station produziert und die Zulässigkeit somit erreicht.

7 Problembeispiele

Die Kombination aus EA zum Scheduling der Operationen und Auswertung mit PPSiM wurde auf drei Beispiele angewendet. Das erste ist ein kleines Beispiel, von dem zwei Varianten mit unterschiedlicher AGV Geschwindigkeit betrachtet werden. In Beispiel 1a können die AGVs mit einer Geschwindigkeit von 0.025 m/s fahren; in Beispiel 1b lediglich mit 0.01 m/s. Beispiel 2 stammt aus einer industriellen Fallstudie [7]. Des Weiteren wurde die Implementierung auf ein Beispiel (3) aus der Literatur [3] angewendet, das im folgenden Abschnitt näher erläutert wird. Die Größe der Beispiele ist in Tabelle 2 angegeben.

	# Operationen	# Behälter	# Stationen	# AGVs	# Chargen
Beispiel 1	18	3	6	2	4
Beispiel 2	84	10	7	2	7
Beispiel 3	35	3	8	3	5

Tabelle 2: Größe der Probleminstanzen

7.1 Benchmark Problem aus der Literatur

Für das Beispiel von Huang & Chung [3] werden insgesamt drei Anlagenlayouts beschrieben. In dieser Arbeit werden jedoch nur das Herringbone (Beispiel 3a) und das lineare Layout (Beispiel 3b) betrachtet, da sich das kreisförmige Layout noch nicht in PPSiM umsetzen lässt.

Huang & Chung treffen in den folgenden Punkten Vereinfachungen gegenüber unserer Modellierung:

1. Es werden feste Bearbeitungszeiten an den Stationen angenommen. Dadurch ist die Simulation, welche in PPSiM die Dauer der verfahrenstechnischen Operationen bestimmt, überflüssig. Im Rezepteditor von PPSiM werden daher feste Bearbeitungszeiten angegeben.
2. Es wird angenommen, dass die AGVs nur auf festgelegten Wegen fahren. Dies kann in PPSiM durch die Markierung von nicht befahrbaren Feldern in der Anlage sichergestellt werden. Für das Herringbone-Layout ist dies in Bild 6 dargestellt. Das Routing muss dann nicht mehr mit dem A* Algorithmus berechnet werden.



Bild 6: Das Herringbone-Layout

3. Es werden feste Transferzeiten für die AGVs angenommen. Um dies umzusetzen wird in PPSiM die Geschwindigkeit der AGVs entsprechend angepasst, so dass sie in Kombination mit der Tatsache, dass es nur einen möglichen Weg gibt, genau die vorgegebenen Transferzeiten einhalten. Diese Annahme ist wenig realistisch.
4. Es werden keine Reinigungszeiten (CIP) der Stationen vorgesehen. Diese werden in PPSiM auf 0 gesetzt.
5. Die Behälter sind selbst mit einem Transportsystem ausgestattet. In PPSiM werden sie jedoch von AGVs aufgenommen und befördert. Das Problem wird gelöst, indem

genauso viele AGVs wie Behälter in die Anlage eingefügt werden. Durch die stark beschränkte Struktur der Anlage bleibt jedes AGV während der ganzen Produktion bei einem Behälter. Es wird kein Behälterwechsel ausgeführt. Die Zuweisung eines AGVs zu einer Operation ist dadurch überflüssig.

In den folgenden Aspekten ist eine Umsetzung des Beispiels in PPSiM nicht möglich. Dadurch löst PPSiM ein schwereres Problem.

1. Die Stationen verfügen über Puffer, so dass ein AGV vor der Station und eins nach der Station warten kann. Diese Puffer können aber nicht als Fahrwege benutzt werden. Die Puffer könnten in PPSiM leicht berücksichtigt werden, würden dann jedoch auch als Fahrwege benutzt. Daher wird dies in PPSiM zunächst nicht berücksichtigt.
2. Die AGVs können sich an bestimmten Wegkreuzungen überholen. Auch dies ist in PPSiM nicht implementiert und wird ignoriert.

8 Ergebnisse

Die Parameter des EA wurden auf $\mu = 10$ und $\lambda = 2$ gesetzt. Crossover wird mit einer Wahrscheinlichkeit von 100 % ausgeführt und die Mutationswahrscheinlichkeit ist 80 %. Es wurde ein Zeitlimit von 60 Sekunden gesetzt. Tabelle 3 zeigt zunächst den Vergleich zwischen der heuristischen ASAP Lösung, die vom ursprünglichen Planungsalgorithmus gefunden wurde, und der Lösung der geänderten Version zur Evaluierung der vorgegebenen ASAP Reihenfolge. Die Daten sind in Sekunden angegeben und die Spalten mit den Prozentwerten zeigen die Verbesserung der geänderten Version gegenüber der Originalversion an. Dabei ist zum einen zu sehen, dass der Makespan durch die Anpassung der Zeitpunkte des diskreten Simulators gesunken ist, und zum anderen, dass die Evaluierung mit vorgegebener Reihenfolge wesentlich schneller ist als die Durchführung einer Planung mit der ASAP Heuristik.

Die Ergebnisse von fünf Läufen des EA sind in Tabelle 4 zu sehen. Die prozentuale Verbesserung ist auf die verbesserte ASAP Lösung bezogen, stellt also die tatsächlich vom EA erzielte Verbesserung dar. Der EA konnte für Beispiel 1 eine wesentlich größere Verbesserung erzielen als für das industrielle Beispiel 2. Die besten gefundenen Lösungen für Beispiel 1 stellen die Chargen nicht parallel sondern eher sequentiell her. Es scheint wichtig zu sein, dass nicht zu viele Chargen auf einmal gestartet werden. Die ASAP Lösung

	Original PPSiM		PPSiM mit vorgegebener Reihenfolge			
	Rechenzeit	Makespan	Rechenzeit		Makespan	
Beispiel 1a	0.906	5551	0.313	-65.5 %	5316	-4.2 %
Beispiel 1b	0.563	8353	0.234	-58.4 %	7453	-10.8 %
Beispiel 2	6.828	89776	1.422	-79.2 %	84929	-5.4 %
Beispiel 3a	1.227	54324	0.740	-38.3 %	52080	-4.1 %
Beispiel 3b	1.406	78561	0.906	-35.6 %	65961	-16.0 %

Tabelle 3: Ergebnisse der ASAP Heuristik

	Best		Worst		Median	
Beispiel 1a	4652	-12.5 %	4700	-11.6 %	4683	-11.9 %
Beispiel 1b	6235	-16.3 %	6303	-15.4 %	6254	-16.1 %
Beispiel 2	82914	-2.4 %	84929	-0.0 %	84053	-1.0 %
Beispiel 3a	49844	-4.3 %	50244	-3.5 %	50003	-4.0 %
Beispiel 3b	58120	-11.9 %	62680	-5.0 %	61281	-7.1 %

Tabelle 4: Ergebnisse des EA (Zeitlimit 60 Sekunden)

forciert jedoch genau das, und daher blockieren sich Behälter gegenseitig und die Stationen werden erst später belegt. Für das industrielle Beispiel 2 scheint die ASAP Lösung hingegen schon sehr gut zu sein.

Unser Ergebnis für Beispiel 3 lässt sich nicht direkt mit den Ergebnissen von Huang & Chung vergleichen. PPSiM bietet Funktionalitäten an, wie z.B. die Simulation und das Routing, die in diesem Beispiel nicht benötigt werden. Da jedoch trotzdem gerechnet wird, braucht unsere Lösung mehr Rechenzeit. Auf der anderen Seite löst PPSiM durch die in Abschnitt 7.1 beschriebenen Annahmen ein schwereres Problem. Da es in PPSiM nicht vorgesehen ist, dass Behälter an Stationen warten können, ist es auch nicht möglich, die leeren Behälter direkt vor der ersten Station zur Wiederverwendung zu platzieren. Stattdessen wurden in PPSiM am Rand der Anlage entsprechende Parkplätze angelegt. Das hat zur Folge, dass bei jeder Charge ein Weg vom Parkplatz zur ersten Station und nach der Abfüllung wieder zurück zum Parkplatz hinzugefügt wird. Das sind z. B. im Herringbone Layout 18 Minuten reine Fahrzeit pro Charge. Es muss bedacht werden, dass diese zusätzliche Fahrzeit aufgrund von Warten wegen Belegung der Strecke größer sein kann. Da in unserem Fall keine Überholmöglichkeit besteht, verschlechtert sich die Lösung ebenfalls.

Die Ergebnisse werden in Tabelle 5 verglichen. Gezeigt sind jeweils die erste gefundene Lösung und die beste gefundene Lösung. Die letzte Spalte zeigt die Optimalitätslücke unserer Ergebnisse an, wobei dies aus den oben genannten Gründen eine obere Schranke ist. Zieht man im Herringbone Layout die angesprochenen 18 Minuten zusätzliche Fahrzeit ab, beträgt die Optimalitätslücke nur noch 0.4 %. Die Rechenzeiten (in Sekunden) können nicht direkt miteinander verglichen werden, da die Rechnerleistungen unterschiedlich sind. Wir verwenden einen Intel Pentium 4 mit 3 GHz und 512 MB RAM, der dem P3500 MHz PC von Huang & Chung leistungsmäßig überlegen ist. Die Ergebnisse zeigen, dass PPSiM mit der ASAP Heuristik eine sehr gute erste Lösung findet und dass diese durch den EA innerhalb nur einer Minute verbessert werden kann. Die Rechenzeit un-

	PPSiM + EA		Huang & Chung		
	Rechenzeit	Makespan	Rechenzeit	Makespan	
Herringbone (Beispiel 3a)					
Erste Lösung	1.97	52080	0.1	63720	
Beste Lösung	25.5	49844	722.599	44280	11.2 %
Linear (Beispiel 3b)					
Erste Lösung	2.31	65961	0.12	70200	
Beste Lösung	31.5	58120	876.44	48600	16.4 %

Tabelle 5: Vergleich mit dem Benchmark

rer Anfangslösung ergibt sich aus der benötigten Zeit für die ursprüngliche ASAP Lösung und der erneuten, angepassten Evaluierung. Die Rechenzeit für die beste Lösung ist der Zeitpunkt, an dem der EA die beste Lösung gefunden hat. Das geschieht in beiden Beispielen vor Ablauf des Zeitlimits.

Bild 7 zeigt die prozentuale Verbesserung des besten Individuums in der Population im Vergleich zur ASAP Lösung in Abhängigkeit von der Generation. Die angegebenen Werte sind die Durchschnittswerte der fünf Läufe. Für die Beispiele 1a, 1b und 3b ist zu erwähnen, dass unter den zufällig generierten Lösungen in der Anfangspopulation bereits ein besseres Individuum zu finden ist.

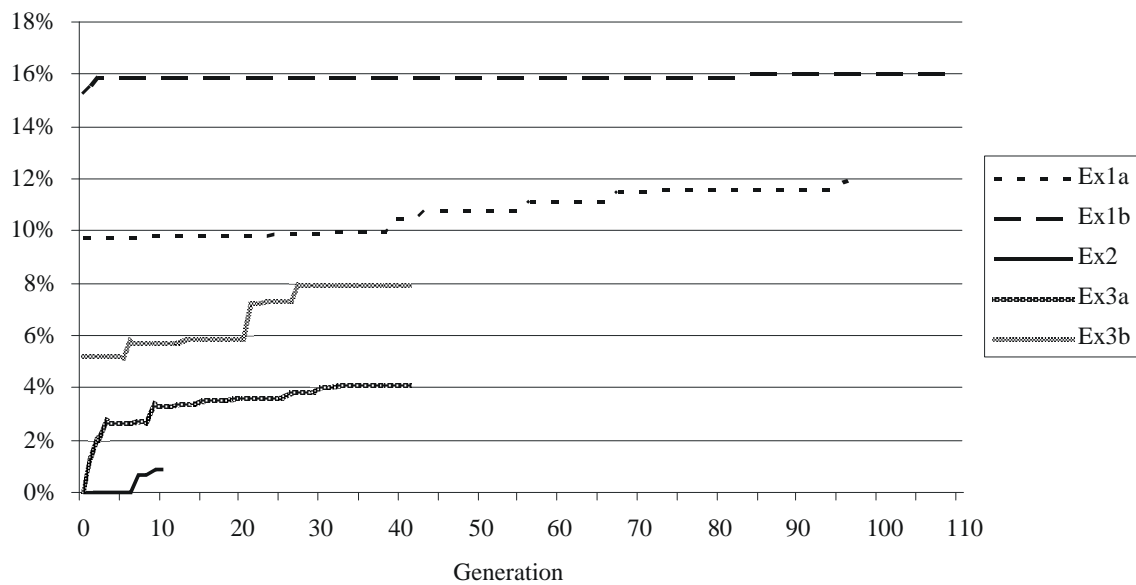


Bild 7: Prozentuale Verbesserung im Vergleich zur ASAP Lösung

9 Ausblick

Zum Abschluss sollen die geplanten Erweiterungen des Planungsalgorithmus angesprochen werden. Bislang arbeitet der EA nur mit zulässigen Lösungen, die Reihenfolgebedingungen, Apparateverfügbarkeit und das Routing der AGVs beachten. Die Berechnung von kollisionsfreien, kürzesten Wegen und die Simulation der Bearbeitungszeiten der Operationen, die durch Differentialgleichungen z. B. zum Aufheizen der Substanzen errechnet werden, sind der zeitaufwendigste Teil in der Evaluierung dieser Lösungen. Daher sollen diese Transfer- und Bearbeitungszeiten lediglich approximiert werden. Dadurch soll die Rechenzeit gesenkt und der Algorithmus effizienter zu guten Lösungen geführt werden. Nach einigen Generationen des EA können die Schätzungen mit exakten Daten aktualisiert werden. Das bedeutet, dass es Generationen mit möglicherweise unzulässigen Individuen gibt, deren Fitness vermutlich überschätzt wird. Dadurch entsteht eine Unsicherheit in der Kostenfunktion des EA. Es ist ein Kompromiss zu finden zwischen präzisen Lösungen und schnell errechneten Lösungen. Die approximierten Transferzeiten sind planabhängig. Ein etwas anderer Plan kann vollkommen andere Transferzeiten zur Folge haben. Daher ist es unter Umständen nicht möglich, planabhängige Schätzungen abzugeben. Deshalb wird als Schätzung einfach der kürzeste Weg ohne Kollisionsbesei-

tigung verwendet, der mit einem Faktor multipliziert wird, der davon abhängt, wie viele AGVs sich zeitgleich in der Anlage befinden.

Die durch den hier beschriebenen EA generierten Lösungen sollen des Weiteren durch einen zweiten EA variiert werden, der für einzelne Rezeptschritte Änderungen an der Apparatebelegung vornimmt. Es wurde angenommen, dass die Reihenfolge der Operationen den größten Einfluss auf den Makespan hat. Jedoch scheint es auch sinnvoll, für einzelne Operationen alternative Apparate zu testen, sobald ein Plan anhand der Reihenfolge errechnet wurde. Dies soll eine Feinplanung darstellen und lokale Verbesserungen liefern. Die Idee ist in Bild 8 dargestellt. Jedes durch den bisherigen EA generierte Individuum wird durch PPSiM evaluiert, d. h. jede Operation erhält einen Behälter, ein AGV und eine Station. Der neue, unterlagerte EA soll dann für eine zufällig ausgewählte Operation alternative Apparate suchen. Dabei ist es nur sinnvoll, das AGV oder die Station zu variieren, da eine Änderung des Behälters nur bei der ersten Operation einer Charge möglich ist und zur Folge hätte, dass alle weiteren Operationen dieser Charge ebenfalls den Behälter wechseln. Wird ein alternativer Apparat gefunden, so wird das Individuum mit dem neuen Apparat ausgewertet. Wenn der Makespan besser ist, wird das geänderte Individuum in die Population aufgenommen; anderenfalls das ursprüngliche Individuum.

Der vorgeschlagene EA zur Variation der Apparatebelegung erfordert hierfür die Entwicklung eines Lückenfüllalgorithmus. Wie in Abschnitt 5 beschrieben, werden bisher nur Apparate verwendet, die zu einem späteren Zeitpunkt noch nicht verplant sind. Dies wird an dieser Stelle jedoch nicht zum Erfolg führen, da es sehr unwahrscheinlich ist, bei einem vollständig berechneten Plan alternative Apparate zu finden, die dieser Bedingung genügen. Der Lückenfüllalgorithmus muss testen, ob eine Operation mit ihrer Routing- und Bearbeitungszeit in eine Lücke passt ohne diese zeitlich nach hinten zu schieben. Es muss beachtet werden, dass das AGV nach der eingefügten Operation an einer anderen Stelle in der Anlage steht und daher das Routing für die nächste Operation neu berechnet werden muss. Das Routing der nächsten Operation auf dem AGV, das ausgetauscht wurde, muss ebenfalls erneut berechnet werden. Da dies relativ aufwändig ist, werden ebenfalls Approximationen benutzt werden müssen.

Danksagung

Das Projekt zur Optimierung des Betriebs von rohrlosen Chemieanlagen wird von der Graduate School of Production Engineering and Logistics des Ministeriums für Innovation, Wissenschaft, Forschung und Technologie des Landes Nordrhein-Westfalen gefördert.

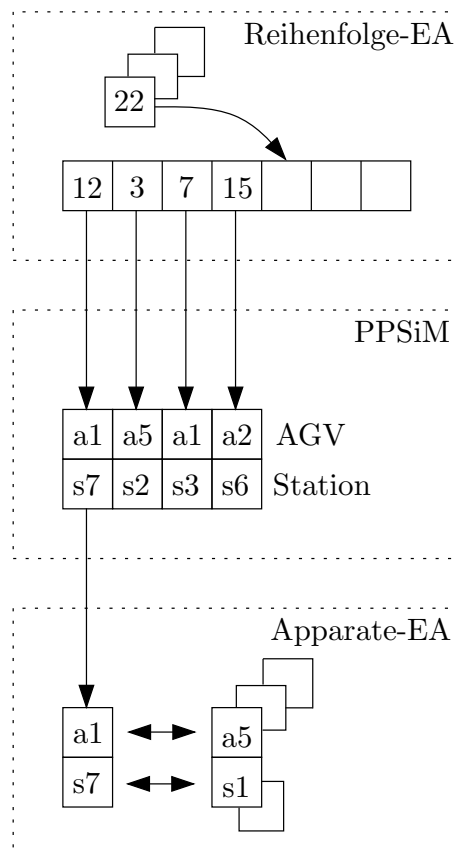


Bild 8: Variation der Apparate

Literatur

- [1] Rauch, J.: *Multiproduct Plants*. Weinheim: Wiley-VCH Verlag. 2003.
- [2] Realff, M.J.; Shah, N.; Pantelides, C.C.: Simultaneous Design, Layout and Scheduling of Pipeless Batch Plants. In: *Computers & Chemical Engineering*, Vol. 20(6/7), S. 869 - 883. 1996.
- [3] Huang, W.; Chung, P.W.H.: Integrating routing and scheduling for pipeless plants in different layouts. In: *Computers & Chemical Engineering*, Vol. 29, S. 1069 - 1081. 2005.
- [4] Patsiatzis, D.I.; Xu, G.; Papageorgiou, L.G.: Layout Aspects of Pipeless Batch Plants. In: *Industrial & Engineering Chemistry Research*, Vol. 44, S. 5672 - 5679. 2005.
- [5] Panek, S.; Engell, S.; Lessner, C.: Scheduling of a pipeless multi-product batch plant using mixed integer programming combined with heuristics. In: *Proc., 15th European Symposium on Computer Aided Process Engineering*, S. 1038 - 1088. 2005.
- [6] Yoo, D.J.; Lee, I.B.; Jung, J.H.: Design of Pipeless Chemical Batch Plants with Queueing Networks. In: *Industrial & Engineering Chemistry Research*, Vol. 44, S. 5630 - 5644. 2005.
- [7] Liefeldt, A.: *Eine Softwareumgebung zur Modellierung, Simulation und Produktionsplanung von Pipeless Plants*. Dissertation, Fachbereich Bio- und Chemieingenieurwesen, Universität Dortmund. 2007.

Pareto Optimization of the Fuzzy c -Means Clustering Model Using a Multi-Objective Genetic Algorithm

Thomas A. Runkler
Siemens AG Corporate Technology
Information and Communications, CT IC 4
81730 München
Tel. 089/636-45372
E-Mail Thomas.Runkler@siemens.com

Abstract

For very high-dimensional data sets such as gene expression data sets in bioinformatics fuzzy clustering often produces trivial solutions where all cluster centers coincide and all memberships are equal. Such an ignorant partition has the minimum partition coefficient, i.e. the cluster validity is minimal. The partition coefficient is maximal for crisp partitions, so fuzzy partitions can never yield the maximum validity. These two conflicting optimization goals of minimizing the fuzzy c -means objective function and maximizing the partition coefficient are joined in a Pareto optimization approach. We find (an approximate subset of) the Pareto optimal set by a multi-objective genetic algorithm, the non-dominated sorting genetic algorithm II (NSGA-II). Experiments with the lung cancer data sets show that NSGA-II finds fuzzy clusters with an acceptable validity.

1 Introduction

In this paper we consider the problem of finding fuzzy clusters in very high-dimensional data. Such high-dimensional data occur, for example, in gene expression data in bioinformatics [6, 11]. The *fuzzy c -means* model [2] has been successfully applied to finding fuzzy clusters in numerous application examples. We show, however, that for very high-dimensional data minimization the fuzzy c -means model will lead to trivial partitions where each object is assigned to all clusters with the same membership. The quality or validity of a fuzzy partition can be quantified by the so-called *partition coefficient* [1]. The partition coefficient is minimal for the trivial solution produced by minimizing the fuzzy c -means model. The partition coefficient is maximal for any crisp partition. However, our goal here is to find fuzzy partitions with a reasonable cluster validity. To jointly optimize the fuzzy c -means model and the partition coefficient a Pareto optimization approach was suggested [16]. In the original work [16] only a quite inefficient greedy algorithm was suggested to find the Pareto optimal set. In this paper we apply a multi-objective genetic algorithm for this purpose: the *non-dominated sorting genetic algorithm* NSGA-II [5]. Experiments with the lung cancer data set show the efficiency of the proposed approach.

This paper is structured as follows: Section 2 briefly introduces the fuzzy c -means model and its reformulation. Section 3 illustrates the problems with clustering high-dimensional data. Section 4 gives an overview of cluster validity measures and particularly reviews the partition coefficient. Section 5 presents the Pareto fuzzy c -means approach. Section

6 discusses different multi-objective genetic algorithms, in particular the non-dominated sorting genetic algorithm NSGA-II. Section 7 presents the experiments with NSGA-II applied to the Pareto fuzzy c-means model for the lung cancer data set. Section 8 finally summarizes the conclusions of this work.

2 Fuzzy c-Means

A popular model for finding clusters in numerical feature data sets is the *fuzzy c-means* (FCM) model [2] defined as the following problem: Given a data set $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^p$, a number of clusters $c \in \{2, \dots, n-1\}$, a fuzziness index $m > 1$, and a norm $|\cdot|$ on \mathbb{R}^p , find a set of prototypes (cluster centers) $V = \{v_1, \dots, v_c\} \subset \mathbb{R}^p$ and a partition matrix $U \in M_{fcn}$ where

$$M_{fcn} = \left\{ U \in [0, 1]^{c \times n} \mid \sum_{i=1}^c u_{ik} = 1, k = 1, \dots, n, \sum_{k=1}^n u_{ik} > 0, i = 1, \dots, c \right\}, \quad (1)$$

by minimizing the objective function

$$J_c(U, V; X) = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m |x_k - v_i|^2. \quad (2)$$

A set of necessary conditions for local extrema or saddle points of J_c specifies the optimal locations of the cluster centers.

$$v_i = \frac{\sum_{k=1}^n u_{ik}^m x_k}{\sum_{k=1}^n u_{ik}^m}, \quad (3)$$

$i = 1, \dots, c$. Inserting (3) into (2) yields a *reformulated* [8] version of the fuzzy c-means objective function.

$$J_c(U; X) = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m \left| x_k - \frac{\sum_{k=1}^n u_{ik}^m x_k}{\sum_{k=1}^n u_{ik}^m} \right|^2 \quad (4)$$

Another set of necessary conditions for local extrema or saddle points of J_c specifies the optimal memberships.

$$u_{ik} = 1 / \sum_{j=1}^c \left(\frac{\left| x_k - \frac{\sum_{k=1}^n u_{ik}^m x_k}{\sum_{k=1}^n u_{ik}^m} \right|^{\frac{2}{m-1}}}{\left| x_k - \frac{\sum_{k=1}^n u_{jk}^m x_k}{\sum_{k=1}^n u_{jk}^m} \right|^{\frac{2}{m-1}}} \right), \quad (5)$$

$i = 1, \dots, c, k = 1, \dots, n$. To find clusters by optimization of the FCM clustering model the partition $U \in M_{fcn}$ is randomly initialized and iteratively updated using (5) until an appropriate termination condition holds. In this paper we will always set $m = 2$ and use the Euclidean distance $|\cdot| = \|\cdot\|$.

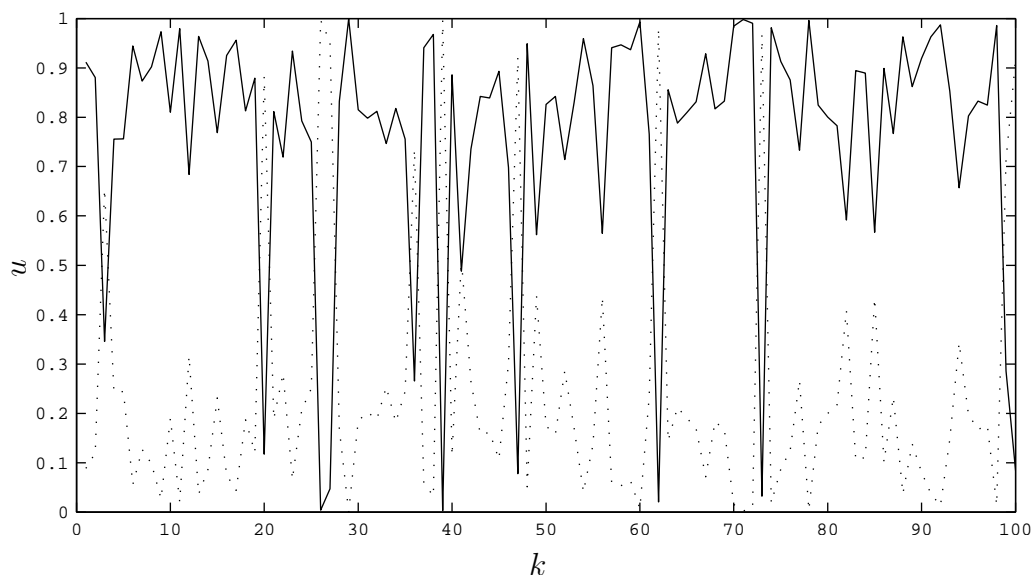


Figure 1: Memberships for 1-dimensional random data.

3 Clustering High-Dimensional Data

The fuzzy c -means model has been successfully applied to numerous data sets. Recently, the scientific community started to show an increasing interest in finding clusters in very high dimensional data sets such as gene expression data sets in bioinformatics [6, 11]. For such high dimensional data sets, however, the fuzzy c -means model proves to be inadequate. To understand this, consider the following experiment that mimics the first step in the optimization of the fuzzy c -means model: We randomly generate a data set X with $n = 100$ $p_{\max} = 10000$ dimensional vectors using a Gaussian distribution with mean zero and variance one. Additionally, we randomly generate a set V of $c = 2$ $p_{\max} = 10000$ dimensional prototypes using the same random distribution. Now we consider only the first $p = 1, \dots, p_{\max}$ dimensions of our data and compute the cluster memberships as

$$u_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{|x_k - v_j|}{|x_k - v_i|} \right)^{\frac{2}{m-1}}}, \quad (6)$$

$i = 1, \dots, c, k = 1, \dots, n$. Fig. 1 shows the resulting memberships for one run of this experiment with $p = 1$ dimensional data. The memberships are distributed in the unit interval $u \in [0, 1]$ with a standard deviation of about $\sigma_u \approx 1/4$. Some points have large memberships ($u_{ik} > 0.9$) in one of the clusters and the next update of the prototypes will be close to these points. Fig. 2 shows the memberships for $p = 10$ dimensional data. The memberships still almost cover the complete unit interval $u \in [0, 1]$, but the standard deviation is reduced to about $\sigma_u \approx 0.1$. Fig. 3 shows the memberships for $p = 100$ dimensional data. Now all the memberships are close to $u = 0.5$, and the standard deviation is only about $\sigma_u \approx 0.03$. In the next step of the FCM optimization these memberships will lead to prototypes that are all almost equal to the mean of all data vectors and hence produce bad clustering results. Fig. 3 shows a double logarithmic plot of the standard deviation of the memberships versus the data dimensionality. Apparently, the illustrated problem becomes quickly more severe with growing data dimensionality. Our examples have shown that high dimensional data tend to have almost equal distances

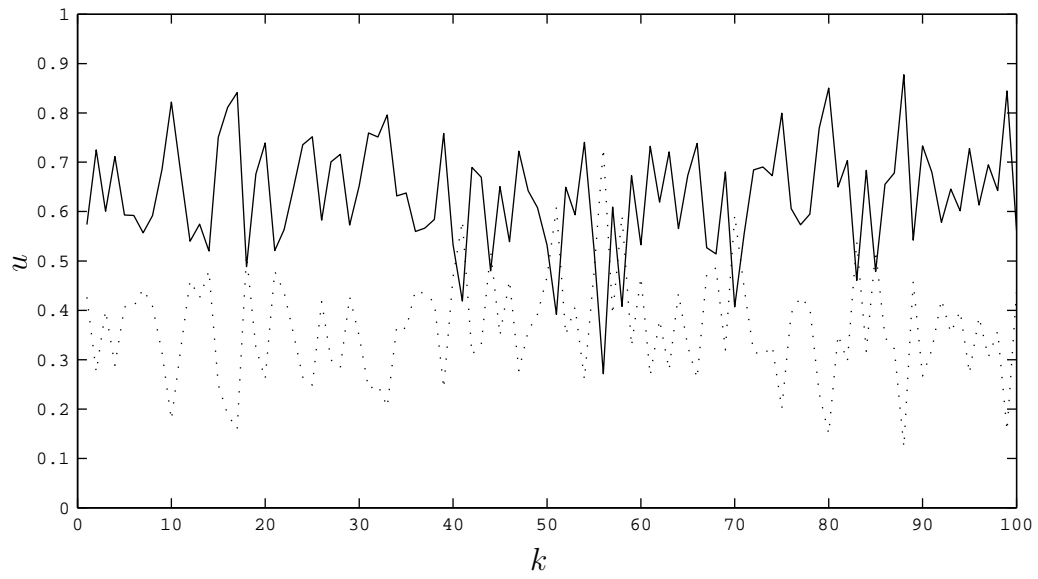


Figure 2: Memberships for 10–dimensional random data.

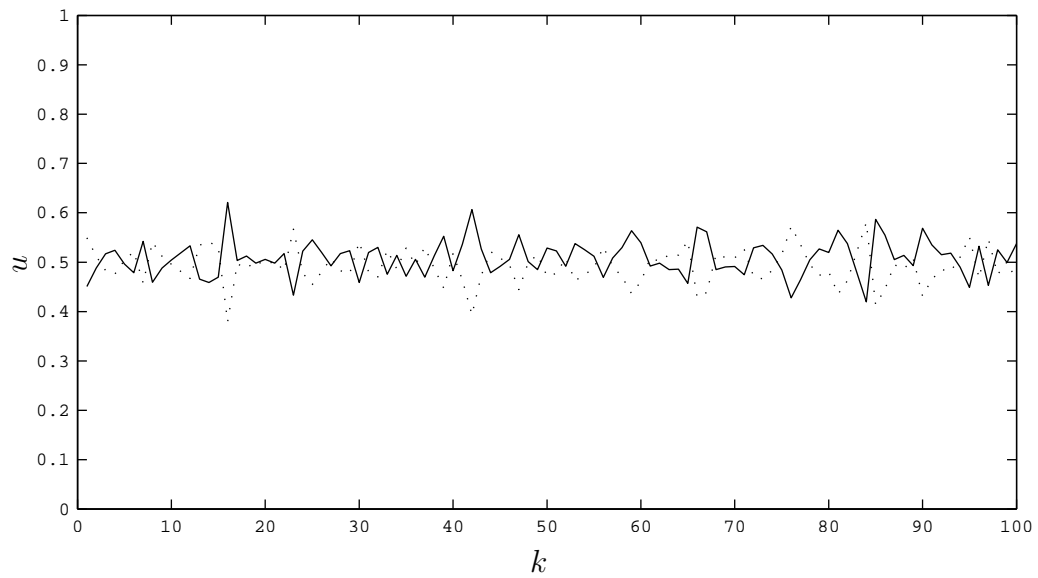


Figure 3: Memberships for 100–dimensional random data.

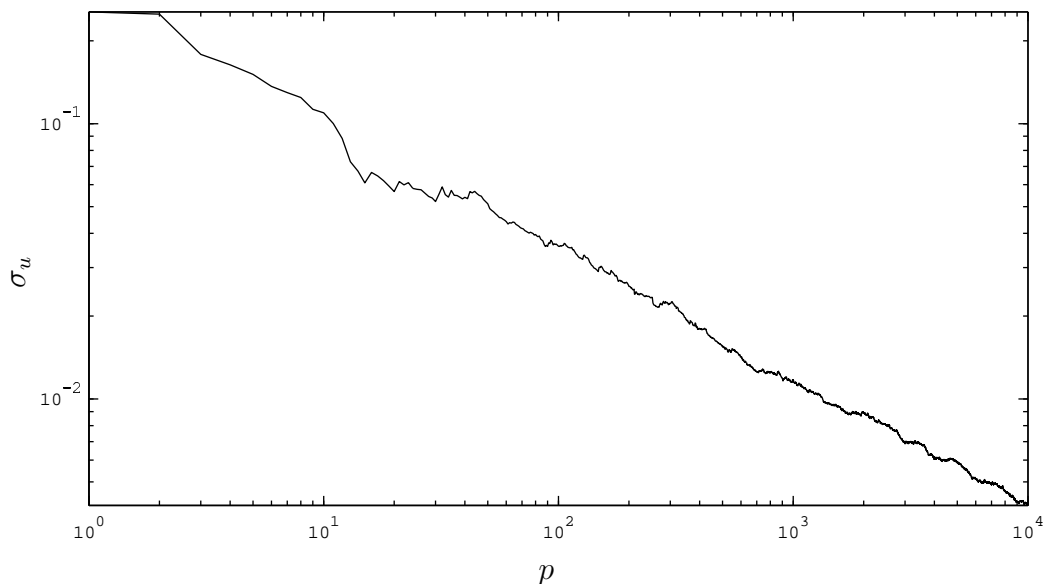


Figure 4: Standard deviation of the memberships versus data dimensionality.

to each other (and to the cluster centers), and therefore the fuzzy c -means clustering model assigns each data point to every cluster with (almost) the same membership of $u_{ik} = 1/c$.

4 Partition Coefficient

The experiments presented in the previous section have shown that for very high-dimensional data the fuzzy c -means clustering model tends to produce memberships of $u_{ik} \approx 1/c$ for all $i = 1, \dots, c, k = 1, \dots, n$. This is the weakest or fuzziest partition possible and obviously does not provide any information about a cluster structure in the data. The quality of a cluster partition can be quantified using *cluster validity measures* such as the Davies-Bouldin index [4], the classification entropy [20], the proportion exponent [20], the cluster height [13], or the Xie-Beni index [21]. In this paper we use the *partition coefficient (PC)* [1] defined as the average quadratic membership:

$$F_c(U) = \frac{1}{n} \sum_{k=1}^n \sum_{i=1}^c u_{ik}^2 \quad (7)$$

For the trivial partition $u_{ik} = 1/c, i = 1, \dots, c, k = 1, \dots, n$, obtained in the examples considered above the partition coefficient becomes $F_{c \min} = 1/c$, which is the minimal value obtainable for any fuzzy partition. The maximum partition coefficient $F_{c \max} = 1$ is achieved when U is a crisp partition, i.e. each object is crisply assigned to exactly one cluster. In fuzzy clustering of high-dimensional data we want to avoid both extremes:

1. The memberships should not all be the same. This means that it is not sufficient to minimize the fuzzy c -means objective function J_c (4).
2. The memberships should not be crisp. This means that it is not sufficient to maximize the partition coefficient F_c (7).

Instead, we want to find a good solution that is associated with a low but not globally minimal value of J_c and a high but not globally maximal value of F_c . Such a compromise solution is found by the *Pareto fuzzy c-means* (PFCM) clustering model [16].

5 Pareto Fuzzy c-Means

Given a set of optimization criteria the set of Pareto optimal solutions is defined as the set of solutions for which no criterion can be improved without deteriorating at least one other criterion. The Pareto fuzzy c-means model [16] uses two criteria: the fuzzy c-means objective function $J_c(U; X)$ (4) and the *negative* partition coefficient $-F_c(U; X)$ (7). For convenience we use the negative instead of the positive partition coefficient here, so that both criteria are minimized, not maximized. So, Pareto fuzzy c-means clustering finds a set of partitions that is a finite subset of all Pareto optimal partitions

$$P_{JF_c} = \left\{ \begin{array}{l} U \in M_{fcn} \mid \\ \neg \exists U' \in M_{fcn} \\ J_c(U'; X) < J_c(U; X) \quad \wedge \\ -F_c(U'; X) < -F_c(U; X) \end{array} \right\} \quad (8)$$

Many methods have been suggested to compute the set of Pareto optimal solutions for different applications. In the context of fuzzy clustering several stochastic optimization methods have been applied, such as *particle swarm optimization* [17], ant colony optimization [15], or *genetic algorithms* [3]. In this paper, we consider a genetic algorithm for Pareto optimization. Notice the difference between applying multi-objective genetic algorithms to clustering and applying clustering techniques to improve multi-objective genetic algorithms [12, 18]!

6 Multi-Objective Genetic Algorithm

Some of the most popular genetic algorithms for Pareto optimization are the multi-objective genetic algorithm (MOGA) [7], the similar non-dominated sorting genetic algorithm NSGA [19], the strength Pareto evolutionary algorithm (SPEA) [23], its extension SPEA2 [22], and the fair evolutionary multi-objective optimizer (FEMO) [10]. In this paper we use a more efficient variant of NSGA called NSGA-II [5]. In all of these algorithms each individual of the population represents one solution, so the algorithms produce a whole set of (approximate) representatives of the Pareto optimal set. All of these algorithms produce the solutions by (more or less standard) selection, crossover, and mutation operators. The main difference in these algorithms is the computation of the fitness of the individual solutions. The basic idea of MOGA, NSGA, and NSGA-II is to consider the number of solutions that *dominate* a certain solution, i.e. the number of solutions that possess at least one worse criterion. The so-called *domination class* of each solution is used as the fitness parameter for the selection process. In this way, non-dominated solutions, in particular those that (approximately) belong to the Pareto optimal set, receive a higher probability to be selected. An important goal of multi-objective algorithms is to produce *diverse* solutions, i.e. the solutions should not be too similar but cover the complete space of Pareto optimal solutions. To assure the diversity of the solution so-called

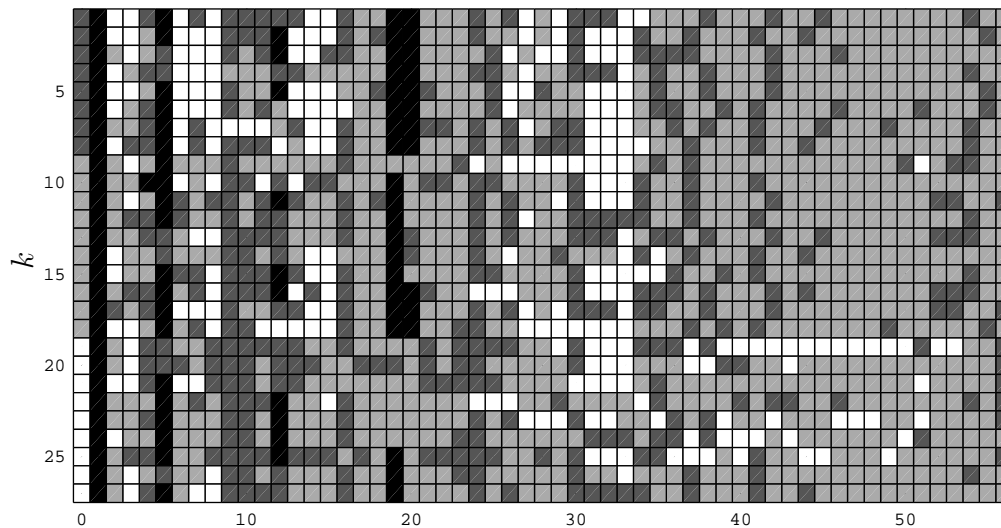


Figure 5: The lung cancer data set.

niching strategies are employed. The niching strategy of NSGA-II computes the fitness of each solution by combination of the fitness described above with the so-called *niching count* defined as the number of other solutions in the vicinity of the considered solution.

In order to apply NSGA-II to the problem of optimizing the Pareto fuzzy c-means model each solution is specified by a particular partition matrix $U \in M_{fcn}$. The main algorithmic parameters to be specified are q , the number of individuals in each generation, and t , the number of generations for the algorithm to run. After t generations the NSGA-II algorithm then produces a set of q partition matrices M_1^*, \dots, M_q^* that approximate the Pareto fuzzy c-means solution.

7 Experiments

In this paper we consider a typical bioinformatics data set called the *lung cancer data set* [9] which can be obtained from the *UCI Machine Learning Repository*. The lung cancer data set contains 32 vectors with 56 features. Each feature can have the discrete values $\{0, 1, 2, 3\}$. Each object belongs to one of $c = 3$ different classes associated with pathological lung cancers. The original data set contains five objects with unknown feature values: $x_1, x_{15}, x_{19}, x_{21}, x_{26}$. These objects are ignored here, so the remaining feature vectors form a 27 by 56 matrix X . This feature matrix X is shown in Fig. 5. Each row represents a feature vector of an object, and each column represents a feature. Column 0 shows the class information. The grey values represent the numerical values 0 (black), 1 (dark grey), 2 (light grey), and 3 (white). Since we are considering the problem of clustering not classification, the class information in column 0 is ignored. However, our hope is that the clusters approximately match the given class labels (that are not provided to the clustering algorithm). Notice that the lung cancer data set is a very high dimensional data set, as discussed before. In particular, we have $p > n$ for this data set, i.e. the number of dimensions is higher than the number of objects. For $p > n$ it is easy to build a classifier with a low training error, but it is hard to build a classifier with a low generalization error.

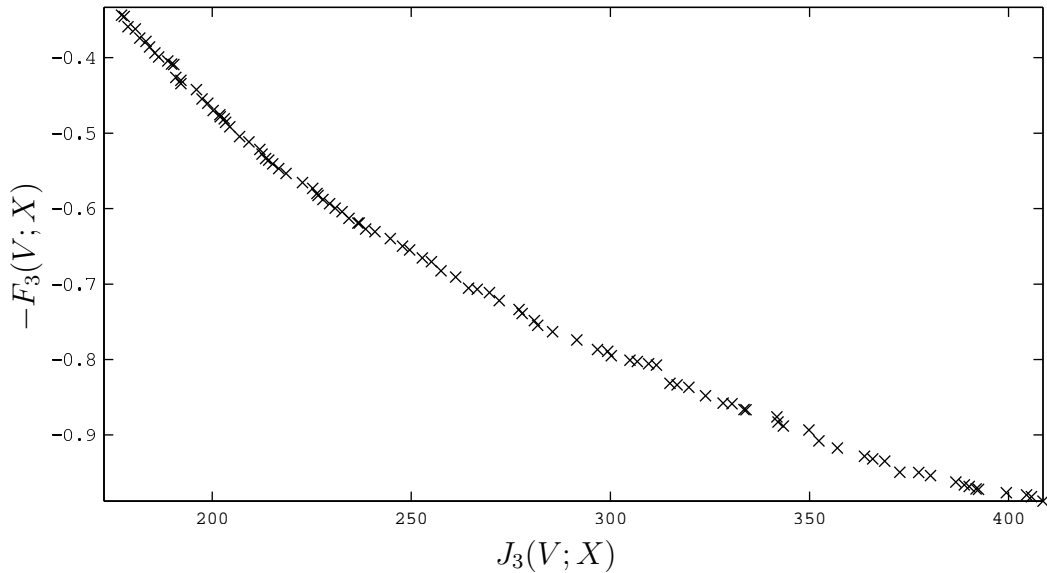


Figure 6: The set of 100 Pareto optimal clustering solutions.

Moreover, as we have seen above, for $p > n$ optimization of the fuzzy c -means model only produces the trivial partition $u_{ik} = 1/c$, $i = 1, \dots, n$, $k = 1, \dots, n$, associated with a c -fold cluster center at

$$V = \left\{ \frac{1}{n} \sum_{k=1}^n x_k \right\} \quad (9)$$

For the lung cancer data set this solution is associated with the numerical values $J_3 \approx 172.7654$ and $F_3 = 1/3$. No partition will yield a lower value of J_3 , but our goal now is to find partitions U that yield higher values of $F_3(U; X)$ and at the same time (approximately) yield the minimum possible values of $J_3(U; X)$, hoping that these partitions represent non-trivial and more useful clustering results. To do so, we apply NSGA-II to optimize the Pareto fuzzy c -means model. For NSGA-II we use the following algorithmic parameters: The number of individuals in each generation is set to $q = 100$, and the number of generations for the algorithm to run is set to $t = 1000$. Since also the initial individuals are evaluated, the algorithm performs a total of $(t + 1) \cdot q = 100100$ evaluations of the objective functions $J_3(U; X)$ and $F_3(U; X)$. Fig. 6 shows the values of the objective functions J_3 and $-F_3$ for the $q = 100$ resulting partitions. The 100 solutions are almost evenly distributed along the Pareto optimal set which is apparently a curve section between the extreme edges $(J_{3 \min}, F_{3 \min}) \approx (170, 0.3)$ and $(J_{3 \max}, F_{3 \max}) \approx (410, 1)$. The solution $(J_{3 \min}, F_{3 \min})$ corresponds with the previously reported solution obtained by minimization of J_3 alone, i.e. by the (regular) fuzzy c -means model. The corresponding memberships for this solution are shown in Fig. 7. As already illustrated in Section 3 all membership values are very close to $1/c = 1/3$ with a maximum deviation of only about 0.02. So, effectively this (almost) trivial solution does not provide any useful information about the cluster structure in the lung cancer data set. Next, we consider the opposite extreme $(J_{3 \max}, F_{3 \max})$ with a maximum partition coefficient of $F_{3 \max} \approx 1$. As pointed out in Section 4 every crisp partition yields a partition coefficient of 1. Among the $c^n = 3^{27} \approx 8 \cdot 10^{12}$ possible crisp partitions Pareto optimization theoretically finds the one with the minimum value of the fuzzy c -means objective function J_3 . All other crisp partitions are dominated by this Pareto optimal solution. The memberships of the

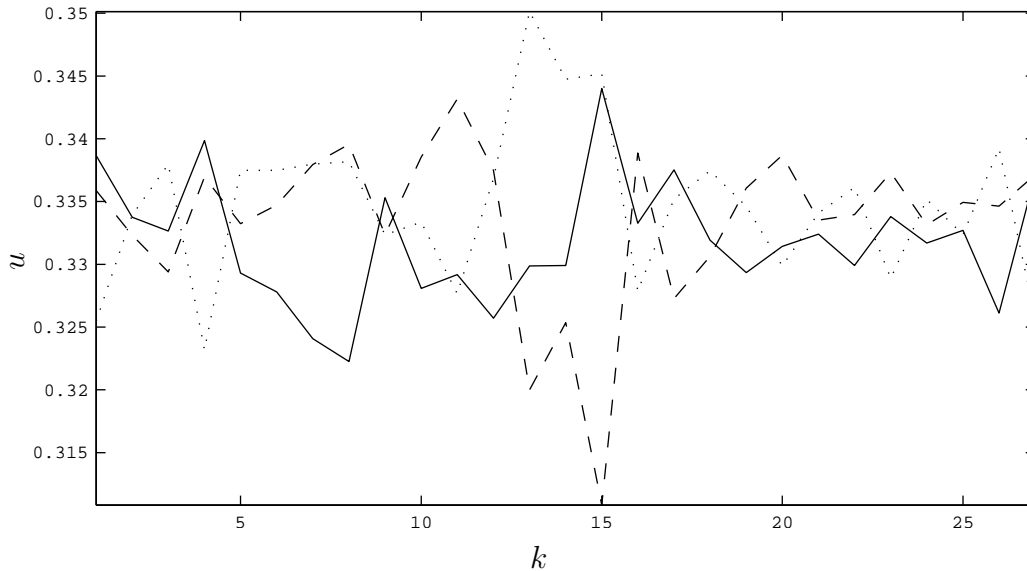


Figure 7: Memberships for minimal J_3 .

solution found by NSGA-II are shown in Fig. 8. Apparently, the cluster validity is not exactly equal to one, but only close to one, so the partitions are not exactly crisp. However, the deviations of the memberships from zero or one are only in the order of about 0.05, so the partition can be viewed as *approximately* crisp. The resulting partition could be transformed into a real crisp solution by defuzzification [14], but we do not pursue this aspect here. Following the original class labels (as shown in the leftmost column in Fig. 5), objects 1–8 belong one class, objects 9–18 belong to another class, and objects 19–27 belong to the third class. We can interpret the cluster with the dashed membership function in Fig. 8 as the first class, so objects 2–6 are correctly classified, which corresponds to a true positive classification rate of $5/8 = 62.5\%$. Accordingly, the solid membership function can be interpreted as the third class, so objects 19–24 are correctly classified, which corresponds to a true positive classification rate of $6/9 \approx 66.7\%$. Then, however, the dotted membership function has to be interpreted as the second class, so only objects 13 and 17 are correctly classified, which corresponds to a true positive classification rate of only $2/10 = 20\%$. So, the classification rate obtained by the Pareto optimal solution with maximal J_3 is definitively better than that for minimal J_3 , but the classification accuracy is not satisfactory. Moreover, we would prefer a fuzzy partition instead of a crisp one in order to see to which degree a certain object is representative for each cluster (class). Fig. 9 shows the Pareto optimal solution with the median value of J_3 which represents a compromise between minimal J_3 (Fig. 7, c -fold prototype) and maximal J_3 (Fig. 8, crisp). Again the dashed membership function can be associated with class 1, dotted with class 2, and solid with class 3. A visual comparison between Figs. 7 and 9 shows that the defuzzified classification rates are comparable. However, the median J_3 solution is fuzzy and therefore indicates which of the objects are most representative for each class. If we consider only memberships $u > 0.7$, then we can associate each class with the objects given in the following table:

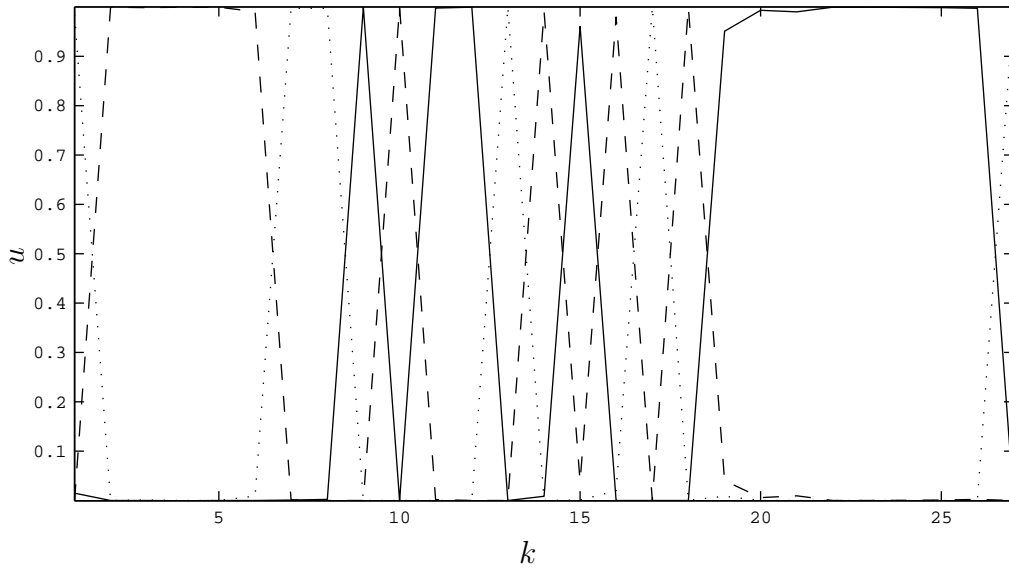


Figure 8: Memberships for maximal J_3 .

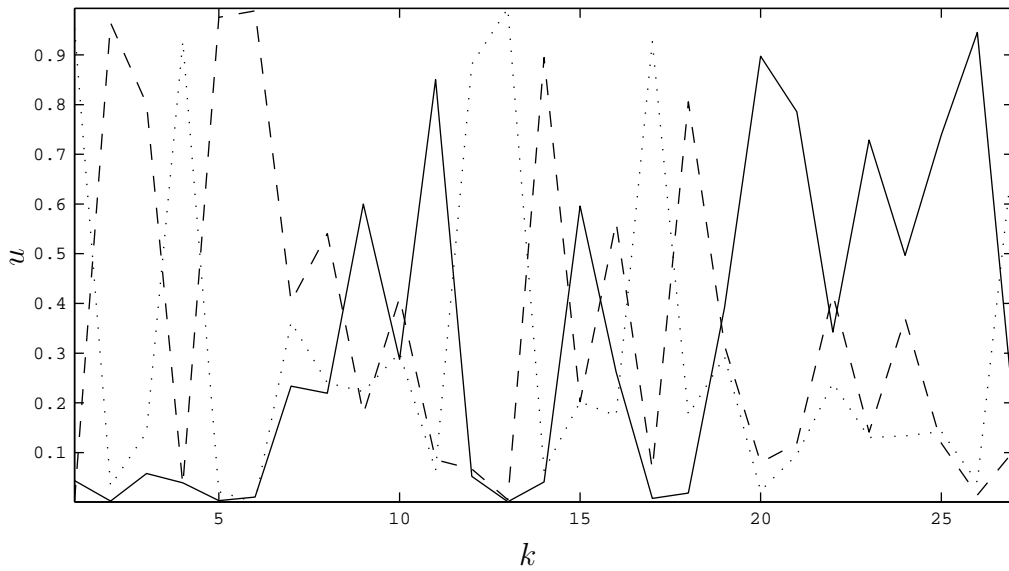


Figure 9: Memberships for median J_3 .

class	style	objects
1	dashed	2, 3, 5, 6 , 14, 18
2	dotted	1, 4, 12, 13, 17 , 27
3	solid	11, 20, 21, 23, 25, 26

Objects whose cluster assignments match the original class labels are printed in bold, indicating an acceptable concordance between cluster-based and expert classification.

8 Conclusions

We have considered the problem of finding clusters in high-dimensional data sets, i.e. those where the number of feature dimensions is much higher than the number of objects. For such high-dimensional data it is easy to build classifiers with a low training error, but it is difficult to build classifiers with a low generalization error. When we apply the fuzzy c-means clustering model to very high-dimensional data then we encounter the following problem: The distances between data points and cluster centers are almost equal, so all the memberships become equal to one divided by the number of clusters. Therefore, all cluster centers collapse to one single cluster center at the centroid of all the data points. So, optimizing the fuzzy c-means model yields only a trivial solution for very high-dimensional data sets. To overcome this problem we also considered a cluster validity measure: the partition coefficient. For the trivial solution found by optimizing the fuzzy c-means model the partition coefficient yields its minimum, so minimizing the fuzzy c-means objective function and maximizing the partition coefficient are mutually exclusive optimization goals. The partition coefficient yields its maximum for crisp partitions, so maximizing the partition coefficient and finding fuzzy clusters are also mutually exclusive optimization goals. We overcome this problem by searching for the Pareto optimal solution for the two conflicting optimization goals. The elements of the Pareto optimal set have the property that no optimization goal can be improved any more without deteriorating at least one other optimization goal. Pareto optimization therefore produces a complete set of solutions that form an approximate subset of the Pareto optimal set. For Pareto optimization we employ a multi-objective genetic algorithm, the non-dominated sorting genetic algorithm II (NSGA-II). For the lung cancer data set NSGA-II finds solutions that are nicely distributed along the Pareto front between the two extremes of ignorant and crisp partitions. The partition at the median of the Pareto front nicely exhibits the fuzzy cluster structure in the data. It approximately corresponds with the originally given class labels (that were of course not used in clustering), and it allows to easily identify the most representative objects for each class.

References

- [1] J. C. Bezdek. Numerical taxonomy with fuzzy sets. *Journal of Mathematical Biology*, 1(1):57–71, 1974.
- [2] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

- [3] J. C. Bezdek and R. J. Hathaway. Optimization of fuzzy clustering criteria using genetic algorithms. In *IEEE Conference on Evolutionary Computation, Orlando*, volume 2, pages 589–594, June 1994.
- [4] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1:224–227, 1979.
- [5] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Parallel Problem Solving from Nature*, pages 849–858, Paris, France, 2000.
- [6] M. Dejori and M. Stetter. Identifying interventional and pathogenic mechanisms by generative inverse modeling of gene expression profiles. *Journal of Computational Biology*, 11(6):1135–1148, 2004.
- [7] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *International Conference on Genetic Algorithms, Urbana–Champaign, USA*, pages 416–423, June 1993.
- [8] R. J. Hathaway and J. C. Bezdek. Optimization of clustering criteria by reformulation. *IEEE Transactions on Fuzzy Systems*, 3(2):241–245, May 1995.
- [9] Z. Q. Hong and J. Y. Yang. Optimal discriminant plane for a small number of samples and design method of classifier on the plane. *Pattern Recognition*, 24(4):317–324, May 1991.
- [10] M. Laumanns, L. Thiele, E. Zitzler, E. Welzl, and K. Deb. Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem. In *Parallel Problem Solving From Nature, Granada, Spain*, pages 117–120, 2002.
- [11] M. Popescu, J. M. Keller, and J. Mitchell. Fuzzy measures on the gene ontology for gene product similarity. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(3):263–274, July 2006.
- [12] G. T. Pulido and C. A. C. Coello. Using clustering techniques to improve the performance of a multi-objective particle swarm optimizer. In *Genetic and Evolutionary Computation Conference*, pages 225–237, Seattle, June 2004.
- [13] T. A. Runkler. Improved cluster validity measures for the fuzzy c-elliptotype algorithm. In *European Congress on Intelligent Techniques and Soft Computing*, pages 1331–1335, Aachen, August 1995.
- [14] T. A. Runkler. Selection of appropriate defuzzification methods using application specific properties. *IEEE Transactions on Fuzzy Systems*, 5(1):72–79, 1997.
- [15] T. A. Runkler. Ant colony optimization of clustering models. *International Journal of Intelligent Systems*, 20(12):1233–1261, December 2005.
- [16] T. A. Runkler. Pareto optimality of cluster objective and validity functions. In *IEEE International Conference on Fuzzy Systems*, London, July 2007.
- [17] T. A. Runkler and C. Katz. Fuzzy clustering by particle swarm optimization. In *IEEE International Conference on Fuzzy Systems*, pages 3065–3072, Vancouver, July 2006.

- [18] M. R. Sierra and C. A. C. Coello. Coevolutionary multi-objective optimization using clustering techniques. In *Mexican International Conference on Artificial Intelligence*, pages 603–612, Monterrey, November 2005.
- [19] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.
- [20] M. P. Windham. Cluster validity for the fuzzy c-means clustering algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-4(4):357–363, July 1982.
- [21] X. L. Xie and G. Beni. A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):841–847, August 1991.
- [22] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, Zurich, Switzerland, 2001.
- [23] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

Die Optimierung von Fuzzy-Zielfunktionen in Fuzzy (Mehrziel-) LP-Systemen - Ein kritischer Überblick

Heinrich J. Rommelfanger

J. W. Goethe Universität Frankfurt am Main,
Professur für Wirtschaftsmathematik
Mertonstr. 17-23, D-60054 Frankfurt am Main, Germany
Tel. (069)798 22268; Fax (069)798 23432
E-Mail: Rommelfanger@wiwi.uni-frankfurt.de

1 Einleitung

Klassische Programmierungsmodelle benötigen eindeutig bestimmte Koeffizienten und exakt festgelegte Restriktionsgrenzen. Um eine Fehlmodellierung zu vermeiden, ist daher in der Regel eine umfangreiche Informationsaufnahme und -verarbeitung notwendig. Oft wird man dennoch bei Realproblemen einige der Modellparameter nur größenordnungsmäßig angeben können. Während in den klassischen Modellen nur der Weg bleibt, diese ungenauen Größen durch "Mittelwerte" zu ersetzen, bieten Fuzzy-Modelle die Möglichkeit, die subjektiven Vorstellungen eines Entscheiders so präzise zu modellieren, wie dieser es ausdrücken will und kann. Das Risiko, mit einem falschen Bild der Realität zu arbeiten und Lösungen auszuwählen, die nicht dem Realproblem entsprechen, wird somit deutlich reduziert.

Beschränken wir die Betrachtung auf den am häufigsten benutzten Modelltyp, die Linearen Programmierungsmodelle, so lässt sich eine Fuzzy-Erweiterung allgemein durch das nachfolgende Fuzzy Lineare Programmierungs-Modelle (FLP-Modell) ausdrücken.

$$\begin{aligned} \tilde{Z}(\mathbf{x}) &= \tilde{C}_1 x_1 \oplus \dots \oplus \tilde{C}_n x_n \rightarrow M\tilde{\alpha}x \\ \text{unter Beachtung der Restriktionen} & \hspace{15em} (1) \\ \tilde{A}_{i1}x_1 \oplus \tilde{A}_{i2}x_2 \oplus \dots \oplus \tilde{A}_{in}x_n &\lesssim \tilde{B}_i, \quad i = 1, \dots, m, \\ x_1, x_2, \dots, x_n &\geq 0 \end{aligned}$$

Dabei sind $\tilde{A}_{ij}, \tilde{B}_i, \tilde{C}_j, i = 1, \dots, m, j = 1, \dots, n$ Fuzzy-Mengen in \mathbf{R} . Das Symbol \oplus steht für erweiterte Addition.

Da jede reelle Zahl als eine Fuzzy-Zahl modelliert werden kann, enthält das Fuzzy Lineare Programmierungssystem (1) die Spezialfälle, dass die Zielfunktion und/oder Restriktionen reelle Koeffizienten aufweisen; ebenso können Restriktionsgrenzen durch reelle Zahlen eindeutig festgelegt sein.

Die Verwendung von FLP-Systemen hat einerseits den Vorteil, dass ein Entscheider ein vorliegende Realproblem im Einklang mit seinem Informationsstand in ein mathematisches Modell abbilden kann. Andererseits kann er aber die bekannten Algorithmen zur Lösung von LP-Systemen nicht mehr zur Berechnung einer optimalen Lösung benutzen. Daher wurden in den letzten 30 Jahren zahlreiche Algorithmen zur Lösung von FLP-Modellen entwickelt, vgl. [13].

Um den Überblick nicht ausufern zu lassen, liegt der Fokus dieser Arbeit auf der Frage, wie die verschiedenen Lösungsansätze die Fuzzy-Zielfunktion interpretieren. Die Menge der zulässigen Lösungen kann dabei durch klassische Restriktionen, Restriktionen mit weichen Grenzen oder Fuzzy-Restriktionen beschrieben sein. Im letzteren Fall wird jeweils auch kurz auf die Interpretation dieser Beschränkungen eingegangen.

Um den Methodenvergleich übersichtlich zu gestalten, wird angenommen, dass alle Fuzzy-Koeffizienten \tilde{C}_j und \tilde{A}_{ij} durch L-R-Fuzzy-Intervalle modelliert werden können, d. h. $\tilde{C}_j = (\underline{c}_j; \bar{c}_j; \underline{\gamma}_j; \bar{\gamma}_j)_{LR}$; $\tilde{A}_{ij} = (\underline{a}_{ij}; \bar{a}_{ij}; \underline{\alpha}_{ij}; \bar{\alpha}_{ij})_{L_i R_i}$. Zwar ist bei vielen Methoden diese Annahme nicht notwendig, aber sowohl in der Literatur als auch in praktischen Anwendungen wird diese Annahme üblicherweise getroffen. Meistens unterstellt man sogar lineare Referenzfunktionen $L(u) = R(u) = \text{Max}(0, 1 - u)$.

Durch Anwendung des Zadeh'schen Erweiterungsprinzips kann dann die Fuzzy-Zielfunktion $\tilde{Z}(\mathbf{x}) = \tilde{C}_1 x_1 \oplus \dots \oplus \tilde{C}_n x_n$ zu dem L-R-Fuzzy-Intervall

$$\tilde{Z}(\mathbf{x}) = (\underline{c}(\mathbf{x}); \bar{c}(\mathbf{x}); \underline{\gamma}(\mathbf{x}); \bar{\gamma}(\mathbf{x}))_{LR} \quad (2)$$

aggregiert werden, wobei gilt

$$\underline{c}(\mathbf{x}) = \sum_{j=1}^n \underline{c}_j x_j, \quad \bar{c}(\mathbf{x}) = \sum_{j=1}^n \bar{c}_j x_j, \quad \underline{\gamma}(\mathbf{x}) = \sum_{j=1}^n \underline{\gamma}_j x_j, \quad \bar{\gamma}(\mathbf{x}) = \sum_{j=1}^n \bar{\gamma}_j x_j.$$

2 Mehrzielansatz

Nach den vorstehenden Ausführungen ist offensichtlich, dass eine Fuzzy-Zielfunktion $\tilde{Z}(\mathbf{x}) = \tilde{C}_1 x_1 \oplus \dots \oplus \tilde{C}_n x_n \rightarrow \tilde{\text{Max}}$ als ein Mehrzielproblem zu interpretieren ist.

So entspricht in dem einfachen Fall mit Koeffizienten der Form $\tilde{C}_j = (\underline{c}_j; \bar{c}_j; \underline{\gamma}_j; \bar{\gamma}_j)_{LR}$ die Fuzzy-Zielfunktion $\tilde{Z}(\mathbf{x}) = (\underline{c}(\mathbf{x}); \bar{c}(\mathbf{x}); \underline{\gamma}(\mathbf{x}); \bar{\gamma}(\mathbf{x}))_{LR}$ den vier crisp Zielen

$$\underline{c}(\mathbf{x}) - \underline{\gamma}(\mathbf{x}) \rightarrow \text{Max}, \quad \underline{c}(\mathbf{x}) \rightarrow \text{Max}, \quad \bar{c}(\mathbf{x}) \rightarrow \text{Max}, \quad \bar{c}(\mathbf{x}) + \bar{\gamma}(\mathbf{x}) \rightarrow \text{Max} \quad (3)$$

die simultan über der Menge der zulässigen Lösungen X_U zu maximieren sind.

A. Lai; Hwang [6] betrachten den Spezialfall mit triangularen Koeffizienten $\tilde{C}_j = (c_j; \underline{\gamma}_j; \bar{\gamma}_j)$ und ersetzen die Fuzzy-Zielfunktion $\tilde{Z}(\mathbf{x}) = (c(\mathbf{x}); \underline{\gamma}(\mathbf{x}); \bar{\gamma}(\mathbf{x}))_{LR}$ durch die drei Zielfunktionen:

$$z_1(\mathbf{x}) = c(\mathbf{x}) \rightarrow \text{Max}, \quad z_2(\mathbf{x}) = \underline{\gamma}(\mathbf{x}) \rightarrow \text{Min}, \quad z_3(\mathbf{x}) = \bar{\gamma}(\mathbf{x}) \rightarrow \text{Max}. \quad (4)$$

Zur Lösung dieses Mehrzielproblems verwenden Lai; Hwang einen Vorschlag von Zimmermann [20] und substituieren diese drei Zielfunktionen durch lineare Fuzzy-Zielfunktionen. Da diese auf der Basis der besten und schlechtesten Lösung der Zielfunktionen (4) zu konstruieren sind, schlagen die Autoren die folgenden Werte vor:

$$\begin{aligned} z_1^{\text{PIS}} &= c(\mathbf{x}_I) = \text{Max}_{x \in X_U} c(\mathbf{x}) & z_2^{\text{PIS}} &= \underline{\gamma}(\mathbf{x}_{II}) = \text{Min}_{x \in X_U} \underline{\gamma}(\mathbf{x}) & z_3^{\text{PIS}} &= \bar{\gamma}(\mathbf{x}_{III}) = \text{Max}_{x \in X_U} \bar{\gamma}(\mathbf{x}) \\ z_1^{\text{NIS}} &= \text{Min}_{x \in X_U} c(\mathbf{x}) & z_2^{\text{NIS}} &= \text{Max}_{x \in X_U} \underline{\gamma}(\mathbf{x}) & z_3^{\text{NIS}} &= \text{Min}_{x \in X_U} \bar{\gamma}(\mathbf{x}). \end{aligned}$$

Diese Auswahl, speziell die Werte z_1^{NIS} , z_2^{NIS} und z_3^{NIS} können nicht überzeugen, da die Werte z_1^{NIS} und z_3^{NIS} im Allgemeinen zu klein und der Wert z_2^{NIS} zu groß gewählt werden. Damit sind aber die linearen Fuzzy-Zielfunktionen fehlerhaft modelliert. Es wäre besser, die nachfolgenden Werte zu verwenden, die auch in der Literatur üblich sind:

$$\begin{aligned} z_1^{\text{NIS}} &= \text{Min}_{x \in X_U} [c(\mathbf{x}_{\text{II}}), c(\mathbf{x}_{\text{III}})] & z_2^{\text{NIS}} &= \text{Max}_{x \in X_U} [\underline{\gamma}(\mathbf{x}_{\text{I}}), \underline{\gamma}(\mathbf{x}_{\text{III}})] \\ z_3^{\text{NIS}} &= \text{Min}_{x \in X_U} [\bar{\gamma}(\mathbf{x}_{\text{I}}), \bar{\gamma}(\mathbf{x}_{\text{II}})] \end{aligned}$$

3 Auf Defuzzifizierung basierende Kompromiss-Zielfunktion(en)

B. Die erste Arbeit zu LP-Systemen mit einer Fuzzy-Zielfunktion wurde von Tanaka; Ichihashi; Asai [18] veröffentlicht. Sie ersetzen $\tilde{Z}(\mathbf{x}) = \tilde{C}_1 x_1 \oplus \dots \oplus \tilde{C}_n x_n$ durch die crise Kompromisszielfunktion

$$Z_1(\mathbf{x}) = \frac{1}{4} \cdot \sum_{j=1}^n (2\underline{c}_j + 2\bar{c}_j - \underline{\gamma}_j + \bar{\gamma}_j) \cdot x_j \rightarrow \text{Max} \quad (5)$$

Jede Fuzzy-Restriktion $\tilde{A}_i(\mathbf{x}) = (\underline{a}_i(\mathbf{x}); \bar{a}_i(\mathbf{x}); \underline{\alpha}_i(\mathbf{x}); \bar{\alpha}_i(\mathbf{x}))_{\text{LR}} \tilde{\leq} \tilde{B}_i$, $i = 1, \dots, m$, substituieren diese Autoren durch die crise Bedingungsungleichung

$$\tilde{B}_i \ominus \tilde{A}_i(\mathbf{x}) \tilde{\geq} 0 \text{ ist fast positiv auf dem Niveau } h \in [0, 1],$$

welche äquivalent ist zu der pessimistischen ρ -Präferenzrelation

$$\sum_{j=1}^n (\bar{a}_{ij}(\mathbf{x}) + \bar{\alpha}_{ij}(\mathbf{x}) \cdot R_{A_i}^{-1}(\rho)) \leq b_i - \underline{\beta}_i \cdot R_{B_i}^{-1}(\rho) \quad \text{mit } \rho = 1 - h. \quad (6)$$

Leider übersehen die Autoren, dass in einer „ $\tilde{\leq}$ “-Relation die Restriktionsgrenze die Form $\tilde{B}_i = (b_i, 0, \bar{\beta}_i)_{\text{LR}}$ aufweisen muss, d. h. die linken Spannweiten sind gleich Null zu setzen.

C. Um die Robustheit der Lösung besser steuern zu können, ergänzt Stanciulescu [17] die crise Zielfunktion (5) durch die folgenden drei Zielfunktionen, welche unterschiedliche Aspekte der Ungenauigkeit ausdrücken:

“Gesamteindruck der Ungenauigkeit eines Fuzzy-Intervalls” (*Impression attached to any fuzzy data*)

$$Z_2(\mathbf{x}) = \frac{1}{2} \cdot \sum_{j=1}^n (2\bar{c}_j + \bar{\gamma}_j - 2\underline{c}_j - \underline{\gamma}_j) \cdot x_j \rightarrow \text{Min} \quad (7)$$

“Fuzziness eines Fuzzy-Intervalls” (*Fuzziness of a fuzzy data*)

$$Z_3(\mathbf{x}) = \frac{1}{2} \cdot \sum_{j=1}^n (\bar{\gamma}_j + \underline{\gamma}_j) \cdot x_j \rightarrow \text{Min} \quad (8)$$

“Ungenauigkeit eines Fuzzy-Intervalls” (*Variety of a fuzzy data*)

$$Z_4(\mathbf{x}) = \frac{1}{2} \cdot \sum_{j=1}^n (\bar{c}_j - \underline{c}_j) \cdot x_j \rightarrow \text{Min} \quad (9)$$

Die Verarbeitung von Fuzzy-Restriktionen und Rsetriktionen mit weichen Grenzen erfolgt in [17] gemäß der $\tilde{\leq}_R$ -Relation von Rommelfanger [10], vgl. Formel (33).

Um eine Lösung dieses Mehrzieloptimierungssystems zu erhalten, verwendet Stanciu-lescu das MAUT-Verfahren, bei dem zunächst jede Zielfunktion $Z_k(\mathbf{x})$ in eine Nutzenfunktion $u_k(Z_k(\mathbf{x}))$ transformiert wird und dann die Gesamtzielfunktion

$$U(\mathbf{x}) = \sum_{k=1}^4 u_k(Z_k(\mathbf{x})) \rightarrow \text{Max} \quad (10)$$

über der verbleibenden crisper Menge der zulässigen Lösungen zu maximieren ist.

4 Auf σ -Niveaus basierende Kompromiss-Zielfunktion(en)

D. Carlsson; Korhonen [3] substituieren die zu maximierende Fuzzy-Zielfunktion $\tilde{Z}(\mathbf{x}) = (\underline{c}(\mathbf{x}); \bar{c}(\mathbf{x}); \underline{\gamma}(\mathbf{x}); \bar{\gamma}(\mathbf{x}))_{LR}$ durch die crsipe lineare Zielfunktion

$$Z(\mathbf{x}) = (\bar{c}_1 + R_{C_1}^{-1}(\sigma)\bar{\gamma}_1) \cdot x_1 + \dots + (\bar{c}_n + R_{C_n}^{-1}(\sigma)\bar{\gamma}_n) \cdot x_n \rightarrow \text{Max} \quad (11)$$

wobei $\sigma \in [0, 1]$ ein vom Entscheider festzulegendes Zugehörigkeitsniveau ist. $R_{C_j}^{-1}$ ist die Umkehrfunktion der rechten Referenzfunktion des Fuzzy-Intervalls \tilde{C}_j .

Nach [3] sind auch die Fuzzy-Restriktionen des FLP-Systems (1) durch eine crsipe lineare Restriktion zu ersetzen, die auf dem gleichen σ -Niveau basiert.

Damit kann das System (1) ersetzt werden durch das crsipe LP-Programm

$$Z(\mathbf{x}) = (\bar{c}_1 + R_{C_1}^{-1}(\sigma)\bar{\gamma}_1) \cdot x_1 + \dots + (\bar{c}_n + R_{C_n}^{-1}(\sigma)\bar{\gamma}_n) \cdot x_n \rightarrow \text{Max}$$

unter Beachtung der Restriktionen (12)

$$\sum_{j=1}^n (\bar{a}_i(\mathbf{x}) + \underline{\alpha}_i(\mathbf{x}) \cdot R_{A_i}^{-1}(\sigma)) \leq b_i + \beta_i \cdot R_{B_i}^{-1}(\sigma), \quad i = 1, 2, \dots, m; \quad \mathbf{x} \in \mathbb{R}_0^n$$

Da ein Entscheider mit der Festlegung eines σ -Niveau überfordert ist, schlagen Carlsson; Korhonen [3] die Berechnung einer Lösungsmenge $\{(z^*(\sigma), x^*(\sigma)) | \sigma\}$ für verschiedene Parameter $\sigma \in [0, 1]$ vor. Diese sollte dann dem Entscheidungsträger präsentiert werden und dieser sollte dann, eventuell. unter Hinzuziehung weiterer Gesichtspunkte, seine Auswahl treffen.

E. Ansatz von Buckley [1; 2] ist dem Vorschlag von Carlsson; Korhonen [3] sehr ähnlich. Der Hauptunterschied liegt in einer optimistischeren Interpretation der Fuzzy-Restriktionen, für die Buckley die crisper Ersatzungleichungen

$$\sum_{j=1}^n (\underline{a}_i(\mathbf{x}) - \underline{\alpha}_i(\mathbf{x}) \cdot L_{A_i}^{-1}(\sigma)) \leq b_i + \beta_i \cdot R_{B_i}^{-1}(\sigma), \quad i = 1, 2, \dots, m; \quad \mathbf{x} \in \mathbb{R}_0^n \quad (13)$$

Hierbei ist $L_{A_i}^{-1}$ die Umkehrfunktion zur linksseitigen Referenzfunktionen des Fuzzy-Intervalls \tilde{A}_j und $R_{B_i}^{-1}$ die Umkehrfunktion zur rechtsseitigen Referenzfunktionen der Fuzzy-Zahl \tilde{B}_i . Diese Interpretation einer Kleiner-Gleich-Relation in Fuzzy-Restriktionen entspricht dem von Slowinski [16] eingeführten „optimistischen Index“, vgl. (32a).

F. Ramik [8] stellt einen optimistischen und einen pessimistischen Lösungsansatz vor. Ramiks optimistischer Ansatz entspricht genau dem Lösungsvorschlag von Buckley. Der pessimistische Ansatz von Ramik ist ähnlich strukturiert:

$$Z(\mathbf{x}) = (\bar{c}_1 - L_{C_1}^{-1}(\sigma)\bar{\gamma}_1) \cdot x_1 + \dots + (\bar{c}_n - L_{C_n}^{-1}(\sigma)\bar{\gamma}_n) \cdot x_n \rightarrow \text{Max}$$

unter Beachtung der Restriktionen (14)

$$\sum_{j=1}^n (\bar{a}_j(\mathbf{x}) + \bar{\alpha}_j(\mathbf{x}) \cdot R_{A_j}^{-1}(1-\sigma)) \leq b_i - \underline{\beta}_i \cdot R_{B_i}^{-1}(1-\sigma), \quad i = 1, 2, \dots, m; \quad \mathbf{x} \in \mathbf{R}_0^n$$

Für die Zielfunktion wird nun der linke Radpunkt des σ -Schnitts benutzt und für die Restriktionen die pessimistische ρ -Präferenz mit $\rho = 1-\sigma$. Dies entspricht damit dem der Aussage “fast positive auf dem Niveau $h = \sigma$ ” in [18]. Meiner Ansicht nach wäre es aber sinnvoller, auch in den Restriktionen des Systems (14) das Niveau σ anstelle von $1-\sigma$ zu verwenden, denn dies ließe sich so interpretieren, dass beim Lösungsansatz auf Werte mit einem Zugehörigkeitsniveau kleiner als σ verzichtet wird.

G. Beim Lösungsvorschlag von Delgado; Verdegay; Vila [5] werden die beiden Randpunkte der σ -Schnitte $C_j^\sigma = [(\underline{c}_j - \underline{\gamma}_j^\sigma), (\bar{c}_j + \bar{\gamma}_j^\sigma)]$ zur Konstruktion neuer crisper Zielfunktionen verwendet. Dabei sind alle möglichen Kombinationen zugelassen. Ein Beispiel könnte wie folgt aussehen:

$$Z(\mathbf{x}) = (\bar{c}_1 + \bar{\gamma}_1^\sigma) \cdot x_1 + (\underline{c}_2 - \underline{\gamma}_2^\sigma) \cdot x_2 + (\underline{c}_3 - \underline{\gamma}_3^\sigma) \cdot x_3 + \dots + (\bar{c}_n + \bar{\gamma}_n^\sigma) \cdot x_n \rightarrow \text{Max} \quad (15)$$

Die Kompromisszielfunktion ist dann nach [5] eine gewichtete Summe aller möglichen so gebildeten crisper Zielfunktionen, besteht also maximal aus 2^n Funktionen des Typs (15).

H. Chanas; Kuchta [4] betrachten ein LP-System mit einer Fuzzy-Zielfunktion und einer crisper Menge zulässiger Lösungen X .

Um eine Fuzzy-Lösung $\tilde{X} = \{(\mathbf{x}, \mu_X(\mathbf{x}) \mid \mathbf{x} \in \mathbf{R}_{0+}^n\}$ dieses Problems zu berechnen, vereinfachen die Autoren die Aufgabenstellung, in dem sie das ursprüngliche LP-System ersetzen durch ein System mit Intervallkoeffizienten $C_j^\sigma = [(\underline{c}_j - \underline{\gamma}_j^\sigma), (\bar{c}_j + \bar{\gamma}_j^\sigma)]$, welche σ -Schnitte der \tilde{C}_j für einen speziellen Zugehörigkeitswert $\sigma \in [0, 1]$ sind:

$$\text{Max}_{\mathbf{x} \in X} (C_1^\sigma x_1 + C_2^\sigma x_2 + \dots + C_n^\sigma x_n) \quad (16)$$

Um eine Lösung des Systems (16) zu erhalten verwenden Chanas; Kuchta [4] das Konzept einer t_0 - t_1 -optimalen Lösung, $0 \leq t_0 < t_1 \leq 1$. Gemäß dieses Verfahrens ist (16) in das zweiwertige LP-Problem

$$\text{Max}_{\mathbf{x} \in X} \left(\begin{array}{l} f_1(\mathbf{x}) = \sum_{j=1}^n \left(\underline{c}_j + t_0 \cdot (\bar{c}_j - \underline{c}_j) + (t_0(\underline{\gamma}_j^\sigma + \bar{\gamma}_j^\sigma) - \underline{\gamma}_j^\sigma) \cdot \theta \right) \cdot x_j \\ f_2(\mathbf{x}) = \sum_{j=1}^n \left(\underline{c}_j + t_1 \cdot (\bar{c}_j - \underline{c}_j) + (t_1(\underline{\gamma}_j^\sigma + \bar{\gamma}_j^\sigma) - \underline{\gamma}_j^\sigma) \cdot \theta \right) \cdot x_j \end{array} \right) \quad (17)$$

umzuformen, wobei $\theta = L^{-1}(\sigma)$.

Mittels eines parametrischen Simplex Algorithmus ist dann eine Basislösung \mathbf{x}^0 von (17) zu berechnen, die effizient auf der Menge

$$\langle \theta_0, \theta_1 \rangle \cup \langle \theta_1, \theta_2 \rangle \cup \dots \cup \langle \theta_{s-1}, \theta_s \rangle \quad (18)$$

ist, wobei die $\langle \theta_{s-1}, \theta_s \rangle$ Teilintervalle von $[L^{-1}(1), L^{-1}(0)]$ sind, die offen oder geschlossen sein können.

Der Zugehörigkeitswert der Basislösung \mathbf{x}^0 ist dann nach [4] gleich

$$\mu_X(\mathbf{x}^0) = \sum_{s=1}^S (L(\theta_{s-1}) - L(\theta_s)) \quad (19)$$

I. Zur Lösung eines Fuzzy Multikriteriellen LP-systems von der allgemeinen Form

$$\begin{array}{l|l} \tilde{C}_{11}x_1 \oplus \tilde{C}_{12}x_2 \oplus \dots \oplus \tilde{C}_{1n}x_n & \\ \vdots & \\ \tilde{C}_{K1}x_1 \oplus \tilde{C}_{K2}x_2 \oplus \dots \oplus \tilde{C}_{Kn}x_n & \rightarrow \text{M\ddot{a}x} \end{array} \quad \text{unter Beachtung der Nebenbedingungen} \quad (20)$$

$$\tilde{A}_{i1}x_1 \oplus \tilde{A}_{i2}x_2 \oplus \dots \oplus \tilde{A}_{in}x_n \lesssim \tilde{B}_i, \quad i = 1, \dots, m, \quad \mathbf{x} \in \mathbb{R}_0^n$$

schlagen Sakawa, Yano [14, 15] einen interaktiven Lösungsprozess vor, bei dem schrittweise eine so genannte "G- σ -pareto-optimale Lösung" errechnet wird. Dabei werden auch hier die Fuzzy-Koeffizienten \tilde{C}_{kj} und \tilde{A}_{ij} reduziert auf ihre σ -Schnitte. Das Zugehörigkeitsniveau σ ist dabei durch den Entscheidungsträger (ET) festzulegen, kann aber im Verlauf des Lösungsprozesses geändert werden.

Bei diesem - unnötig kompliziert formulierten- Lösungsverfahren wird jede Fuzzy-Restriktion durch eine crisper Bedingungsungleichung ersetzt, wobei Sakawa; Yano den optimistischen Ansatz (13) verwenden, der auch von Buckley [1, 2], Ramik [8] und Slowinski [16] vorgeschlagen wird. Die Menge, die diesen Bedingungen genügt, sei mit X^σ bezeichnet.

Die Fuzzy-Zielfunktionen substituieren Sakawa; Yano durch monoton steigende Zugehörigkeitsfunktionen $\mu_{Z_k}(\mathbf{c}_k' \cdot \mathbf{x})$ für alle $k = 1, \dots, K$, die im Sinne der Arbeit von Zimmermann [20] als Nutzenfunktionen interpretiert werden können. Zur Bestimmung einer Kompromisslösung benutzen die Autoren aber nicht die direkte Gesamtnutzenmaximierung sondern einen Regretansatz. Dazu muss der ET für jedes Ziel einen Referenzzugehörigkeitswert $\bar{\mu}_{Z_k}$ festlegen.

Zu minimieren ist dann die Gesamtregretfunktion

$$Z = \text{Max}([\bar{\mu}_{Z_1} - \mu_{Z_1}(\bar{\mathbf{c}}_1^\sigma' \cdot \mathbf{x})], \dots, [\bar{\mu}_{Z_K} - \mu_{Z_K}(\bar{\mathbf{c}}_K^\sigma' \cdot \mathbf{x})]) \quad (21)$$

über der Menge der zulässigen Lösungen X^σ . Dabei ist $\bar{\mathbf{c}}_k^\sigma = (\bar{c}_{k1} + \bar{\gamma}_{k1}^\sigma, \dots, \bar{c}_{kn} + \bar{\gamma}_{kn}^\sigma)$.

J. Luhandjula [7] interpretiert die Zugehörigkeitswerte als Möglichkeitswerte (possibilities). Für eine gegebene Menge $X \subset \mathbb{R}_+^n$ an zulässigen Lösungen bezeichnet er einen Vektor $\mathbf{x}^0 \in X$ als "β-possible efficient"-Lösung des Fuzzy-Optimierungssystems

$$\text{Max}_{\mathbf{x} \in X} (\tilde{Z}_1(\mathbf{x}), \dots, \tilde{Z}_K(\mathbf{x})),$$

wenn kein anderer Vektor $\mathbf{x} \in X$ existiert mit der Eigenschaft

$$\text{Poss} \left(\begin{array}{l} \tilde{Z}_1(\mathbf{x}) \geq \tilde{Z}_1(\mathbf{x}^0), \dots, \tilde{Z}_{k-1}(\mathbf{x}) \geq \tilde{Z}_{k-1}(\mathbf{x}^0), \\ \tilde{Z}_k(\mathbf{x}) > \tilde{Z}_k(\mathbf{x}^0), \tilde{Z}_{k+1}(\mathbf{x}) \geq \tilde{Z}_{k+1}(\mathbf{x}^0), \dots, \tilde{Z}_K(\mathbf{x}) \geq \tilde{Z}_K(\mathbf{x}^0) \end{array} \right) \geq \beta \quad (22)$$

Gemäß dieser Definition ist ein Vektor \mathbf{x}^0 genau dann “ β -possible” auf X , wenn \mathbf{x}^0 eine effiziente Lösung des Optimierungssystems mit Intervallkoeffizienten

$$\text{Max}_{\mathbf{x} \in X} (\mathbf{c}_1^{\beta'}, \dots, \mathbf{c}_K^{\beta'}) \quad (23)$$

Dabei sind die Größen $\mathbf{c}_k^{\beta'} = ([\underline{c}_{k1}^{\beta} - \underline{\gamma}_{k1}^{\beta}, \bar{c}_{k1}^{\beta} + \bar{\gamma}_{k1}^{\beta}], \dots, [\underline{c}_{kn}^{\beta} - \underline{\gamma}_{kn}^{\beta}, \bar{c}_{kn}^{\beta} + \bar{\gamma}_{kn}^{\beta}])$ Vektoren, die aus den β -Schnitten der Koeffizienten \tilde{C}_{kj} , $j = 1, 2, \dots, n$; $k = 1, 2, \dots, K$, zusammengesetzt sind.

Zur Berechnung einer effizienten Lösung von (23) verwendet Luhandjula ein Lösungsverfahren, das auf den Intervallgrenzen $\underline{c}_{kj}^{\beta} - \underline{\gamma}_{kj}^{\beta}$ und $\bar{c}_{kj}^{\beta} + \bar{\gamma}_{kj}^{\beta}$ der β -Schnitte der Koeffizienten \tilde{C}_{kj} basiert.

K. Während in den vorstehenden Lösungsvorschlägen jeweils nur mit einem σ -Schnitt gearbeitet wird, schlagen Rommelfanger, Hanuscheck, Wolf [11] in dem Lösungsverfahren „ α -Niveau bezogenen Paarbildung“ die Verwendung mehrere Schnitte vor. Dies geschieht im Sinne des Darstellungssatzes der Fuzzy-Mengenlehre, dass eine Fuzzy-Menge eindeutig durch die Gesamtmenge α -Niveau-Mengen beschrieben werden kann. Für die praktische Anwendung sind dann zur Beschreibung der Koeffizienten \tilde{C}_j wenige Schnitte ausreichend. Speziell lassen sich Fuzzy-Intervalle \tilde{C}_j hinreichend genau beschreiben durch eine Menge von Intervallen

$$C_j^{\sigma} = [\underline{c}_j - \underline{\gamma}_j^{\sigma}; \bar{c}_j + \bar{\gamma}_j^{\sigma}] \quad \text{for } \sigma = \sigma_1, \dots, \sigma_S \in [0, 1], S \in \mathbf{N} \quad (24)$$

Die Autoren ersetzen dann die Fuzzy-Zielfunktionen $Z^{\sigma_s}(\mathbf{x}) = C_1^{\sigma_s} x_1 \oplus \dots \oplus C_n^{\sigma_s} x_n$ mit Intervallkoeffizienten jeweils durch zwei crisper Zielfunktionen, wobei die Intervallränder verwendet werden.

$$\left. \begin{array}{l} Z_{\text{Min}}^{\sigma_1}(\mathbf{x}) = (\underline{c}_1 - \underline{\gamma}_1^{\sigma_1})x_1 + \dots + (\underline{c}_n - \underline{\gamma}_n^{\sigma_1})x_n \\ Z_{\text{Max}}^{\sigma_1}(\mathbf{x}) = (\bar{c}_1 + \bar{\gamma}_1^{\sigma_1})x_1 + \dots + (\bar{c}_n + \bar{\gamma}_n^{\sigma_1})x_n \\ \vdots \\ Z_{\text{Min}}^{\sigma_S}(\mathbf{x}) = (\underline{c}_1 - \underline{\gamma}_1^{\sigma_S})x_1 + \dots + (\underline{c}_n - \underline{\gamma}_n^{\sigma_S})x_n \\ Z_{\text{Max}}^{\sigma_S}(\mathbf{x}) = (\bar{c}_1 + \bar{\gamma}_1^{\sigma_S})x_1 + \dots + (\bar{c}_n + \bar{\gamma}_n^{\sigma_S})x_n \end{array} \right\} \rightarrow \text{Max} . \quad (25)$$

Um eine Kompromisslösung dieses Mehrzieloptimierungssystems auf einer crisper Menge X zulässiger Lösungen zu berechnen, schlagen die Autoren vor, in einem ersten Schritt die extremen Zielfunktionen in (25) jeweils paarweise durch Fuzzy-Zielfunktionen der nachfolgenden Form zu ersetzen:

$$\mu_k^\alpha(\mathbf{x}) = \begin{cases} 1 & \text{if } z_k^{\alpha*} < Z_k^\alpha(\mathbf{x}) \\ \frac{Z_k^\alpha(\mathbf{x}) - z_k^{\alpha L}}{z_k^{\alpha*} - z_k^{\alpha L}} & \text{if } z_k^{\alpha L} < Z_k^\alpha(\mathbf{x}) < z_k^{\alpha*} \\ 0 & \text{if } Z_k^\alpha(\mathbf{x}) < z_k^{\alpha L} \end{cases} \quad (26)$$

Dabei steht der Index k für Min oder Max und es gilt

$$\begin{aligned} z_{\text{Min}}^{\alpha*} &= Z_{\text{Min}}^\alpha(\mathbf{x}_{\text{Min}}^{\alpha*}) = \text{Max}_{\mathbf{x} \in X} Z_{\text{Min}}^\alpha(\mathbf{x}) \\ z_{\text{Max}}^{\alpha*} &= Z_{\text{Max}}^\alpha(\mathbf{x}_{\text{Max}}^{\alpha*}) = \text{Max}_{\mathbf{x} \in X} Z_{\text{Max}}^\alpha(\mathbf{x}) \\ z_{\text{Min}}^{\alpha L} &= Z_{\text{Min}}^\alpha(\mathbf{x}_{\text{Max}}^{\alpha*}) \quad \text{und} \quad z_{\text{Max}}^{\alpha L} = Z_{\text{Max}}^\alpha(\mathbf{x}_{\text{Min}}^{\alpha*}). \end{aligned}$$

Die Kompromisslösung ist dann zu bestimmen als Lösung des crisen Optimierungssystems

$$\text{Max}_{\mathbf{x} \in X} \text{Min}(\mu_{\text{Min}}^{\sigma_1}(\mathbf{x}), \mu_{\text{Max}}^{\sigma_1}(\mathbf{x}), \dots, \mu_{\text{Min}}^{\sigma_S}(\mathbf{x}), \mu_{\text{Max}}^{\sigma_S}(\mathbf{x})), \quad (27)$$

das äquivalent ist zu dem crisen LP-System:

$$\lambda \rightarrow \text{Max}$$

unter Beachtung der Nebenbedingungen (28)

$$\mathbf{x} \in X, \quad \mu_{\text{Min}}^{\sigma_s}(\mathbf{x}) \geq \lambda, \quad \mu_{\text{Max}}^{\sigma_s}(\mathbf{x}) \geq \lambda, \quad s = 1, \dots, S.$$

5 Satisfizierende Lösungen

Da normalerweise bei einem Mehrzieloptimierungssystem eine ideale Lösung nicht existiert, empfehlen Slowinski [16] und Rommelfanger [9, 10, 12] die Berechnung einer satisfizierenden Lösung. Dies entspricht auch besser dem in der Praxis üblichen Vorgehen, dass vorgegebene Anspruchsniveaus (z. B. Zielvereinbahrungen) erfüllt werden sollen.

Analog zur Modellierung einer unscharfen rechten Seite \tilde{B}_i ist dann für jedes Ziel k ein Anspruchsniveau $\tilde{N}_k = (n_k; \underline{v}_k; 0)_{RR}$ festzulegen. Die Satisfizierungsbedingungen treten dann als zusätzliche Restriktionen auf:

$$\tilde{N}_k \tilde{\leq} \tilde{Z}_k(\mathbf{x}) \quad (29)$$

L. Um eine möglichst hohe Übereinstimmung zwischen $\tilde{N}_k = (n_k; \underline{v}_k; 0)_{RR}$ und $\tilde{Z}_k(\mathbf{x}) = (\underline{c}_k(\mathbf{x}); \bar{c}(\mathbf{x}); \underline{\gamma}_k(\mathbf{x}); \bar{\gamma}_k(\mathbf{x}))_{LR}$ zu erreichen, schlägt Slowinski [16] die Maximierung der Ordinaten $F_k(\mathbf{x})$ des Schnittpunktes dem rechten Kurvenstück von $\tilde{Z}_k(\mathbf{x})$ mit dem linken Kurvenstück von \tilde{N}_k vor, d. h.

$$F_k(\mathbf{x}) = R\left(\frac{\bar{c}_k(\mathbf{x}) - n_k}{\bar{\gamma}_k(\mathbf{x}) - \underline{v}_k}\right) \rightarrow \text{Max}. \quad (30)$$

Für den in der Praxis zumeist anzutreffenden Fall, dass die Referenzfunktion R linear oder stückweise linear ist, nimmt $F_k(\mathbf{x})$ die linear gebrochene Gestalt an:

$$F_k(\mathbf{x}) = 1 - \frac{\bar{c}_k(\mathbf{x}) - n_k}{\bar{\gamma}_k(\mathbf{x}) - \underline{v}_k}. \quad (31)$$

Nach Slowinski [16] werden die Fuzzy-Restriktionsungleichungen des Systems (20) ersetzt durch zwei crise Ungleichungen

$$\underline{a}_i(\mathbf{x}) - b_i \leq (\underline{\alpha}_i(\mathbf{x}) + \beta_i) \cdot L^{-1}(\tau), \quad \tau \in]0, 1] \quad \text{optimistischer Index} \quad (32a)$$

$$\bar{a}_i(\mathbf{x}) + \bar{\alpha}_i(\mathbf{x}) \cdot R^{-1}(\eta) \leq b_i + \beta_i \cdot L^{-1}(\eta), \quad \eta \in [0, 1] \quad \text{pessimistischer Index} \quad (32b)$$

M. Rommelfanger [9, 10, 12] empfiehlt für die Umformung der Restriktionen die Verwendung der Ungleichheitsrelation " $\tilde{\leq}_R$ ":

$$\tilde{A}_i(\mathbf{x}) \tilde{\leq}_R \tilde{B}_i \Leftrightarrow \begin{cases} \bar{a}_i(\mathbf{x}) + \bar{\alpha}_i(\mathbf{x}) \cdot R^{-1}(\varepsilon) \leq b_i + \beta_i \cdot R^{-1}(\varepsilon) \\ \mu_i(\bar{a}_i(\mathbf{x})) \rightarrow \text{Max.} \end{cases} \quad (33)$$

Hierbei wird eine Fuzzy-Restriktion ersetzt durch eine crise Restriktion, die dem pessimistischen Index von Slowinski entspricht, und einer Fuzzy-Zielfunktion

$$\mu_i(\bar{a}_i(\mathbf{x})) = \begin{cases} 1 & \text{if } \bar{a}_i(\mathbf{x}) < b_i \\ R\left(\frac{b_i - \bar{a}_i(\mathbf{x})}{\beta_i}\right) & \text{if } b_i \leq \bar{a}_i(\mathbf{x}) \leq b_i + \beta_i, \\ 0 & \text{if } b_i + \beta_i < \bar{a}_i(\mathbf{x}) \end{cases} \quad (34)$$

die interpretiert werden kann als subjektive Bewertung der Größe $\bar{a}_i(\mathbf{x}) = \sum_{j=1}^n \bar{a}_{ij} x_j$ in

Bezug auf die rechte Seite b_i .

Für die Zielbedingungen wird die analoge Ungleichrelation " $\tilde{\geq}_R$ " vorgeschlagen:

$$\tilde{Z}_k(\mathbf{x}) \tilde{\geq}_R \tilde{N}_k \Leftrightarrow \begin{cases} \underline{c}_k(\mathbf{x}) + \underline{\gamma}_k(\mathbf{x}) \cdot L^{-1}(\varepsilon) \geq n_k + \underline{v}_k \cdot L^{-1}(\varepsilon) \\ \mu_{Z_k}(\underline{c}_k(\mathbf{x})) \rightarrow \text{Max} \end{cases} \quad (35a)$$

$$(35b)$$

mit

$$\mu_{Z_k}(\underline{c}_k(\mathbf{x})) = \begin{cases} 0 & \text{if } \underline{c}_k(\mathbf{x}) < n_k - \underline{v}_k \\ L\left(\frac{n_k - \underline{c}_k(\mathbf{x})}{\underline{v}_k}\right) & \text{if } n_k - \underline{v}_k \leq \underline{c}_k(\mathbf{x}) \leq n_k \\ 1 & \text{if } n_k < \underline{c}_k(\mathbf{x}) \end{cases} \quad (36)$$

Somit wird das Mehrzieloptimierungssystem (20) ersetzt durch das Optimierungssystem

$$\text{Max}_{\mathbf{x} \in R^n} (\mu_{Z_1}(\underline{c}_1(\mathbf{x})), \dots, \mu_{Z_K}(\underline{c}_K(\mathbf{x})), \mu_1(\bar{a}_1(\mathbf{x})), \dots, \mu_m(\bar{a}_m(\mathbf{x})))$$

unter Beachtung der Restriktionen (37)

$$\underline{c}_k(\mathbf{x}) + \underline{\gamma}_k(\mathbf{x}) L^{-1}(\varepsilon) \geq n_k + \underline{v}_k L^{-1}(\varepsilon), \quad k = 1, \dots, K$$

$$\bar{a}_i(\mathbf{x}) + \bar{\alpha}_i(\mathbf{x}) \cdot R^{-1}(\varepsilon) \leq b_i + \beta_i \cdot R^{-1}(\varepsilon), \quad i = 1, \dots, m$$

$$\mathbf{x} \geq \mathbf{0},$$

Zur Ermittlung einer Kompromisslösung schlägt der Autor vor, die Kompromissziel-funktion

$$\lambda(\mathbf{x}) = \text{Min} (\mu_{Z_1}(\underline{c}_1(\mathbf{x})), \dots, \mu_{Z_K}(\underline{c}_K(\mathbf{x})), \mu_1(\bar{a}_1(\mathbf{x})), \dots, \mu_m(\bar{a}_m(\mathbf{x}))) \quad (38)$$

zu maximieren, die interpretiert werden kann als Ausdruck für die Gesamtzufriedenheit des Entscheidungsträgers mit der gefundenen Lösung \mathbf{x} .

Sind alle Zugehörigkeitsfunktionen μ_{Z_k} and μ_i stückweise linear und konkav, dann stellt das aus der Zielfunktion (38) und den Restriktionen von (37) gebildete System

$$\begin{aligned} \lambda &\rightarrow \text{Max} \\ \text{unter Beachtung der Nebenbedingungen} & \quad (39) \\ \lambda &\leq \mu_{Z_k}(\underline{c}_k(\mathbf{x})), \quad k = 1, \dots, K \\ \lambda &\leq \mu_i(\bar{a}_i(\mathbf{x})), \quad i = 1, \dots, m \\ \underline{c}_k(\mathbf{x}) + \underline{\gamma}_k(\mathbf{x})L^{-1}(\varepsilon) &\geq n_k + \underline{v}_kL^{-1}(\varepsilon), \quad k = 1, \dots, K \\ \bar{a}_i(\mathbf{x}) + \bar{\alpha}_i(\mathbf{x}) \cdot R^{-1}(\varepsilon) &\leq b_i + \beta_i \cdot R^{-1}(\varepsilon), \quad i = 1, \dots, m \\ \mathbf{x} &\geq \mathbf{0}, \end{aligned}$$

Nach Rommelfanger ist die Kompromisslösung mittels eines interaktiven Prozesses zu ermitteln, der über Anspruchsniveaueinstellung gesteuert wird. Dazu werden die durch stückweise lineare Referenzfunktionen beschriebenen Anspruchsniveaus \tilde{N}_k und die rechten Seiten \tilde{B}_i durch Vorgabe reeller Anspruchsniveaueinstellungswerte $n_k^{\lambda_A}$ bzw. $b_i^{\lambda_A}$, die für eine akzeptable Lösung nicht unter bzw. überschritten werden dürfen, genauer beschrieben. Zum Vergleich der Anspruchsniveaus werden diesen crispen Anspruchsniveaus das gleiche Zugehörigkeitsniveau λ_A , $0 < \lambda_A \leq 1$ zugeordnet.

6 Fazit

Der Überblick belegt, dass es eine Vielzahl an praktikablen Verfahren zur Ermittlung einer optimalen Lösung für Lineare (Mehrziel-) Optimierungssysteme mit Fuzzy-Koeffizienten gibt. Um die richtige Auswahl zu treffen, sind die für die einzelnen Methoden getroffenen Annahmen mit dem real vorliegenden Problem zu vergleichen. Grundsätzlich gilt die Empfehlung, die Lösung schrittweise in einem interaktiven Prozess zu ermitteln, bei dem die Zwischenergebnisse und/oder Informationen von außen zusätzlich berücksichtigt werden können. Auch wenn im Verlauf der Lösungsverfahren vereinfachende Annahmen getroffen werden, so bieten Fuzzy-LP-Systeme die Chance, das Realproblem adäquat abzubilden und eine zufrieden stellende Kompromisslösung zu ermitteln. Nicht unterschätzt werden sollte der Vorteil, dass durch die Fuzzy-Modellierung Informationskosten eingespart werden können, da man zunächst mit ungenauen Größen arbeiten kann, die dann nach Bedarf schrittweise verbessert werden.

7 Literatur

1. Buckley, J.J.: Solving possibilistic linear programming problems. *Fuzzy Sets and Systems* 31 (1989) 329-341
2. Buckley, J.J.: Joint solution to fuzzy programming problems. *Fuzzy Sets and Systems* 72 (1995) 215-220
3. Carlsson, C.; Korhonen, P.: A parametric approach to fuzzy linear programming. *Fuzzy Sets and Systems* 20 (1986) 17-30
4. Chanas, S.; Kuchta, D.: Linear programming problems with fuzzy coefficients in the objective function. In: Delgado M.; Kacprzyk J.; Verdegay J.-L.; Vila M.A., *Fuzzy Optimization*. Physica-Verlag, Heidelberg (1994) 148-157

5. Delgado, M., Verdegay, J.L.; Vila, M.A.: A general model for fuzzy linear programming., *Fuzzy Sets and Systems* 29 (1989) 21-30
5. Iniguchi, M.; Ramik, J.; Tanino, T.; Vlach, M.: Satisficing solutions and duality in interval and fuzzy linear programming. *Fuzzy Sets and Systems* 135 (2003) 151-177
6. Lai, Y.-J.; Hwang, C.-L.: Interactive fuzzy linear programming. *Fuzzy Sets and Systems* 45 (1992) 169-183
7. Luhandjula, M.K.: Multiple objective programming with possibilistic coefficients. *Fuzzy Sets and Systems* 21 (1987) 135-146
8. Ramik, J.: Duality in fuzzy linear programming with possibilistic and necessity relations. *Fuzzy Sets and Systems* 157 (2006) 1283-1302
9. Rommelfanger H.: *Fuzzy decision support systems - Decisions in fuzzy environment* (in German). Springer Verlag, Berlin, Heidelberg, second edition (1994)
10. Rommelfanger, H.: FULPAL 2.0 - An interactive algorithm for solving multicriteria fuzzy linear programs controlled by aspiration levels. In: Schweigert D. (Ed.), *Methods of multicriteria decision theory*. Pfalz Akademie, Lamprecht (1995) 21-34
11. Rommelfanger, H., Hanuscheck, R.; Wolf, J.: Linear programming with fuzzy objectives, *Fuzzy Sets and Systems* 29 (1995) 31-48
12. Rommelfanger, H.; Keresztfalvi, T.: Multicriteria fuzzy optimization based on Yager's parametrized t-norm. *Foundations of Computing and Decision Sciences* 16 (1995) 99-110
13. Rommelfanger, H.; Slowinski, R.: *Fuzzy Linear Programming with single or multiple Objective Functions*. In: Slowinski, R. (Ed.) *Fuzzy Sets in Decision Analysis, Operations Research and Statistics*. Kluwer Academic Publishers Norwell, Massachusetts, USA 1998 (1999) 179-213
14. Sakawa, M.; Yano, H.: Interactive fuzzy satisficing method for multiobjective nonlinear programming problems with fuzzy parameters. *Fuzzy Sets and Systems* 30 (1989) 221-238.
15. Sakawa, M.; Yano, H.: Interactive decision making for multiobjective programming problems with fuzzy parameters. In: Slowinski R. and Teghem J. (Eds.), *Stochastic versus fuzzy approaches to multiobjective mathematical programming under uncertainty*. Kluwer Academic Publishers, Dordrecht (1990) 191-228
16. Slowinski, R.: 'FLIP': an interactive method for multiobjective linear programming with fuzzy coefficients. In: Slowinski R. and Teghem J. (Eds.), *Stochastic versus fuzzy approaches to multi-objective mathematical programming under uncertainty*. Kluwer Academic Publishers, Dordrecht (1990) 249-262
17. Stanculescu, Ch.: *Multiobjective decision support tools using concepts of fuzzy sets*. Université catholique de Louvain, Louvain-La-Neuve, Belgium (2001)
18. Tanaka, H.; Ichihashi, H.; Asai, K.: A formulation of linear programming problems based on comparison of fuzzy numbers. *Control and Cybernetics* 13 (1984) 185-194.
19. Zadeh, L.A.: Fuzzy sets. *Information and Control* 8 (1965) 338-353
20. Zimmermann, H.-J.: Fuzzy programming and linear programming with several objective functions. *Fuzzy Sets and Systems* 1 (1978) 45-55

Formalisierung menschlicher Interaktionen durch Situations-Operator-Modellbildung

Dennis Gamrad und Dirk Söffker

Lehrstuhl Steuerung, Regelung und Systemdynamik, Universität Duisburg-Essen
Lotharstr. 1, 47057 Duisburg, Deutschland
Tel. (0203) 379-3423
Fax (0203) 379-3027
E-Mail: {dennis.gamrad, soeffker}@uni-due.de
<http://www.srs.uni-due.de>

1 Einleitung

Die Modellbildung menschlicher Interaktionen bildet den Schlüssel für eine Reihe von Problemstellungen vieler Anwendungsbereiche. Typische Beispiele hierfür sind Überwachungs- und Assistenzsysteme sowie autonom handelnde technische Systeme. Die Modellbildung kognitiver Funktionen und Prozesse erlaubt ein prinzipielles Verständnis menschlicher Interaktionslogik.

Lernmechanismen auf Basis einer geeigneten Repräsentation des Wissens (Struktur der Zusammenhänge sowie der Logik der Interaktion) spielen bei der Betrachtung interagierender Systeme eine zentrale Rolle. Durch die Formalisierung des Lernvorganges sowie die entsprechend laufende Optimierung des aufgenommenen Wissens, z. B. in einem mentalen Modell, können neue Zugänge zu klassischen Fragen der Überwachung und Assistenz von Mensch-Maschine-Systemen möglich werden.

In den folgenden Abschnitten wird die **Situations-Operator-Modellbildung (SOM)** vorgestellt, mit der sich menschliche bzw. kognitiv basierte Interaktionen formalisieren lassen. Danach folgt die Beschreibung aktueller Anwendungen, die aufgrund von SOM-basierten Modellen über kognitive Eigenschaften verfügen. Aufbauend darauf wird SOM-basiertes Lernen vorgestellt.

2 Situations-Operator-Modellbildung (SOM)

Durch die Situations-Operator-Modellbildung lassen sich komplexe Szenarien mit innerer Strukturvariabilität, unter der Annahme kausaler Abhängigkeit, informationstheoretisch beschreiben. Die Formalisierung von Abläufen in der realen Welt, wie sie von Menschen wahrgenommen wird, als Folge zeitlich fixierter Zustände, wurde erstmalig 1963 von McCarthy beschrieben [1]. Hierauf aufbauend werden durch Situations-Operator-Modellbildung [2] die Ausschnitte und Veränderungen der realen Welt als Folge von Szenen und Aktionen betrachtet. Eine Szene wird hierbei durch eine Situation und eine Aktion durch einen Operator modelliert.

Situationen bestehen aus einer Menge Merkmalen c und einer Menge Verknüpfungen r . Die Merkmale enthalten Parameter, welche die konkreten Werte oder allgemein Informationen der Merkmale sind. Durch die Verknüpfungen werden die Beziehungen unterschiedlicher Merkmale zueinander detailliert, wodurch die innere Struktur einer

Situation beschrieben wird. Eine Situation ohne Verknüpfungen wird als vorgefiltert und eine Situation mit Verknüpfungen wird als interpretiert bezeichnet. Auf technische Systeme übertragen, sind Merkmale physikalische Größen und Verknüpfungen beispielsweise mathematische Gleichungen oder neuronale Netze.

Durch die Anwendung eines Operators auf eine Anfangssituation wird diese in eine Endsituation überführt, indem die Merkmale und/oder die innere Struktur der Anfangssituation verändert werden. Ein Operator besteht aus einer Funktion F sowie expliziten und impliziten Voraussetzungen (eA , iA). Durch die Funktion F werden die Veränderungen, die ein Operator bewirkt, beschrieben und die expliziten und impliziten Voraussetzungen beschreiben die Bedingungen, die festlegen auf welche Anfangssituationen ein Operator angewendet werden kann.

Zur übersichtlicheren Darstellung von SOM-basierten Zusammenhängen steht eine spezielle Symbolik zur Verfügung. Diese grafische SOM-Notation ist in Bild 1 anhand eines einfachen Beispiels dargestellt. Die Anfangssituation, die aus zwei Verknüpfungen und drei Merkmalen besteht, wird durch die Anwendung des Operators in die Endsituation überführt. Hierdurch ändern sich die Merkmale sowie in diesem Fall die Verknüpfungen und demnach die innere Struktur. Sowohl Situationen (siehe Bild 1, Merkmal c_3) als auch Operatoren sind hierarchisierbar, was die Modellbildung unterschiedlicher Abstraktionsebenen erlaubt.

2.1 SOM-Muster höherer Petri Netze

Die Implementierung von SOM-basierten Modellen kann mittels Mustern höherer Petri Netze erfolgen. Hierfür stehen eine Reihe von Werkzeugen zur Verfügung, die eine rechnergestützte grafische Modellbildung, Simulation und Analyse ermöglichen.

In [3] wurden die beiden Programme CPN Tools für Coloured Petri Nets (vgl. [4]) und Renew für Referenznetze (vgl. [5]) untersucht, mit denen, abhängig von den konkreten Anforderungen an das Modell, verschiedene SOM-basierte Muster höherer Petri Netze erstellt werden können. Grob unterschieden werden Muster für rein diskrete und Muster für hybride (diskrete und kontinuierliche) SOM-basierte Modelle. Die Bezeichnungen diskret und hybrid beziehen sich hier auf die Situationen, die entweder ausschließlich aus zeitunabhängigen oder aus teilweise zeitvarianten Merkmalen bestehen. In Bild 2 sind beispielhaft zwei einfache Muster höherer Petri Netze dargestellt, die eine Situation bzw. einen Operator repräsentieren können. Auf der linken Seite ist das Muster einer

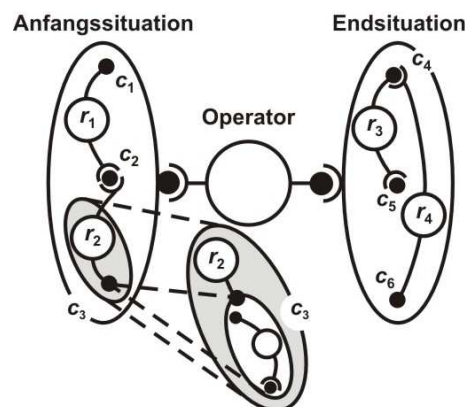


Bild 1: Eine Situation wird durch einen Operator verändert [2]

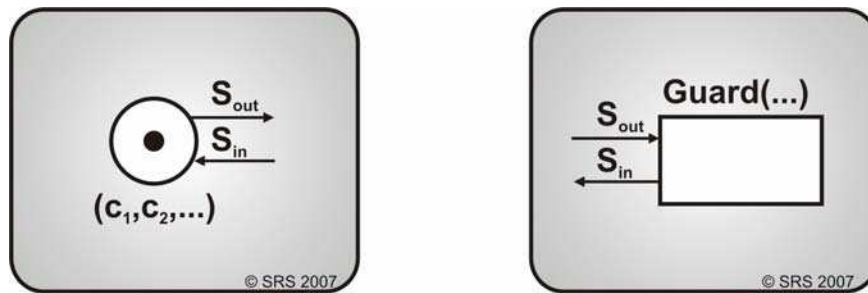


Bild 2: Beispiele für SOM-basierte Muster höherer Petri Netze

Situation dargestellt, die aus einem Platz mit einer zusammengesetzten Marke besteht. Das Muster für einen Operator ist auf der rechten Seite abgebildet, der aus einer Transition mit Kantengewichtungen und Guard-Funktion besteht. Ausgehend von diesen einfachen Mustern können komplexere Muster, durch strukturelle Änderung oder Verkettung, abgeleitet werden.

3 Anwendungen der Situations-Operator-Modellbildung

Am Lehrstuhl Steuerung, Regelung und Systemdynamik der Universität Duisburg-Essen werden innerhalb verschiedener Anwendungen kognitive technische Systeme realisiert. Im Folgenden werden auszugsweise zwei Beispielanwendungen vorgestellt, die die Situations-Operator-Modellbildung verwenden. Diese sind zum Einen der autonom lernende Roboter ALRob und zum Anderen das Projekt COSA, bei dem ein kognitives Fahrer-Assistenz- bzw. Überwachungssystem entwickelt wird. Die Zusammenfassung der vorausgehenden Arbeiten ist beispielsweise in [7] angegeben.

3.1 ALRob

Der ALRob (**A**utonomous **L**earning **R**obot) ist ein mobiler Roboter, der durch Interaktion mit seiner Umgebung ein mentales Modell aufbaut und verfeinert (vgl. Bild 3). Die Situations-Operator-Modellbildung wurde hierbei durch Strukturierung eines objektorientierten Hochsprachenprogrammes genutzt, wodurch kognitive Funktionen und Prozesse ausgeführt werden können. Als Wissensbasis wird eine objektorientierte Datenbank verwendet, wodurch die Datenstruktur, die zur Laufzeit verwendet wird, auch dauerhaft erhalten bleibt. Um die Datenbank aufzubauen bzw. zu verfeinern, wurden SOM-basierte Lernverfahren entworfen und implementiert, die in Abschnitt 4 näher beschrieben werden.

Die SOM-basierte Programmierung ermöglicht ALRob die Interaktion in unbekanntem Umgebungen. In einer kurzen Trainingsphase ist er in der Lage, die grundlegenden Interaktionsregeln (welcher Operator verändert wie welche Situation) zu 'erkennen'. Dies wird in erster Linie durch die innere Struktur der wahrgenommenen Situation ermöglicht, wodurch die Komplexität der sensorisch gemessenen Rohdaten interpretiert und somit reduziert wird. Neben den Fähigkeiten zur Komplexitätsreduzierung und Interaktion in unbekanntem Umgebungen verfügt ALRob über eine ausgeprägte Handlungsflexibilität, wodurch unerwartete Veränderungen der aktuellen Situation verarbeitet werden können. Ändert sich die Umgebung während ALRob eine Handlung ausführt oder einen Plan generiert, werden die detektierten Veränderungen umgehend

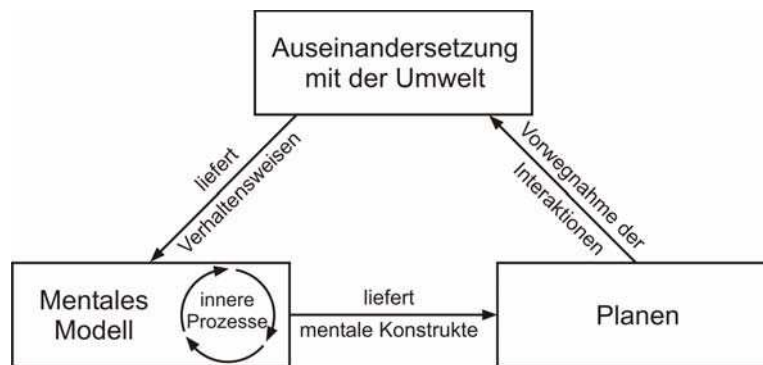


Bild 3: Verknüpfungen des mentalen Modells [2]

erkannt und zur Neuplanung verwendet. Die grundlegende Idee zur Steigerung der Handlungsflexibilität autonomer Systeme besteht hier also in der Kombination von Lernverfahren mit der Strukturierung der Interaktion mittels des SOM-Ansatzes.

3.2 COSA

Im Projekt COSA (**C**ognitive **S**upervision and **A**ssistance) wird das in [6] vorgestellte Konzept eines Fahrer-Überwachungsautomaten softwaretechnisch umgesetzt. Die softwaretechnische Umsetzung erfolgt hier, im Gegensatz zum ALRob, erstmalig durch Muster höherer Petri Netze, die mit Hilfe von entsprechenden Softwarewerkzeugen erstellt und ausgeführt werden können. Ziel des Projektes ist es, einen Automaten zu entwickeln, der die Interaktion zwischen einem Fahrer und seiner Umwelt überwacht und bei gefährlichen Situationen sowie bei erkannten Gefahrenmustern warnt. Damit die Akzeptanz für das Assistenzsystem erhalten bleibt, verwendet der COSA-Automat ein kognitives Fahrermodell, das während seiner Laufzeit fortlaufend auf einen bestimmten Fahrer individualisiert wird und sich an dessen spezifisches Verhalten anpasst.

Das Softwarewerkzeug, das den COSA-Automaten ausführt, wird innerhalb einer Simulationsumgebung eingesetzt. Der strukturelle Aufbau ist in Bild 4 schematisch

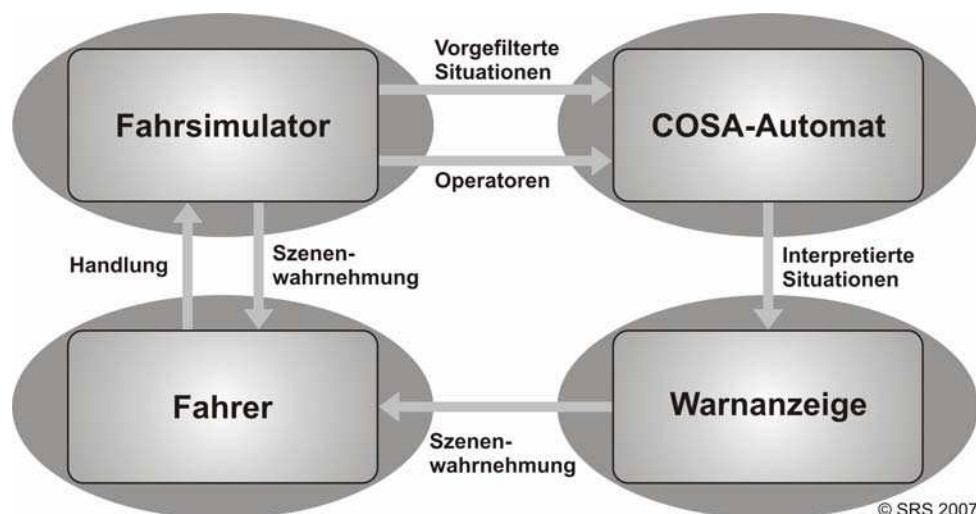


Bild 4: Struktureller Aufbau des COSA-Automaten

dargestellt. Ein Fahrer, der einen Fahrsimulator bedient, nimmt die reale Welt, die aus einer simulierten Verkehrsszene und einer Warnanzeige besteht, wahr und führt am Fahrsimulator Handlungen bzw. Aktionen aus, um sein Fahrzeug zu steuern. Vom Fahrsimulator werden die aktuellen Verkehrsszenen und Aktionen des Fahrers als vorgefilterte Situationen bzw. Operatoren modelliert und an den COSA-Automaten gesendet. Der COSA-Automat interpretiert und analysiert die empfangenen Daten und sendet als Ergebnis eine interpretierte Situation an die Warnanzeige. Auf der Warnanzeige wird die interpretierte Situation so dargestellt, dass sie vom Fahrer möglichst intuitiv wahrgenommen werden kann.

4 SOM und Lernen

In [7] wurden verschiedene Konzepte für SOM-basiertes Lernen vorgestellt und teilweise am Beispiel eines mobilen Roboters implementiert. Hierbei werden Erfahrung, die eine Dreierkombination aus Anfangssituation, Operator und Endsituation sind, in einer Datenbank als Wissensbasis abgespeichert und modifiziert. Die Konzepte sind in Bild 5 dargestellt und in vier Stufen gegliedert, die im Folgenden kurz beschrieben werden.

4.1 Lernen auf Stufe 1

Jede Erfahrung, die während der Interaktion mit der Umgebung gemacht wird, erhält eine Übergangswahrscheinlichkeit. Diese besagt, wie oft ein bestimmter Operator eine bestimmte Anfangssituation in eine bestimmte Ausgangssituation überführt hat. Die Übergangswahrscheinlichkeit ergibt sich aus dem Quotienten der beiden Werte p und k . Sobald auf die Anfangssituation einer bestimmten Kombination der in der Kombination enthaltene Operator ausgeführt wird, erhöht sich der Wert k um eins. Sollte sich die, in der Kombination enthaltene, Endsituation einstellen, erhöht sich auch der Wert p um eins.

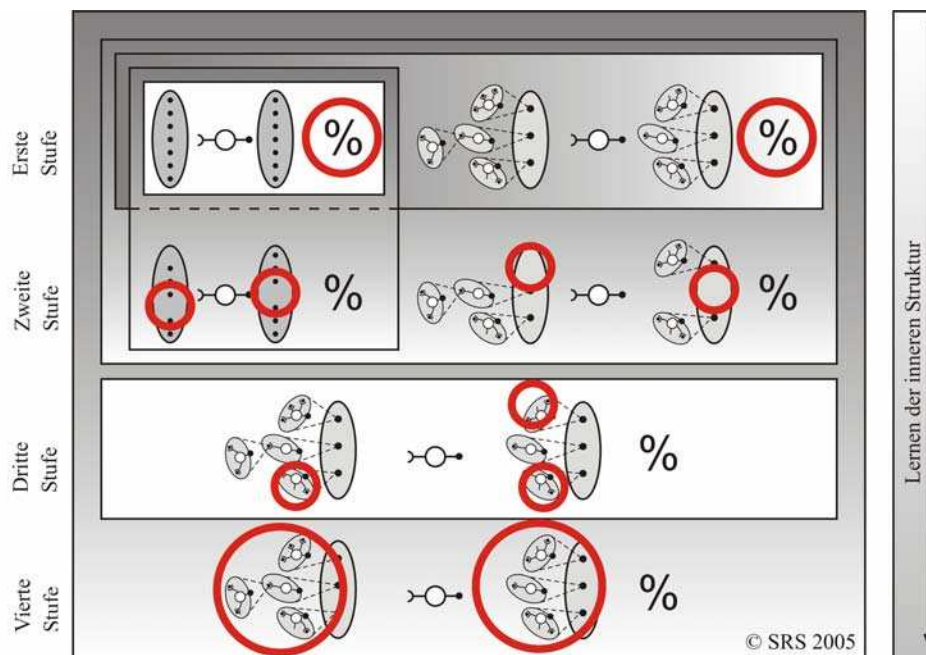


Bild 5: Die vier Stufen des SOM-basierten Lernens [8]

4.2 Lernen auf Stufe 2

Die Wissensbasis kann durch die Anwendung von Heuristiken (vgl. [9], [10]) verfeinert werden. Durch die vier Heuristiken Retraction, Exclusion, Inclusion und Restriction können die Interpretationen der Anfangs- und Endsituationen verändert werden. Die Heuristiken Inclusion und Retraction bewirken, dass ein Merkmal einer Anfangs- bzw. einer Endsituation nicht mehr beachtet wird. Bei der Exclusion und Restriction werden neue Merkmale der Anfangs- bzw. Endsituation hinzugefügt.

4.3 Lernen auf Stufe 3

Die Interpretation von Situationen kann auch durch die Anpassung der Verknüpfungen geschehen, indem deren Parameter modifiziert werden. Die Parameter von Verknüpfungen sind feste Werte, welche die Funktion der Verknüpfungen beeinflussen, wie beispielsweise Gewichtungen oder Grenzwerte.

4.4 Lernen auf Stufe 4

Neue Interpretationen von Situationen können durch die Kombination von mehreren schon bestehenden Interpretationen erstellt werden. Hierzu könnten beispielsweise genetische Algorithmen verwendet werden, die in Abhängigkeit von der Übergangswahrscheinlichkeit einer Erfahrung die erfolgreichsten Interpretationen modifizieren und / oder kombinieren.

5 Zusammenfassung und Ausblick

In den vorherigen Abschnitten wurde vorgestellt, in welcher Weise die Situations-Operator-Modellbildung zur Formalisierung von Interaktionen zwischen Menschen bzw. künstlich intelligenter Automaten und ihrer Umwelt eingesetzt werden kann. Hierdurch wird eine strukturierte Vorgehensweise vorgeschlagen, die den Verlauf von Änderungen der realen Welt als eine Folge aus Situationen und Operatoren abbildet. Als Werkzeuge zur Implementierung der SOM-basierten Modelle wurden bisher objektorientierte Hochsprachenprogrammierung und grafische Modellbildung mittels höherer Petri Netze verwendet. Der aktuelle Forschungsschwerpunkt im Zusammenhang mit Situations-Operator-Modellbildung bildet das SOM-basierte Lernen. Hierzu wurden bereits vier Verfahren konzeptionell vorgestellt, von denen zwei anhand eines mobilen Roboters implementiert wurden.

In Zukunft soll das SOM-basierte Lernen weiter detailliert und erweitert werden. Auch die Verknüpfung oder Integration von bereits bestehenden, bewährten Lernverfahren soll untersucht und verglichen werden. Hierzu bestehen eine Reihe von Anwendungsfeldern aus dem Bereich der mobilen Robotik und der Fahrerassistenz.

Literatur

- [1] McCarthy, J.: Situations, actions, and causal laws. Memo 2, Stanford University Artificial Intelligence Project, Stanford, CA, 1963
- [2] Söffker, D.: Systemtheoretische Modellbildung der wissensgeleiteten Mensch-Maschine-Interaktion. Logos-Wissenschaftsverlag, Berlin; 2003. - auch: Habilitationsschrift, Bergische Universität Wuppertal, Deutschland; 2001

- [3] Gamrad, D.: Entwicklung von Mustern höherer Petri Netze zur rechnergestützten Simulation und Analyse von Situations-Operator-Modellen. Diplomarbeit, Universität Duisburg-Essen; November 2006
- [4] Jensen, K.: Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Bd. 1-3. Springer-Verlag, 1997
- [5] Kummer, O.: Referenznetze. Berlin : Logos Wissenschaftsverlag, 2002
- [6] Söffker, D.; Ahle, E.: Supervision of human operators using a situation-operator modeling approach. Proc. 6th IFAC Symposium Fault Detection, Supervision and Safety of Technical Processes Safeprocess 2006, Beijing, China, August 30 - September 1, 2006, pp. 1015-1020.
- [7] Ahle, E.: Autonomous Systems: A Cognitive-Oriented Approach Applied to Mobile Robotics. Dissertation, Universität Duisburg-Essen; March 2007
- [8] Söffker, D.; Ahle, E.: Idea, Conception, and Realisation of Learning Abilities for Robot Control using a Situation-Operator-Model. Int. J. Intelligent Systems Technologies and Applications, Vol. 2, Nos. 2/3, 2007, pp. 271-283.
- [9] Hayes-Roth, F.: Using Proofs and Refutation to Learn from Experience. In: Michalski, R. S. (Ed.) ; Carbonell, J. G. (Ed.) ; Mitchell, T. M. (Ed.): Machine Learning: An Artificial Intelligence Approach. Morgan Kaufmann Publishers, 1983, pp. 221–240
- [10] Salzberg, S.: Heuristics for Inductive Learning. In: Proceedings of the 9th International Joint Conference on Artificial Intelligence. Los Angeles, CA, 1985, pp. 603–609

Optimale Planung und Betriebsführung im liberalisierten Energiemarkt

Sebastian Ritter, Peter Bretschneider

Fraunhofer Anwendungszentrum Systemtechnik

Am Vogelherd 50

Tel. (03677) 4610

Fax (03677) 461100

E-Mail: info@ast.iitb.fraunhofer.de

1 Einleitung

Mit der Liberalisierung der Energiemärkte traten in den vergangenen Jahren Gesetze in Kraft, die Veränderungen bezüglich der Strukturen in den Bereichen des Energieeinkaufs, der Energiebereitstellung und des Energieverkaufs der Energieversorgungsunternehmen verlangten. Durch EnWG, EEG und KWK-Gesetz sind nicht nur Unbundling und andere regulatorische Maßnahmen an die Tagesordnung getreten, sondern es sind auch neue Bezugs- und Absatzmöglichkeiten geschaffen worden, die durch Art und Struktur die Freiheitsgrade einer optimalen Planung und Betriebsführung stark erhöhen und damit das manuelle Finden des kostenminimalen Optimums unüberschaubar und damit weitestgehend unmöglich machen.

Anhand eines Praxisbeispiels soll in diesem Beitrag gezeigt werden, dass durch die Anwendung mathematischer Werkzeuge eine Abbildung der sehr komplexen Optimierungsprobleme erfolgen kann und welche Optimierungspotentiale sich im liberalisierten Energiemarkt für den Energieversorgungsprozess ergeben können. Im Weiteren werden die Ergebnisse verschiedener Szenarien verglichen und bewertet.

2 Der Energieversorgungsprozess im liberalisierten Energiemarkt

Durch die vielen Teile der Energielieferkette, die in einem liberalisierten Energiemarkt dem freien Wettbewerb unterliegen, ist der Energiemarkt durch eine vielschichtige Struktur charakterisiert. Damit verbunden ist eine hohe Anzahl an Marktteilnehmern, die die Verantwortung für die einzelnen Teilprozesse tragen (Abbildung 1).

Im Bereich der Energiebeschaffung existieren wie auch schon vor der Liberalisierung Energiebezüge auf Basis vertraglicher Vereinbarungen mit dem Energie-Vorlieferanten. Diese sogenannten Bezugsverträge sind durch verschiedenste Parameter definiert und wurden in ihrer Struktur seit der Liberalisierung novelliert. Insbesondere zeitlich variierende Preise erhöhen die Freiheitsgrade bei der Beschaffungsoptimierung.

Eine zusätzliche Möglichkeit des Energiebezuges und auch der Energielieferung bieten Energiespotmärkte und -börsen. Hier kann Energie zu Marktpreisen gehandelt werden. Um beispielsweise die Energiebörse in die zukünftige Planung einzubeziehen gibt es die Möglichkeit, anhand von Preisforwards die geplanten Energiemengen zu bewerten. Somit können Börsenprodukte im Zuge der Optimierung vorgeschlagen werden, die bei realisiertem Einkauf bzw. Verkauf das Portfolio im Sinne der Optimalität unterstützen.

Des Weiteren kann Energie auch außerbörslich gehandelt werden. Dieser OTC-

Handel zwischen Marktteilnehmern gilt für den Energieeinkauf wie auch –verkauf. Da die an den Börsen gehandelten standardisierten Produkte häufig nicht dem Wunsch der handelnden Partner entsprechen, ist der OTC-Handel zu einem wichtigen Instrument für Energiebeschaffung und den Energieabsatz geworden.

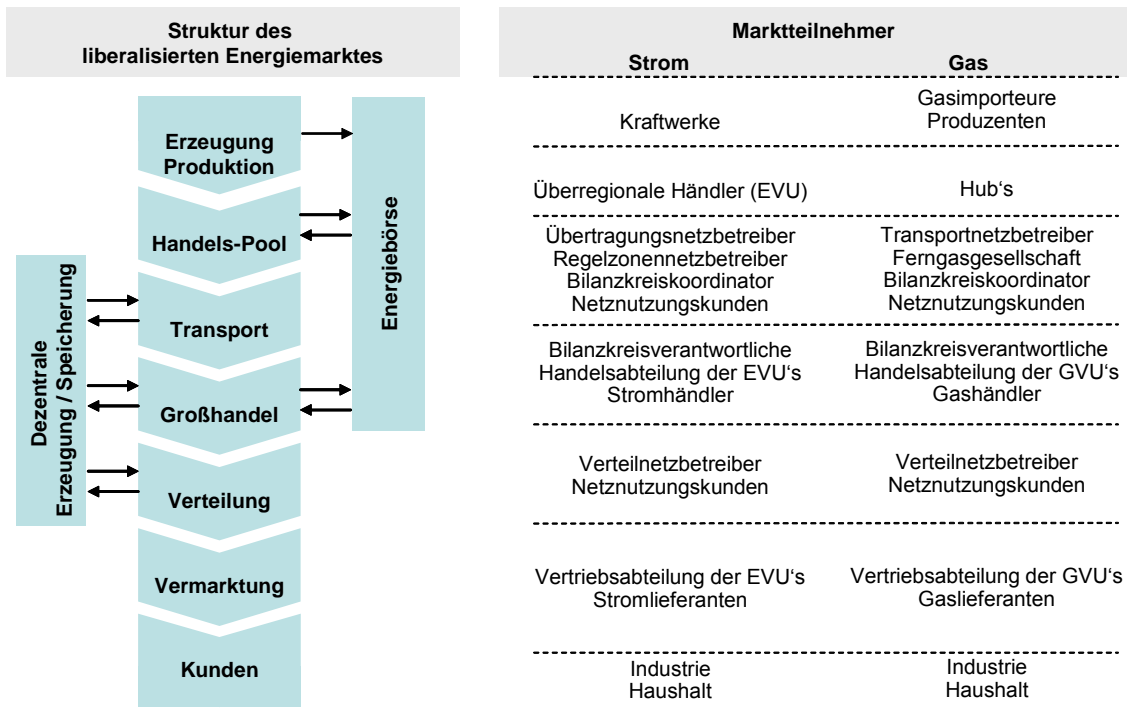


Abbildung 1: Struktur und Marktteilnehmer des liberalisierten Energiemarktes

Auf Basis des EEG gibt es im Bereich des Energiebezuges eine Energieabnahmeverpflichtung. Dies ist eine Zwangsabnahme elektrischer Energie, die auf Grundlage von Berechnungen der Übertragungsnetzbetreiber in einer festgelegten Menge und zu einem festgelegten Preis abzunehmen ist. Die sogenannte EEG-Quote ist die aufgenommene Strommenge aus durch das EEG geförderten Anlagen bezogen auf die gesamte in Deutschland an Endverbraucher abgegebene Strommenge [1]. Im Jahr 2007 beträgt die EEG-Quote schätzungsweise 13,39 % [2].

Für die sichere und wirtschaftliche Deckung der Netzlast ist die Bereitstellung von Energie durch Eigenerzeugung unerlässlich. Hierzu gibt es verschiedenste Möglichkeiten. In dem vorgestellten Beispiel geschieht dies mittels Kraft-Wärme-Kopplung, wodurch ein wesentlich höherer Wirkungsgrad bezogen auf die resultierende Nutzenergie erreicht wird als bei reinen Stromerzeugungsprozessen. Die anfallende Wärme wird zur Deckung der Fernwärmelast verwendet. In diesem Zusammenhang gibt es laut dem am 1. April 2002 in Kraft getretenen KWK-Gesetz eine Förderung auf aus KWK-Prozessen eingespeisten Strom. Es erfolgt eine Vergütung pro kWh elektrischer Energie, die von Größe und Modernisierungsrad der Anlagen abhängig ist. Damit soll eine verstärkte Nutzung der KWK-Anlagen erreicht werden, um eine weitere Minderung der Kohlendioxyd-Emissionen zu forcieren.

Um Versorgungsgenpässe auszugleichen oder Lastspitzen nicht zu ungünstigen Zeiten in die Bereiche des Vorlieferanten wirken zu lassen, werden Speicher vorwiegend im Gas- und Wärmebereich eingesetzt.

3 Optimierung des Versorgungsprozesses

Durch die Liberalisierung der Energiemärkte sind die Energieversorgungsunternehmen zunehmend unter Druck, ihre Energie immer effizienter anzubieten. Ein enormes Optimierungspotential bietet die Querverbundoptimierung.



Abbildung 2: Randbedingungen eines Versorgungsprozesses

Hierbei werden Verknüpfungen der einzelnen Sparten Strom, Gas und Fernwärme dazu benutzt, um eine ganzheitliche Betriebsplanung durchzuführen. In Folge dessen werden alle Erzeugeranlagen und Bezugsverträge spartenübergreifend dazu verwendet, das wirtschaftlichste Ergebnis zu erhalten. Die Ergänzung des Querverbunds um Spotmarkthandel/Börse und Speichernutzung bietet im Moment das größte ökonomischste Optimierungspotential. Zusätzlich kommen durch Kraft-Wärme-Kopplung (KWK) und Speichernutzung ökologisch sinnvolle Konzepte zum Einsatz.

Die Aufgabe besteht nun darin, alle zur Verfügung stehenden Ressourcen so einzusetzen, dass unter Einhaltung aller technischen, ökonomischen und ökologischen Randbedingungen die Energieversorgung garantiert ist und dabei die geringsten Kosten entstehen. Einen groben Überblick hierzu liefert Abbildung 2.

Der gesamte zu optimierende Versorgungsprozess ist symbolisch in Abbildung 3 dargestellt.

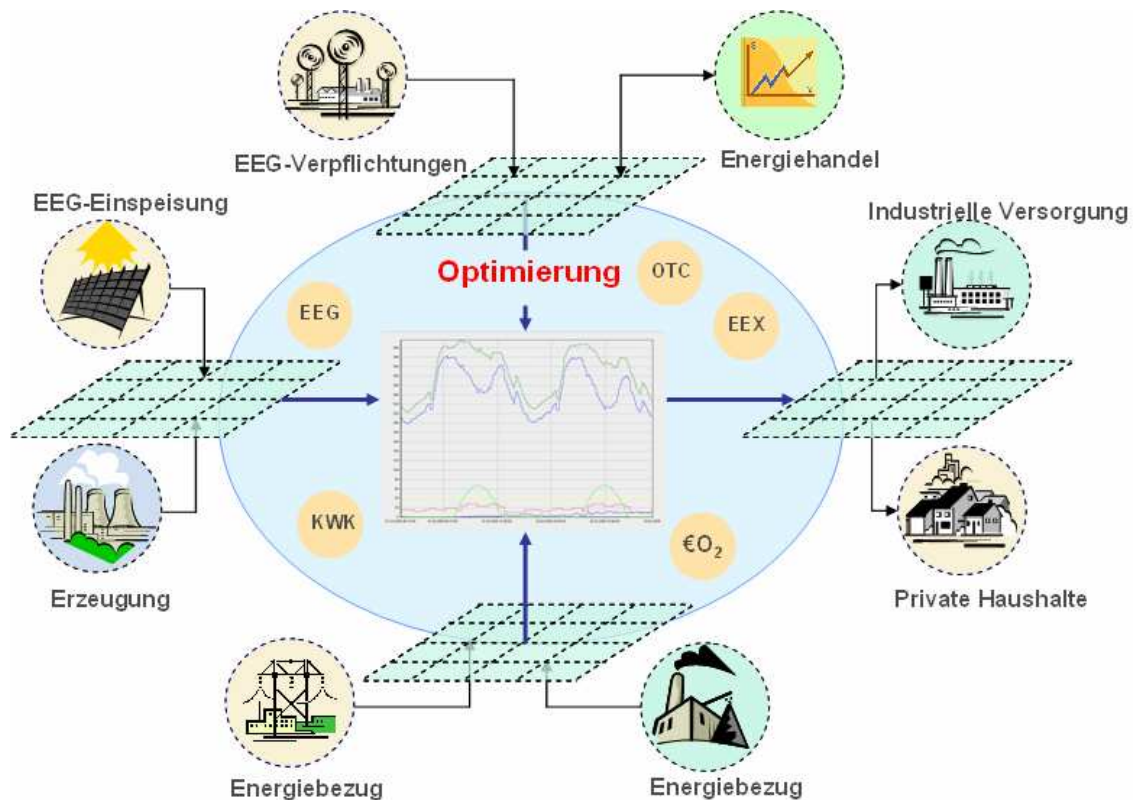


Abbildung 3: Gesamtübersicht Energieversorgungsprozess [Quelle: visIT, Fraunhofer IITB/AST]

4 Modellierung des Optimierungsproblems

Die enorme Komplexität des zu betrachtenden Gesamtsystems und die hohe Dynamik der beteiligten Prozesse erfordert die rechnergestützte Lösung einer Optimierungsaufgabe mit den Kosten als Zielfunktion. Bei dem im Folgenden vorgestellten Beispiel erfolgte die Modellierung mittels der Gemischt-Ganzzahligen-Linearen-Programmierung (GGLP). Für diese Methode existieren standardisierte Lösungsverfahren und es besteht eine hohe Flexibilität hinsichtlich der Modellzusammenstellung und damit des zu optimierenden Versorgungsprozesses [3].

Mittels GGLP gelingt eine gute Abbildung und Approximation der Prozesse und jegliche Randbedingungen können berücksichtigt werden [4].

Die Modellierung des Beispiels wurde in Gams umgesetzt und als Solver wurde CPLEX verwendet. Die einzelnen Teilaufgaben des Optimierungsmodells, wie z. B. Verträge oder Kraftwerksanlagen, wurden als separate Komponenten modelliert. Somit ist es möglich, gezielt einzelne Teile des Optimierungsproblems bei Bedarf zu deaktivieren oder durch andere zu ersetzen. Die Modellierung wird dabei durch ein GUI unterstützt, welches die mathematischen Beschreibungen in Gams-Code übersetzt. Der Aufbau des Optimierungsmodells erfolgt dabei in einem grafischen Editor.

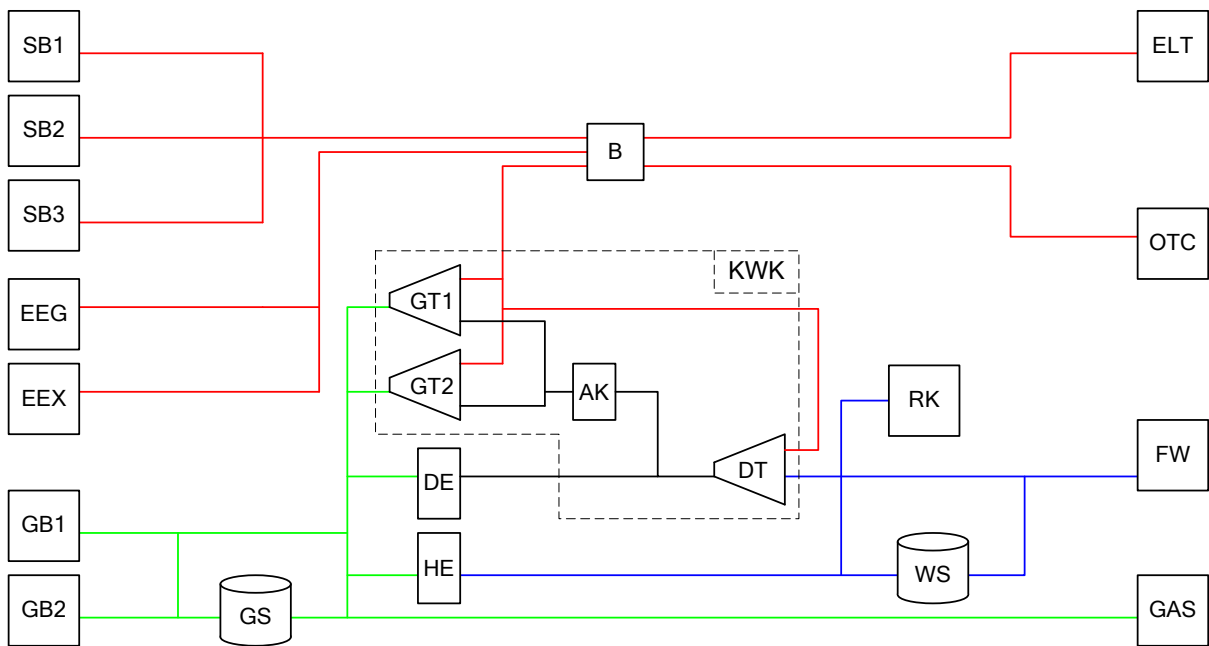


Abbildung 4: Struktur des Querverbund-Optimierungsmodells

In Abbildung 4 ist die Struktur des zu betrachtenden Optimierungsmodells dargestellt. Wobei die Abkürzungen der Komponenten folgende Bedeutungen haben:

SB1, SB2, SB3	Strombezugsverträge
GB1, GB2	Gasbezugsverträge, GB1 ist ein offener Liefervertrag
EEG	EEG-Zwangsabnahme
EEX	EEX-Strombörse
GS	Gasspeicher
WS	Wärmespeicher
GT1, GT2	Gasturbinen
AK	Abhitzeessel
DE	Dampferzeuger
HE	Heißwassererzeuger
DT	Dampfturbine
B	Bilanz, hier kann Netznutzungsproblematik abgebildet werden
ELT	Bedarfsprognose Strom
OTC	Außerbörslicher Handel
FW	Bedarfsprognose Fernwärme
GAS	Bedarfsprognose Gas
RK	Rückkühler für Überschusswärme

5 Auswertung verschiedener Kostenszenarien

Zur Verdeutlichung der Kosteneinsparung, welche durch eine Querverbundoptimierung inkl. EEX-Börse und Speicherverwendung erzielt werden kann, wurden anhand des Beispiels 4 Kostenszenarien untersucht:

1. reiner Querverbund
2. Querverbund mit EEX-Börse
3. Querverbund mit EEX-Börse und Gasspeicher
4. Querverbund mit EEX-Börse, Gas- und Wärmespeicher

Die Ergebnisse der 4 Szenarien wurden für einen Optimierungshorizont von einer Woche ermittelt und sind in Tabelle 1 dargestellt. Anhand der ermittelten Kosten kann gezeigt werden, dass die Erweiterung des Querverbunds um EEX-Börsenbezug und Speichernutzung ein immenses Einsparpotential besitzt. Die größte Kosteneinsparung konnte im untersuchten Beispiel durch die Speichernutzung erreicht werden.

Tabelle 1: Kostenszenarien, KS – Kostenstelle, E – Energie in MWh, K – Kosten in €

KS	Szenario S1		Szenario S2		Szenario S3		Szenario S4	
	E	K	E	K	E	K	E	K
SB1	1282	39771	1508	46765	1500	46514	1502	46579
SB2	221	7120	525	16662	523	16598	526	16667
SB3	2114	69774	840	27722	870	28742	876	28931
GB1	3835	46024	4076	48917	312	3755	249	2988
GB2	41192	123651	41550	124723	45282	136435	45045	135238
EEX		0	304	12298	299	11842	297	11635
HE (Steuer)		11289		11286		11290		9903
RK	202		295		285		7	
ΣE ΣK	48644	297629	48803	288373	48786	255176	48495	251941
ΔK zu S1	-	-	+159	-9256	+142	-42453	-149	-45688

Hier kann gezeigt werden dass z. B. durch Peak-Shaving Gasbezug aus teuren Bezugsverträgen vermieden werden kann. Des Weiteren sinkt die Verlustwärme des Rückkühlers, die ungenutzt an die Umgebungsluft abgegeben wird, auf ein Minimum. Bei einem Marktwert von durchschnittlich 20 € pro MWh KWK-Fernwärme ist das Einsparpotential enorm. Durch Strombezug von der Börse gelingt es ferner, zu günstigen Zeiten Bezug aus den Verträgen zu vermeiden und Kosten zu sparen. Der Gesamtenergiebezug konnte bei absoluter Lastdeckung um ca. 150 MWh verringert werden. Dies ist ein wichtiger Beitrag zum ökologisch sinnvollen Umgang mit Energie und trägt unmittelbar zur Verringerung der Treibhausgasemission bei. In Abbildung 5 sind der Energiebezug und die Kosten der untersuchten Szenarien grafisch dargestellt. Insgesamt besteht für das untersuchte Beispiel ein Einsparpotential von rund 45.000 Euro. Hochgerechnet auf ein ganzes Wirtschaftsjahr könnten für das Beispiel Kosten von rund 2,3 Mio Euro eingespart werden. Im praktischen Einsatz werden während der Optimierung noch weitere Parameter wie u. a. KWK-Vergütung, Netznutzungsentgelte und vermiedene Netznutzung berücksichtigt. Dadurch nimmt die Komplexität des Optimierungsproblems weiter zu.

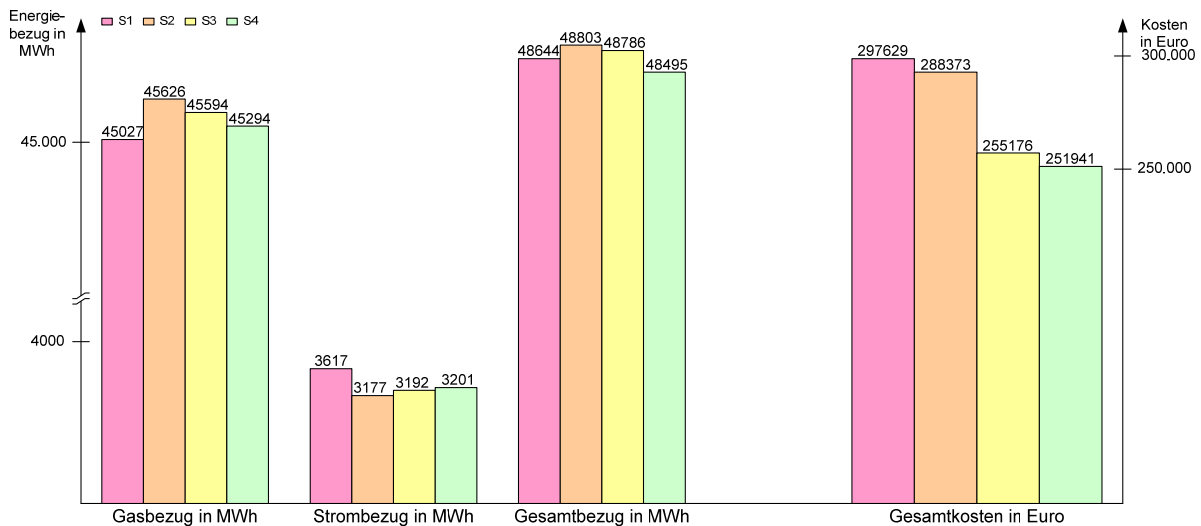


Abbildung 5: Energiebezug und Kosten der Szenarios 1 bis 4

6 Anwendung von Stochastic Programming

Die mit Unsicherheiten behafteten Eingangsgrößen der Optimierungsmodelle werden bisher mit geeigneten Vorhersagestrategien prognostiziert, die korrelative Zusammenhänge ausnutzen, um eine Aussage bezüglich des Verlaufs der Eingangsgrößen für den zukünftigen Zeitbereich treffen zu können.

Durch Stochastic Programming (SP) gelingt es, die Verteilung einer Zufallsgröße als direkten Informationseingang zu verwenden. Des Weiteren findet der zukünftige Handlungsbedarf Berücksichtigung, welcher infolge der Zufallsgröße entsteht. Dadurch ist es möglich Entscheidungen aus den Folgekosten der gesamten Zufallsgrößenverteilung generieren können [5, 6].

In aktuellen Forschungsarbeiten wird untersucht, welche Vorteile sich im praktischen Anwendungsfall durch die Berücksichtigung von Unsicherheiten während der Optimierung ergeben. Damit gelingt ein Vergleich mit der bereits langjährig im Einsatz befindlichen Optimierung mit einseitig aufgeschalteten Größen der Prognoseverfahren.

In ersten Ergebnissen ist festzustellen, dass eine pauschale Aussage über Vorteile von Stochastic Programming gegenüber den bisherigen Methoden nicht getroffen werden kann. Je nach Anwendungsfall und Dynamik der aufgeschalteten Eingangsgrößen können deterministische Optimierung basierend auf prognostizierten Größen und Stochastic Programming Vorteile gegenüber der anderen Vorgehensart aufweisen. Solange der Prognosefehler einer Vorhersagemethode kleiner ist als die Streuung der Zufallsgröße liegt der Vorteil bei den deterministischen Modellen. Bei Modellen, wo Größen vorhanden sind, welche sich schlecht bis gar nicht prognostizieren lassen und damit keine Aussagen über die Zukunft möglich sind, ist die kostenoptimale Positionierung des SP-Modells die beste Wahl [5]. Dies entspricht dem eigentlichen Grundgedanken zum Einsatzbereich von Stochastic Programming.

In weiterführenden Arbeiten wird ein hybrider Ansatz untersucht, der die Vorteile beider Technologien ausnutzen kann. Als Aufgabe steht hierbei die Minimierung des bestehenden Prognosefehlers durch Anwendung des Stochastic Programming.

7 Zusammenfassung

Zusammenfassend kann festgestellt werden, dass sich durch die ganzheitliche Betrachtung des Versorgungsprozesses während der Optimierung immense Kosten- und Energieeinsparungen erreichen lassen. In dem untersuchten Beispiel ist eine ökologisch verantwortungsvolle Nutzung der Energie erst durch Nutzung von Wärmespeichern erreicht worden. Denn erst im vierten Szenario konnte der Gesamtenergiebezug gegenüber der Ausgangssituation vermindert werden. Eine reine Kosteneinsparung ist schon im zweiten Szenario erreicht worden.

Die verzweigten Strukturen und Prozesse als Konsequenz der Liberalisierung im Energiesektor können durch mathematische Beschreibungen unter Beachtung jeglicher Randbedingungen abgebildet werden. Die damit generierten Optimierungsmodelle liefern bei Nutzung der marktspezifischen Freiheitsgrade optimale Vorschläge, für einen wirtschaftlichen Betrieb der beteiligten Strukturen/Anlagen, der auf manuellem Wege im Kontext der geringsten Kosten aufgrund der Komplexität nicht mehr möglich ist.

8 Literatur

- [1] Bundesverband Windenergie e.V.,
“<http://www.wind-energie.de/de/themen/windenergie-von-a-z/eeg-quote/>”
- [2] VDN e.V., „http://www.vdn-berlin.de/eeg_prognose_2007.asp“
- [3] Hanselmann, M.: Entwicklung eines Programmsystems zur Optimierung der Fahrweise von Kraft-Wärme-Kopplungsanlagen, Universität Stuttgart, 1996
- [4] Jonas Albiger: Integrierte Ressourcenplanung in der Energiewirtschaft mit Ansätzen aus der Kraftwerkseinsatzplanung, Universität Stuttgart, 1997
- [5] Hanf, R.: Untersuchung der Leistungsfähigkeit der stochastischen Optimierung für die optimale Betriebsführung energiewirtschaftlicher Problemstellungen, Diplomarbeit, TU Ilmenau, 2007
- [6] Römisch, W.: Optimierungsmethoden für die Energiewirtschaft - Stand und Entwicklungstendenzen, VDI Bericht 1627, 2001

Heinrich Hertz, Ludwig Wittgenstein und die Fuzzy-Strukturen

Eine kleine „Bildergeschichte“ zur Erkenntnisphilosophie

Rudolf Seising

Medizinische Statistik und Informatik, Medizinische Universität Wien
Spitalgasse 23, A-1090 Wien, Österreich
Tel. (+43) 1 404006664
Fax (+43) 1 404006667
E-Mail: rudolf.seising@meduniwien.ac.at

1 Bildergeschichten

Bildergeschichten beflügeln die Phantasie, seien es Bilderbücher für Kinder, Comics für Jugendliche (und Erwachsene), Illustrierte Zeitschriften oder auch Filmmaterial. Es heißt: Ein Bild sagt mehr als tausend Worte! Der immer weiter verbreitete Einsatz über einen Beamer projizierter Computerbildschirm-Präsentationen bei Vorträgen auch im wissenschaftlichen Bereich bestätigt, dass sich komplizierte Sachverhalte oftmals mit Hilfe von Bildern, Ablaufschemata, strukturierten Darstellungen etc. leichter verständlich darstellen und/oder erklären lassen, als mit langen Textpassagen. Zudem haben Bilder auch im Bereich der Wissenschaft eine lange Tradition, angefangen bei den Illustrationen in den ersten noch von Hand vieler Mönche geschriebenen Büchern bis zu den heutigen CD-Roms der verschiedensten digitalen Bibliotheken. Hervorzuheben ist beispielsweise der tschechische Theologe und Pädagoge Johann Amos Comenius (1592-1670), der in seinem 1657 erschienenen Sach- und Sprachlehrbuch *Orbis sensualium Pictus* (*Die gemalte Welt*), das für ein ganzes Jahrhundert das verbreitetste Lehrbuch Deutschlands war, dem lernenden Kind zu den in deutschen und lateinischen Sätzen beschriebenen Gegenständen auch Abbildungen zeigte, so dass es von allen Dingen, deren Namen es liest, eine Anschauung hatte.

Etwa zur gleichen Zeit nutzte René Descartes (1596-1650) die Möglichkeit von Abbildungen, um seine Theorie der Sinneswahrnehmung zu verdeutlichen (Abb. 1).

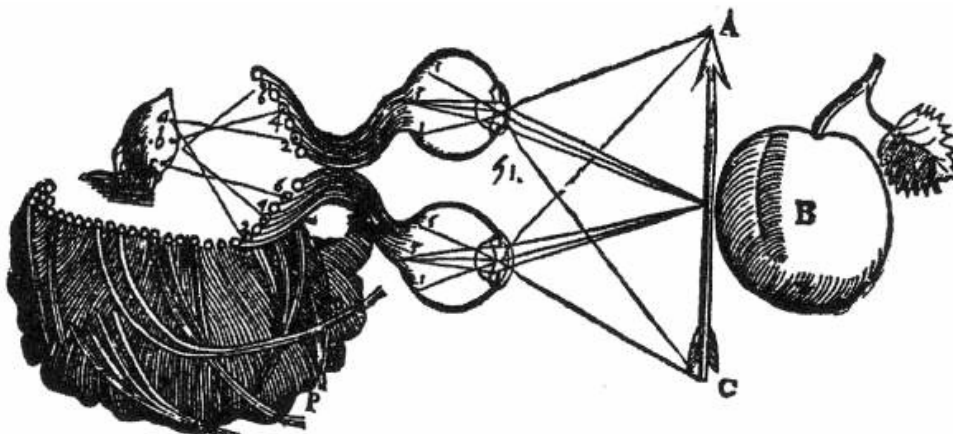


Abb. 2: Netzhautbilder eines Pfeils (A) und eines Apfels (B) werden zu Eindrücken (a und b) in der Zirbeldrüse, wo Descartes den Sitz der menschlichen Seele annahm. Abb.: [4].

In seinem 1649 veröffentlichten Buch *Les passions de l'âme*. (*Die Leidenschaften der Seele*) zeigte er z. B. anhand einer Zeichnung (Abb. 1), wie er sich die Verschmelzung der Netzhautbilder in den beiden Augen des Menschen zu einem gemeinsamen optischen Eindruck in der Zirbeldrüse dachte. Ausgehend davon, dass die Welt aus ausgedehnter Substanz (Materie, *res extensa*) und nicht ausgedehnter Substanz (Geist, *res anima*) besteht, nahm Descartes an, dass der Mensch als einziges Lebewesen an beiden Substanzen teilhat und Wechselwirkungen beider Substanzen in der Seele erlebt (Leib-Seele- oder Körper-Geist-Problem). [2].

Im letzten Jahr wurden erstmals und mit großem Medienecho Comics in einem Übersichtsartikel für eine wissenschaftliche Zeitschrift mit peer-review-Verfahren verwendet, um wissenschaftliche Ergebnisse zu veranschaulichen. Der Artikel "Behavioural assessment of drug reinforcement and addictive features in rodents: an overview" von Carles Sanchis-Segure und Rainer Spanagel erschien in der Zeitschrift *Addiction Biology* [1]. Die beiden Wissenschaftler vom *Mannheimer Zentralinstitut für Seelische Gesundheit*, berichten dort über ihre Erforschungen der neurobiologischen Grundlagen von Alkoholabhängigkeit, Angst- und Depressionserkrankungen. Dazu wurden Ratten bzw. Mäuse alkoholabhängig gemacht und dann verhaltenspharmakologisch und neurochemisch untersucht.



Abb. 2: „Alcohol Deprivation Effect“: Freiwillig alkoholtrinkenden Mäusen wird der Alkohol entzogen, später wieder angeboten. Die Tiere trinken nun viel mehr Alkohol, meist bis zur Intoxikation (Comic: Hollstein)

Die Autoren ließen das Verhalten der Versuchstiere in ihrem Artikel mit Comics des Schweizer Comicszeichners Walter Hollenstein illustrieren [2], weil sie hofften, dass auf diese Weise „die interdisziplinären Kommunikationsbarrieren abgebaut werden“ ... und „komplexe wissenschaftliche Sachverhalte auch einer breiten Öffentlichkeit leichter zugänglich gemacht werden können.“ [3] Abb. 2 zeigt als Beispiel den Comicstrip über den Rückfalleffekt in der Alkoholforschung [4].

Auch wenn Bilder zwar gezeichnet, gemalt, photographiert oder gefilmt wurden, so entstanden sie doch zuvor in der Vorstellung eines Menschen. Descartes hatte Erfahrung über das Körperinnere, da er viele Tiersektionen selbst durchgeführt und bei mindestens einer Sektion eines menschlichen Gehirns zugesehen hatte. Die Zeichnung der Augen und ihrer Verbindung zur Zirbeldrüse im Gehirn eines Menschen bildet das ab, was er sich aufgrund seiner Erfahrungen vorstellte. Auch wenn seine Theorie heute nicht mehr vertreten wird, entsteht beim Betrachter von Abb. 1 erneut ein Abbild dieser Vorstellung, von der er vor dem Hintergrund heutiger wissenschaftlicher Ergebnisse allerdings nicht annehmen wird, dass sie der Wirklichkeit entspricht. Der Comic in Abb. 2 bildet die Geschehnisse im Labor ebenfalls nicht realitätsgetreu ab, in diesem Falle war das auch niemals die Absicht des Zeichners gewesen, und ein Betrachter dieses Bildes wird dies auch nicht glauben. Allerdings entsteht auch bei der Betrachtung des Comics eine Vorstellung, die jener ähnelt, die die Mannheimer Wissenschaftler hatten, und nach der Hollenstein seinen Comic zeichnete. Bilder sind keine exakten Abbildungen der Realität, zuweilen betonen sie lediglich gewisse Aspekte der Wirklichkeit, mit der sie ansonsten nur wenig zu tun haben. Es sind Modelle der Wirklichkeit und insofern immer auch Interpretationen des Dargestellten: Weder ist die Zirbeldrüse der Sitz der menschlichen Seele und damit der Ort, an dem die materielle Substanz (*res extensa*) und die geistige Substanz (*res anima*) interagieren können, wie Descartes annahm, noch sitzen die Mäuse im Versuchslabor vor dem Fernsehgerät, wie im Comicstrip gezeigt. Dennoch kann aus den Bildern Wissen gewonnen, etwas gelernt werden: Wenn wir von der Annahme ausgehen, dass es eine materielle und eine geistige Substanz gibt, dann ist nicht geklärt, wie der Mensch Wechselwirkungen zwischen den beider Substanzen durchführt oder erlebt (Leib-Seele- oder Körper-Geist-Problem). Die Mäuse zeigen Reaktionen auf Reize, und durch die „Vermenschlichung“ als Comic-Figuren sind diese verhaltenstheoretisch interpretierbar, auch wenn diese Mäuse in der Realität – anders als auf den Comic-Zeichnungen – nichts zu lachen haben! Auch dann nicht, wenn ihnen der Alkohol, von dem sie zuvor abhängig gemacht wurden, wieder angeboten wird.¹

2 Bilder und Modelle

Bilder, Modelle werden in der Wissenschaft gebraucht, um Theorien darzustellen. Berühmte Beispiele sind die Abbildungen des geozentrischen bzw. heliozentrischen Weltbildes in dem 1660 erschienenen bzw. 1708 zum zweiten Mal aufgelegten Himmelsatlas *Harmonia Macrocosmica* von Andreas Cellarius (1503-1562).

Im geozentrischen Weltbild (griechisch γεοκεντρικό: erdzentriert) steht die Erde in der Mitte des Universums und alle anderen Himmelskörper (Mond, Sonne, Planeten) kreisen in Sphären (durchsichtigen Hohlkugeln) um die in der Mitte befindliche Erde, während die Fixsterne die äußerste Sphäre besetzen (Abb. 4, links). Es stammt aus dem antiken Griechenland, wo es u. a. von Hipparchos von Nikaia (~190-120 v. Chr.), Aristo-

¹ Die ethischen Probleme solcher Tierversuche werden hier ausdrücklich nicht diskutiert, ich möchte aber betonen, dass diese ethischen Probleme existieren und sicherlich diskutiert werden sollten!

teles (384-322 v. Chr.) und Ptolemäus (~100~175) gelehrt wurde. Dieses daher oft auch „Ptolemäisch“ genannte Weltbild war bis ins späte Mittelalter in Europa verbreitet, da es von den christlichen Kirchen vehement vertreten und verteidigt wurde. Schon mit dem 1543 von Nicolaus Copernicus (1473-1543) geschriebenen Buch *De Revolutionibus Orbium Coelestium* setzte eine wissenschaftliche Revolution ein, mit der das geozentrische vom heliozentrischen Weltbild (ἥλιος: Sonne) abgelöst wurde. Dieses bald auch „Copernicanisch“ genannte Weltbild war schon lange vor dem Ptolemäischen in der Antike vertreten worden, und es gilt als gesichert, dass Copernicus sich von dieser heliozentrischen Theorie des antiken Astronomen Aristarchos von Samos (~310-230 v. Chr.) leiten ließ. Nachdem Johannes Kepler (1571-1630) dann anstelle von Kreisbewegungen die mathematischen Gesetze der elliptischen Planetenbewegung eingeführt hatte, wurde am geozentrischen Weltbild immer weniger festgehalten und stattdessen das heliozentrische Weltbild vertreten, in dem sich alle Planeten in Ellipsen um die Sonne bewegen (Abb. 4, rechts).

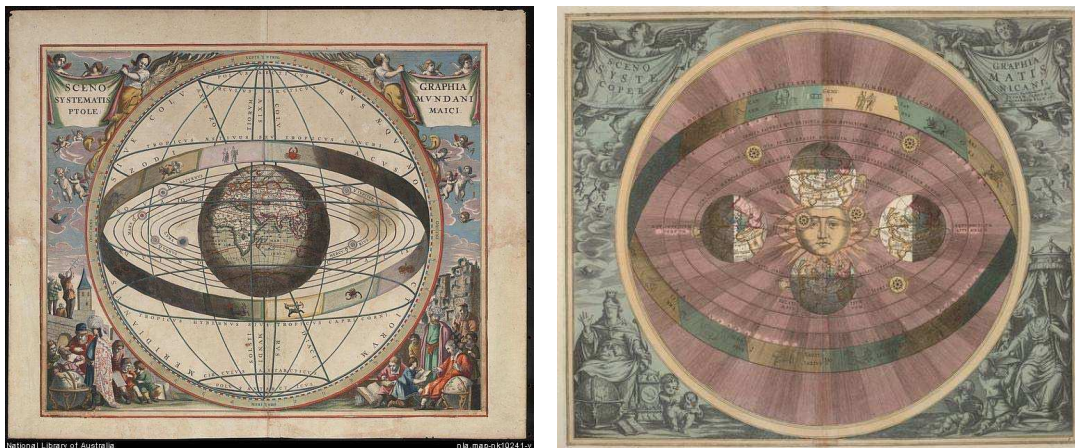


Abb. 1: Links: Geozentrisches Weltbild, Quelle: [5], rechts: heliozentrisches Weltbild, Quelle: [6].

Weitere Beispiele für wissenschaftliche Theorien darstellende Bilder sind die im 20. Jahrhundert aufgestellten „Atommodelle“, also wie sich Wissenschaftler Atome vorstellten, nachdem deren Existenz lange Zeit erbittert diskutiert, schließlich aber als gesichert angesehen wurde.

Der britische Physiker Joseph John Thomson (1856-1940) hatte 1897 das Elektron entdeckt und kurz darauf das heute nach ihm benannte Atommodell entwickelt: Er stellte sich das Atom als eine gleichmäßig verteilte positive Ladung vor, in die die sehr kleinen Elektronen eingebettet sind. Dieses „plum pudding model“ (deutsch: „Rosinenkuchenmodell“) wurde 1911 von dem britischen Physiker Ernest Rutherford (1871-1937) korrigiert. Rutherfords Streuversuche mit Alphateilchen auf einer Goldfolie zeigten, dass die Verteilung der positiven Ladungen im Atomkern nicht gleichmäßig ist. Das Rutherford'sche Atommodell besagt, dass Atome aus einem positiv geladenen Atomkern mit nahezu aller Atommasse und einer Atomhülle mit den darin um den Kern kreisenden Elektronen bestehen. Mit diesem „planetarische Modell“ übernahm Rutherford die schon im Jahre 1904 vom japanischen Physiker Nagaoka Hantarō (1865-1950) aufgestellte „saturnian theory“ des Atoms.

Vom „Planetenmodell“ abweichend und einem „Sonnensystemmodell“ ähnlicher ist das 1913 entstandene Atommodell des dänischen Physikers Nils Bohr (1885-1962). Demnach besteht das Atom aus einem positiv geladenen, die Masse enthaltenden Kern und Elektronen, die diesen auf diskreten Bahnen umkreisen. Damit konnte Bohr zwar Wi-

dersprüche des Rutherford-Modells mit der klassischen Theorie der Elektrodynamik lösen, er handelte sich allerdings neue Probleme ein.²

Noch ähnlicher einem „Sonnensystemmodell“ war das vom deutschen Physiker Arnold Sommerfeld (1868-1951) um elliptische Bahnen ergänzte Bohr-Sommerfeldsche Atommodell, doch auch dieses Modell hielt sich nicht lange, und seit der Begründung der Quantenmechanik ist fraglich, wie ein Bild vom Atom aussehen kann, wie wir es uns vorstellen können: „*Versuchen Sie es gar nicht erst!*“ hat darauf der deutsche Physiker Werner Heisenberg (1901-1976) geantwortet.

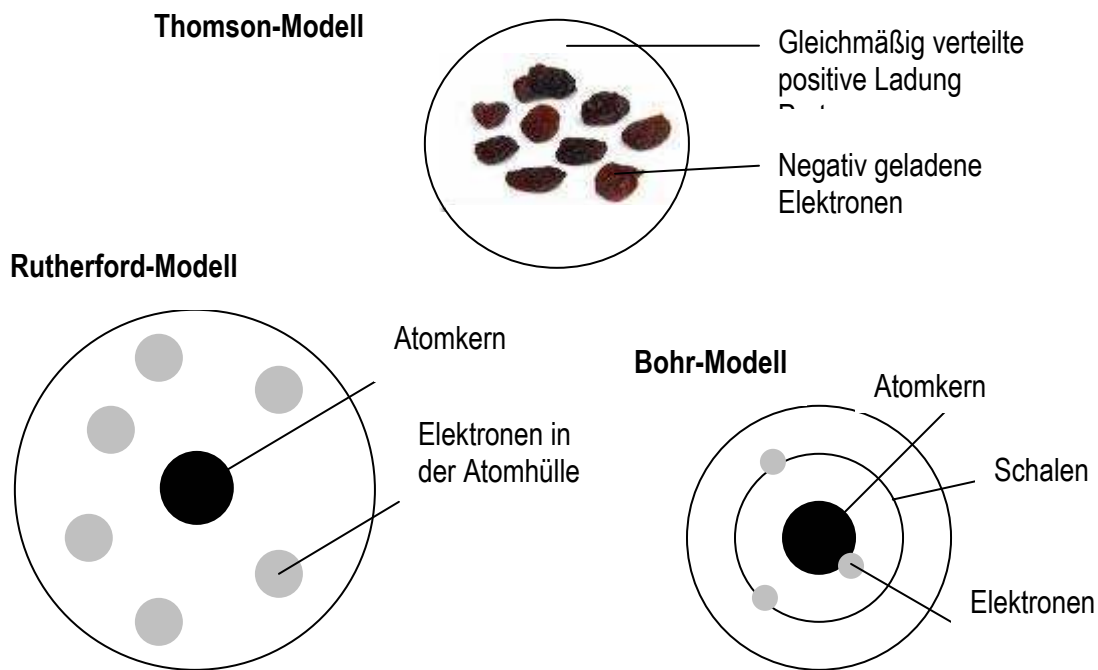


Abb. 2: Oben: Thomsons Atommodell. Unten, links: Rutherfords Atommodell, rechts: Bohrs Atommodell.

Obwohl der Abstraktionsgrad vieler wissenschaftlicher Theorien im 20. Jahrhundert ungeheuer zugenommen hat, stellen sich Wissenschaftler weiterhin ihre Forschungsobjekte vor; sie brauchen Bilder, auf die sie ihre empirischen Beobachtungsergebnisse reduzieren, um Theorien entwickeln zu können. Im Folgenden soll dazu die erkenntnistheoretische Position des deutschen Physikers Heinrich Rudolf Hertz (1857-1894) betrachtet werden, wie dieser sie in der Einleitung zu seinem Buch *Die Prinzipien der Mechanik in neuem Zusammenhange dargestellt* ausgeführt hat, das 1894 posthum von Philipp Lenhard (1862-1947) herausgegebenen wurde [8]. Hertz (Abb. 3), der vor 150 Jahren geboren und nur 36 Jahre alt wurde, hatte in den 1880er Jahren die von Michael Faraday (1791-1867) und James Clerk Maxwell (1831-1879) vorausgesagte Existenz elektromagnetischer Wellen nachgewiesen. Damit ist er einer der Wegbereiter heutiger Informations- und Kommunikationstechnologie. Der großen Öffentlichkeit weniger bekannt sind seine Überlegungen zur Naturerkenntnis, die er in dem genannten Buch entfaltetete, und mit denen er „einen der großen klassischen Texte der Philosophie geschaffen“ hatte – wie sein Biograph Albrecht Fölsing schrieb [9]. Bereits in seiner 1884 geschriebenen Kieler Vorlesung *Die Constitution der Materie*, die er allerdings nicht veröffentlichte, und die Fölsing erst 1999 herausgab, hatte Hertz „eine Konzeption der

² Auf diese Details kann an dieser Stelle leider nicht eingegangen werden.

»Bilder« zur Beschreibung der Wirklichkeit“ entwickelt, die er dann gut ein Jahrzehnt später in den *Prinzipien der Mechanik* ausarbeitete.

3 Heinrich Hertz: Wenn unser Verstand uns Bilder schafft ...

Hertz' Ausgangspunkt war die Atomhypothese: Die Frage war: Gibt es Atome oder sind sie nur eine mathematische Hilfskonstruktion? Hertz schrieb damals: „Ich kann mich darauf beschränken, es als meine Aufgabe zu betrachten, die sinnlich wahrnehmbaren Thatsachen möglichst einfach zu beschreiben, alles, was über die sinnliche Wahrnehmung hinausgeht, ist dann Fiction, die der Beschreibung dient und den Zweck hat, diese Beschreibung zu vereinfachen.“ ([8], S. 35) Dann argumentiert Hertz jedoch, dass die Physik nicht zu dieser Beschränkung gezwungen sei: „Es ist eine allgemeine und nothwendige Eigenschaft des menschlichen Verstandes, daß wir uns die Dinge weder anschaulich vorstellen noch sie begrifflich definiren können, ohne ihnen Eigenschaften hinzuzufügen, die in ihnen an sich durchaus nicht vorhanden sind. Das geschieht nicht nur in allem Denken und Vorstellen des gewöhnlichen Lebens, auch die Wissenschaften verfahren nicht anders; einzig die Philosophie sucht und findet den Unterschied zwischen den Dingen, die wir wahrnehmen, aber sie sieht auch die Nothwendigkeit dieses Unterschiedes ein.“ ([8], S. 35)



Abb. 3: Links: Heinrich Hertz, rechts: Titelseite seines philosophischen Hauptwerks.

In der doch unbezweifelbar exakten Wissenschaft der Geometrie müssen wir uns die Raumgebilde doch vorstellen, schreibt Hertz: „Aber alle diese sind solche, deren sinnliche Vorstellung unmöglich ist, wenn wir nicht ihnen Eigenschaften verleihen, von denen die Geometrie nichts wissen will, von denen wir ausdrücklich abstrahiren sollen. Wenn uns gesagt wird: stelle Dir eine unendlich dünne Kugelschale oder ein unendlich kleines Raumelement vor, so erscheinen vor unserem geistigen Auge die gewünschten Objecte, aber weder erscheinen dieselben unendlich dünn oder unendlich klein, noch ohne Farbe oder ohne andere Eigenschaften, die absolut dem eigentlich gewollten Object fremd sind.“ ([8], S. 36) In der Physik werde man zu ähnlicher Weise zu unmöglichen Vorstellungen aufgefordert: „Denke dir ein Atom als einen kugelförmigen mit Materie erfüllten Raum von 1 Millionen mm Durchmesser [...] Freilich, wir können uns einen solchen Raum in wirklicher Größe nicht vorstellen, wir können ihn nicht erfüllt denken, ohne ihn mit Glas, Eisen oder irgend einer bestimmten Materie erfüllt zu den-

ken, aber wir können doch uns klar machen, was von diesen Eigenschaften unwesentlich ist, und ein Kern wird bleiben, der die wesentlichen Eigenschaften, um die es uns zu thun ist, besitzt. Was wir hinzufügen, sind dann falsche Vorstellungen überhaupt; wir können sie nicht fortnehmen und bessere an ihre Stelle setzen, sondern wir müssen sie hinzuthun oder auf alle Vorstellungen in diesem Gebiete verzichten.“ ([8], S. 36)

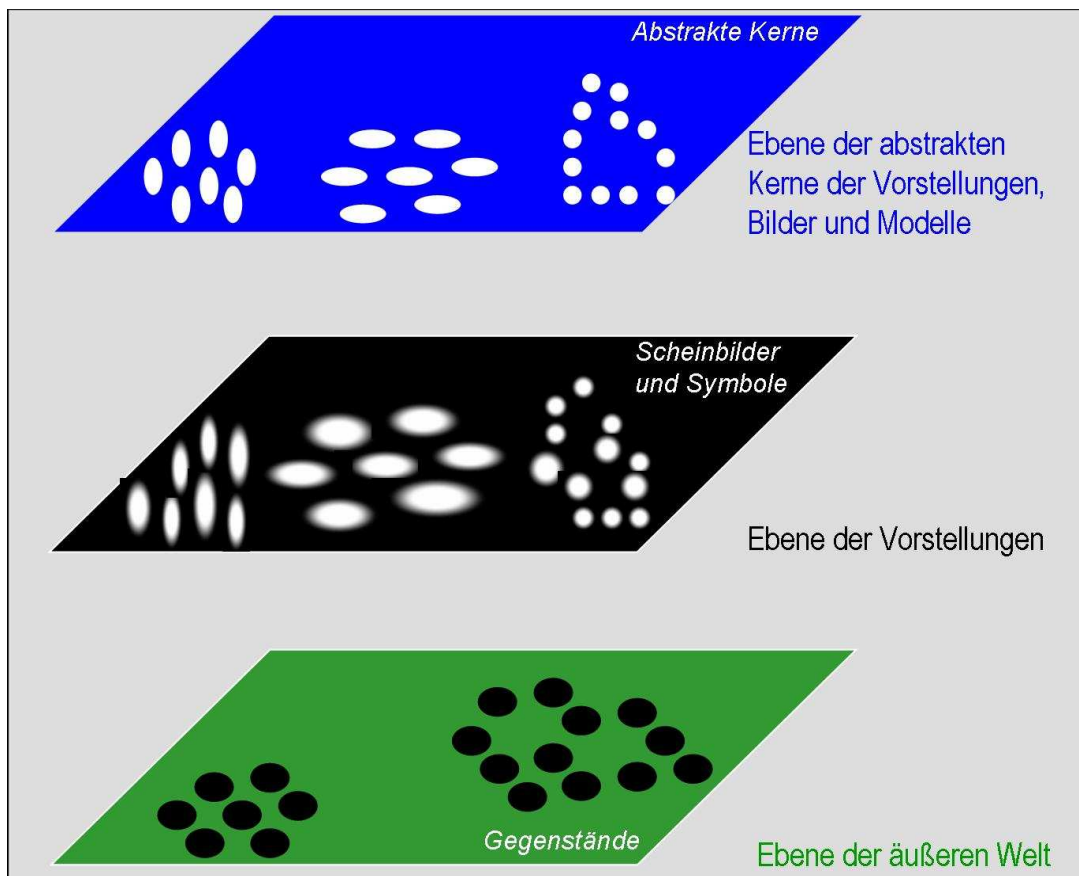


Abb. 4: Versuch einer Darstellung der Erkenntnisphilosophie von Heinrich Hertz. In der untersten Ebene befinden sich die Gegenstände der äußeren Welt, darüber liegt die Ebene der Scheinbilder. Abstrahieren wir bei diesen allerdings noch von den „unwesentlichen Eigenschaften“ („falsche Vorstellungen überhaupt“), so bleiben „Kerne“ zurück. Die Ebene dieser Kerne ist die obere Ebene.

In der Einleitung zu seinen *Prinzipien der Mechanik* hat Hertz nach Meinung des deutschen Mathematikers und Physikers Max Born (1882-1970) „wohl die erhellendsten Sätze über das Wesen der theoretischen Naturforschung überhaupt“ geschrieben [9]; sie lauten: „Es ist die nächste und in gewissem Sinne wichtigste Aufgabe unserer bewußten Naturerkenntnis, daß sie uns befähige, zukünftige Erfahrungen vorauszusehen“ ([8], S. 66). Born erläuterte dies 78 Jahre später für seine Zeit so: „mittels der theoretischen Systeme der Physik lassen sich nämlich Voraussagen ableiten, ‚zukünftige Erfahrungen vorauszusehen‘, wie es Hertz ausdrückte“. [9]

Zukünftige Erfahrungen vorauszusehen befähige uns „unser gegenwärtiges Handeln einrichten zu können“ schrieb Hertz ([8], S. 66). Unser wissenschaftliches Erkennen gründet auf unserer Erfahrung, die unser Verstand in „Bilder“ fasst, wie Hertz dann in Sätzen ausdrückte, die Born für „wohl die knappste, klarste und treffendste Formulierung dessen, was unter der modernen Physik zu verstehen ist“ hielt: Wir machen uns innere Scheinbilder oder Symbole der äußeren Gegenstände, und zwar machen wir sie von solcher Art, daß die denotwendigen Folgen der Bilder stets wieder die Bilder seien von den naturnotwendigen Folgen der abgebildeten Gegenstände.“ ([10], S. 67)

Dass es die dazu notwendige Übereinstimmung zwischen der Natur und unserem Geiste gibt, lehre uns die Erfahrung: (*Logisch*) *unzulässig* sind für Hertz jene Bilder, „die einen Widerspruch gegen die Gesetze unseres Denkens in sich tragen“. Darüber hinaus können solche zulässigen Bilder *unrichtig* sein, „wenn ihre wesentlichen Beziehungen den Beziehungen der äußeren Dinge widersprechen“. Es bleiben dann die „richtigen“ Bilder übrig. Der Plural ist hier wichtig, denn es kann mehrere „richtige“ Bilder“ derselben äußeren Gegenstände geben, die sich aber hinsichtlich ihrer „Zweckmäßigkeit“ unterscheiden: „Von zwei Bildern desselben Gegenstandes wird dasjenige das zweckmäßigere sein, welches mehr wesentliche Beziehungen des Gegenstandes widerspiegelt als das andere; welches, wie wir sagen wollen, das deutlichere ist. Bei gleicher Deutlichkeit wird von zwei Bildern dasjenige zweckmäßiger sein, welches neben den wesentlichen Zügen die geringere Zahl überflüssiger oder leerer Beziehungen enthält, welches also das einfachere ist.“ ([8], S. 68)

Hertz' Auffassung von wissenschaftlichen Theorien als Bilder, die vom Verstand des Wissenschaftlers auf Grund bisher gesammelter Erfahrungen geschaffen werden, steht im Gegensatz zu der vorher herrschenden Ansicht, dass solche Theorien objektiv fest stehen und vom Wissenschaftler nur gefunden werden, und dass nicht mehrere sondern nur eine richtige Theorie geben könne. Hertz' Erfahrungen mit der Elektrodynamik haben ihn gelehrt, dass es verschiedene Theorien mit unterschiedlichen Begriffssystemen geben, sich aber mit der Zeit auch eine Theorie durchsetzen kann. (Für den Versuch einer Darstellung seiner Erkenntnisphilosophie siehe Abb. 4.) In seiner „Bildersprache“ formulierte er: „Was den Bildern zukommt um ihrer Richtigkeit willen, ist enthalten in den Erfahrungstatsachen, welche beim Aufbau der Bilder gedient haben. Was den Bildern zukommt damit sie zulässig seien, ist gegeben durch die Eigenschaften unseres Geistes. Ob ein Bild zulässig ist oder nicht, können wir eindeutige mit ja und nein entscheiden, und zwar mit Gültigkeit unserer Entscheidung für alle Zeiten. Ob ein Bild richtig ist oder nicht, kann ebenfalls eindeutig mit ja und nein entschieden werden, aber nur nach dem Stand unserer gegenwärtigen Erfahrung und unter Zulassung der Berufung an spätere reifere Erfahrung. Ob ein Bild zweckmäßig sei oder nicht, dafür gibt es überhaupt keine eindeutige Entscheidung, sondern es können Meinungsverschiedenheiten bestehen. Das eine Bild kann nach der einen, das andere nach der andern Richtung Vorteile bieten, und nur durch allmähliches Prüfen vieler Bilder werden im Laufe der Zeit schließlich die zweckmäßigsten gewonnen.“([8], S. 68f)

Hertz wandte sich dann – dem Titel seines Buchs gemäß – drei verschiedenen Bildern der Mechanik zu.³ Leser aus dem nachfolgenden Jahrhundert konnten seine Erkenntnisphilosophie allerdings auch vor dem Hintergrund der damals neuen wissenschaftlichen Theorien, der Relativitätstheorie und der Quantenmechanik sehen, z. B. der schon oben zitierte Max Born, dessen Ausführungen Josef Kuczera in seinen Anmerkungen zum Hertz'schen Text paraphrasierte: „Der Physiker hat sich ein gedankliches, „abstraktes“ Modell von der Natur zurechtgezimmert und versucht dabei, mit möglichst wenigen, einheitlichen Typen von „Bauklötzchen“ auszukommen. Wie er aber damit sinnvoll „spielen“, also „operieren“ dürfe, das sagt ihm der Mathematiker, indem er eine Auswahl von Rechenverfahren zur Verfügung stellt. Ob nun die „Bauklötzchen“, die Hertz „innere Scheinbilder“ oder „Symbole“ nennt, richtig geformt seien, das kontrolliere der Physiker im Experiment an der Natur selbst. Seine Experimente sind seine „Fragen an die Natur“. Die Versuchsaufbauten verraten dem Physiker, welches mathematische Re-

³ Auf diese Aspekte seiner Einleitung zu den Prinzipien der Mechanik kann hier leider nicht eingegangen werden. Siehe dazu: [8], S. 70-139.

chenverfahren er für seine „Baukastenspiele“ zugrundelegen dürfe. Jede mögliche „Operation“ müsse in der Natur ihr Spiegelbild haben. Daß Baukästchen und Spielregeln manchmal geordnet werden müssten, haben die komplizierten Ergebnisse gezeigt, die den Anstoß für die Quanten- und Relativitätstheorie gaben. Für diese Wissenschaft sei es eine zentrale Aufgabe geworden, daß der Physiker versucht, mit immer einfacher geformten „Klötzchen“ immer komplizierter werdende Sachverhalte seiner Wissenschaft darzustellen. Der Physiker hat sich eine *künstliche* Bilderwelt geschaffen, um *natürliche* Dinge und Erscheinungen zu beschreiben.“ ([10], Anm. 35.)

Offensichtlich hat Born die Vorstellungen und (Schein)Bilder von Hertz als mathematische Objekte angesehen und die von Hertz betonte Unterscheidung zwischen den Vorstellungen und ihren Kernen nicht beachtet. Hertz abstrahierte diese Kerne von den Vorstellungen, die ja noch „unwesentliche Eigenschaften“ haben, die wir „nicht fortnehmen und bessere an ihre Stelle setzen“ können, sondern die wir „hinzuthun oder auf alle Vorstellungen in diesem Gebiet verzichten“ müssen. Eine Darstellung der Bornschen Interpretation von Hertz' Erkenntnisphilosophie sähe daher so aus (Abb. 5):

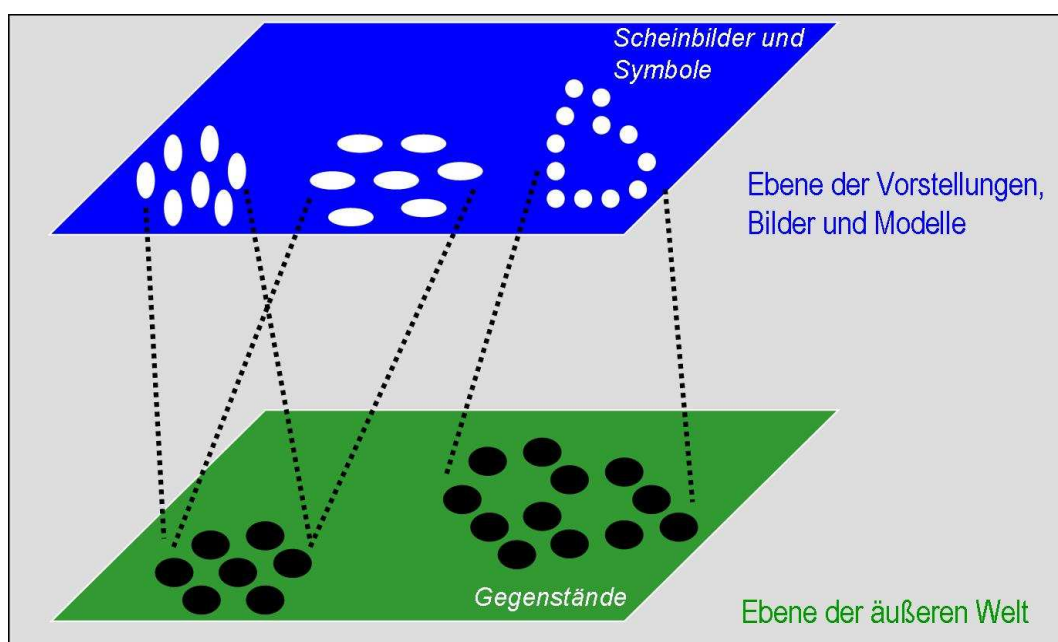


Abb. 5: Gegenstände der äußeren Welt und die Scheinbilder (Symbole, Modelle), die nach Borns Interpretation mathematische Objekte sind.

Mit Aufkommen der Quantenmechanik, der heute grundlegenden Theorie der Physik wurde das „Baukastenspiel“ schwieriger, denn plötzlich gab es verschiedene Klötzchenarten und keine Spielanleitung zum gemeinsamen Gebrauch. Mit den komplementärer „Bildern“ – „Ortsbild“ und „Impulsbild“ – konnten die natürlichen Dinge jeweils nicht vollständig beschrieben werden. Diese klassischen Bilder – Abstraktionen – waren zulässig, aber sie waren nicht mehr richtig. Ein neues noch abstrakteres Bild entstand mit der neuen Theorie „Quantenmechanik“, als die Physiker einen neuen Baukasten fanden, mit neuen „Klötzchen“, den so genannten „Wahrscheinlichkeitsamplituden“, die sich vorzustellen wir „gar nicht erst versuchen sollten“, wie Heisenberg meinte.

Die philosophischen Probleme der Quantenmechanik sollen hier nicht weiter diskutiert werden. Stattdessen soll eine durch Hertz' Erkenntnisphilosophie angestoßene Entwicklung betrachtet werden: ein Aspekt aus dem Werk des österreichischen Philosophen Ludwig Wittgenstein (1889-1951).

4 Ludwig Wittgenstein: Wenn unser Verstand sich Beulen holt ...

“Das Bild ist ein Modell der Wirklichkeit“, “Das Bild ist eine Tatsache“, und “Wir machen uns Bilder von den Tatsachen” schrieb Ludwig Wittgenstein in seinem 1918 fertig gestellten und erstmals 1921 publizierten *Tractatus logico-philosophicus* ([11], 2.1, 2.12, 2.141) (Abb. 6). Mit diesen Sätzen bestätigte er den in seinem Tagebuch erwähnten Einfluss, den Heinrich Hertz mit seinen *Prinzipien der Mechanik* auf Wittgensteins Denken ausgeübt hatte ([12], S. 476). Dass unsere Bilder von den Tatsachen keineswegs eindeutig sein müssen, hatte Hertz schon betont; für die empirische Wissenschaft bedeutet das: denselbe Sachverhalt kann von verschiedenen Theorien dargestellt werden.

Wittgensteins *Tractatus* beginnt mit den beiden Sätzen:

„1. Die Welt ist alles, was der Fall ist.

1.1 Die Welt ist die Gesamtheit der Tatsachen, nicht der Dinge.“ ([11], S. 11)

Es folgen Sätze, die hier erläuternd zusammengefasst sind: Eine Tatsache ist ein bestehender Sachverhalt; ein Sachverhalt ist eine Verbindung von Gegenständen. Die Sprache wird als die Gesamtheit der Sätze aufgefasst. Umgangssprachliche Sätze sind komplexe Sätze. Sie werden durch logische Analyse auf einfache Sätze (Elementarsätze) zurückgeführt. Elementarsätze sind Verkettungen von Namen. Namen sind die einfachsten Zeichen. (Sie können nicht weiter durch andere Zeichen charakterisiert werden.

Die Menschen benützen Sätze, um damit etwas mitzuteilen. Dazu muss ein Satz ein logisches Bild der Wirklichkeit sein. So folgen in Wittgensteins *Tractatus* die Sätze:

„3. Das logische Bild der Tatsachen ist der Gedanke.

4. Der Gedanke ist der sinnvolle Satz.“ ([11], S. 19, 32)

Offenbar hatte sich Wittgenstein zu dieser Theorie durch das Nachstellen einer Unfallszene während einer Gerichtsverhandlung inspirieren lassen, wie einem seiner Tagebucheinträge zu entnehmen ist: „Im Satz wird eine Welt probeweise zusammengestellt. (Wie wenn im Pariser Gerichtssaal ein Automobilunglück mit Puppen etc. dargestellt wird.“ ([12], Tagebücher 1914-1916, 29. 9. 1914.)

Der übliche Sprachgebrauch ist allerdings sehr vage. Diesen Mangel wollte Wittgenstein in seiner frühen, vom *Tractatus* charakterisierten Philosophie, durch die Konstruktion einer Präzisionssprache beseitigen – eine exakte logische Sprache, die in der Lage ist, die Welt genau und eindeutig abzubilden.

In seiner späteren Philosophie hatte sich Wittgenstein von diesem idealen Abbildungscharakter zwischen einer logisch-wissenschaftlichen Sprache und den Gegenständen und Phänomenen der realen Welt abgewandt. In seinen *Erinnerungen an Wittgenstein* beschrieb amerikanische Philosoph Norman Malcolm (1911-1990) der zu Wittgensteins engsten Freunden zählte, eine Schlüsselszene, die zu dieser Wende führte:

„Eines Tages [...] bestand Wittgenstein darauf, daß ein Satz und das, was er darstellt, dieselbe »logische Form«, dieselbe »logische Mannigfaltigkeit« besitzen müssen. Sraffa machte eine Geste, wie sie den Neapolitanern geläufig ist, wenn sie so etwas wie Abscheu oder Verachtung ausdrücken wollen: Er fuhr mit den Fingerspitzen der nach außen gekehrten Hand über die Unterseite des Kinns und fragte: »Was ist die logische Form davon? «

Sraffas Beispiel rief in Wittgenstein das Gefühl hervor, es sei absurd, darauf zu beharren, daß ein Satz und das, was er darstellt, dieselbe »Form« haben müssen. Dadurch löste er sich von der Auffassung, der Satz müsse buchstäblich ein »Bild« der Wirklichkeit sein, die er darstellt. ([13], S. 94)⁴

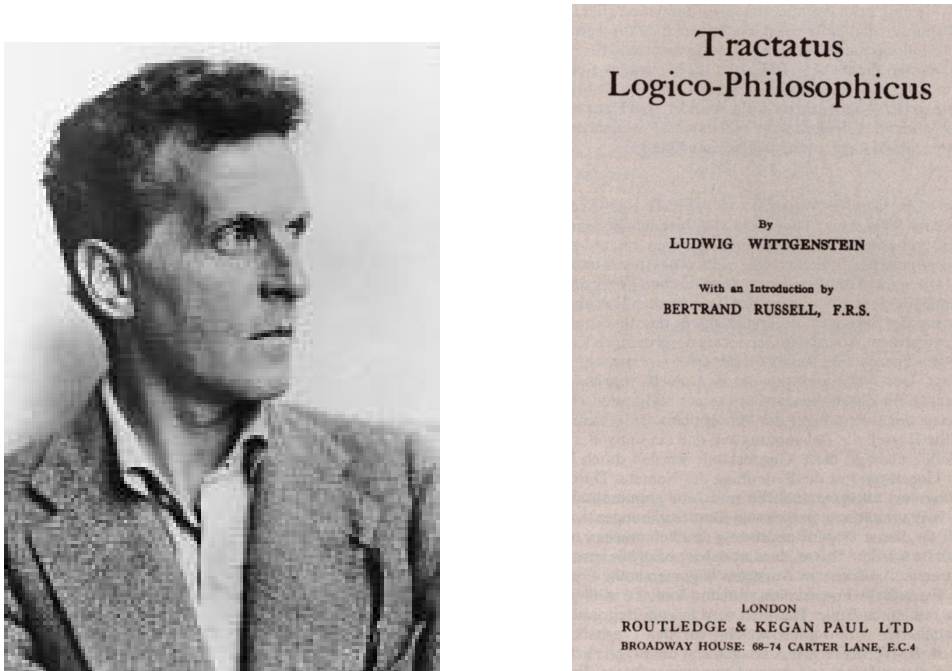


Abb. 6: Ludwig Wittgenstein und sein erstes und einziges zu seinen Lebzeiten publiziertes Hauptwerk.

Wenn es nun nicht gelingt, eine logisch präzise Sprache zu finden, und wir uns also mit unserem vagen Sprachgebrauch abfinden müssen, bleiben die Bilder, Modelle, Theorien, die wir mit den Worten und Sätzen unserer Sprache bilden, um darüber kommunizieren zu können „unscharf“. In Wittgensteins Blauem Buch, das er seinen Freunden, nach seiner Rückkehr aus Österreich nach Cambridge in den 1930-er Jahren diktieren hatte, steht das so: „Wir sind unfähig, die Begriffe, die wir gebrauchen, klar zu umschreiben; nicht weil wir ihre wirkliche Definition nicht wissen, sondern weil sie keine wirkliche »Definition« haben. Die Ausnahme, daß sie eine solche Definition haben müssen, wäre wie die Ausnahme, daß ballspielende Kinder grundsätzlich nach strengen Regeln spielen.“ [13], S. 49)

Bleiben wir beim Begriff „Spiel“! – In dem seine Spätphilosophie charakterisierenden und erst nach seinem Tode veröffentlichten zweiten Hauptwerk Philosophische Untersuchungen schrieb Wittgenstein:

„66. Betrachte z. B. einmal die Vorgänge, die wir ‚Spiele‘ nennen. Ich meine Brettspiele, Kartenspiele, Ballspiele, Kampfspiele, usw. Was ist allen diesen gemeinsam? – Sag nicht: »Es muß ihnen etwas gemeinsam sein, sonst hießen sie nicht ‚Spiele‘« – sondern schau, ob ihnen allen etwas gemeinsam ist. – Denn, wenn du sie anschaust, wirst du zwar nicht etwas sehen, was allen gemeinsam ist, aber du wirst Ähnlichkeiten, Ver-

⁴ Der italienische Wirtschaftswissenschaftler Piero Sraffa (1898-1983) war ein wichtiger Gesprächspartner von Wittgenstein in Cambridge. Auf Vorschlag von John Maynard Keynes (1883-1946) an das King's College nach Cambridge kam, nachdem er Italien verlassen musste. Ein Artikel, den er im *Manchester Guardian* publizierte, hatte Mussolinis Zorn erregt. Sraffa wurde später an das Trinity College berufen.

wandtschaften, sehen, und zwar eine ganze Reihe. Wie gesagt, denk nicht, sondern schau! – Schau z. B. die Brettspiele an, mit ihren mannigfachen Verwandtschaften. Nun geh zu den Kartenspielen über: hier findest du viele Entsprechungen mit jener ersten Klasse, aber viele gemeinsame Züge verschwinden, andere treten auf. Wenn wir nun zu den Ballspielen übergehen, so bleibt manches Gemeinsame erhalten, aber vieles geht verloren. -- Sind sie alle unterhaltend? Vergleiche Schach mit dem Mühlfahren. Oder gibt es überall ein Gewinnen und Verlieren, oder eine Konkurrenz der Spielenden? Denk an die Patienen. In den Ballspielen gibt es Gewinnen und Verlieren; aber wenn ein Kind den Ball an die Wand wirft und wieder auffängt, so ist dieser Zug verschwunden. [...] Und so können wir durch die vielen, vielen anderen Gruppen von Spielen gehen, Ähnlichkeiten auftauchen und verschwinden sehen. Und das Ergebnis dieser Betrachtung lautet nun: Wir sehen ein kompliziertes Netz von Ähnlichkeiten im Großen und Kleinen.

67. Ich kann diese Ähnlichkeiten nicht besser charakterisieren als durch das Wort „Familienähnlichkeiten“; denn so übergreifen und kreuzen sich die verschiedenen Ähnlichkeiten, die zwischen den Gliedern einer Familie bestehen: Wuchs, Gesichtszüge, Augenfarbe, Gang, Temperatur, etc. etc. – Und ich werde sagen: die „Spiele“ bilden eine Familie.“ ([14], §§ 66, 67, S. 56 f.

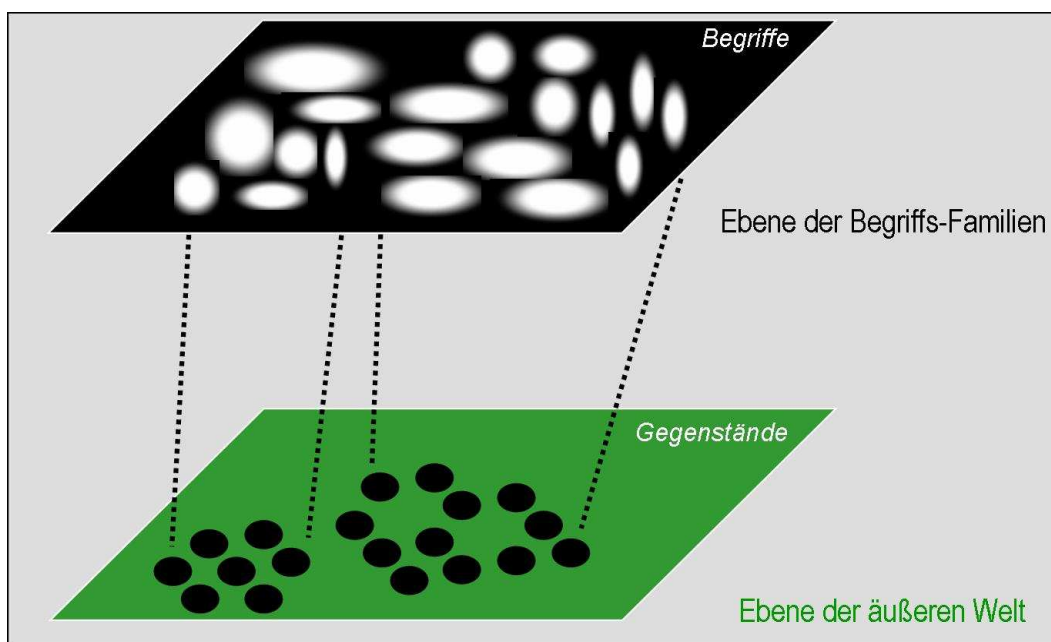


Abb. 7: Gegenstände der äußeren Welt und die sie bezeichnenden und Familien bildenden Begriffe.

In Abb. 7 wurde versucht, diese Verhältnisse zwischen den Gegenständen und ihren Begriffen, sowie deren Familien darzustellen. Die Begriffe haben keine scharfen Grenzen. Sie entsprechen den Scheinbildern bzw. Vorstellungen bei Hertz, doch eine darüber liegende Schicht der „Kerne“ dieser Vorstellungen, die keine der „unwesentlichen Eigenschaften“ mehr haben, und die in Borns Interpretation der Hertzschen Philosophie noch darüber liegt, existiert hier nicht. Wittgensteins frühes philosophisches System des *Tractatus* hätte eine solche Ebene für die Worte (Begriffe) der exakten logischen Sprache vorgesehen, doch in seiner späten Philosophie ging er davon aus, dass es eine solche Sprache nicht geben kann und die Begriffe unserer Sprache keine eindeutigen Abbilder der äußeren Gegenstände sind. Die Voraussetzung solcher eindeutiger Abbildungen führte zu keinen nützlichen Resultaten; Paragraph 119 seiner *Philosophischen Untersuchungen* lautet:

„Die Ergebnisse der Philosophie sind die Entdeckung irgendeines schlichten Unsinn und Beulen, die sich der Verstand beim Anrennen an die Grenzen der Sprache geholt hat. Sie, die Beulen, lassen uns den Wert jener Entdeckung erkennen.“ ([14], § 119, S. 83)

Das philosophisch traditionsreiche „Bild“-Konzept ist offenbar unscharf geworden. Im letzten Abschnitt soll es gerade deswegen zum der Ausgangspunkt für eine Interpretation des Zusammenhanges zwischen realen Systemen und theoretischen Strukturen moderner Wissenschaften dienen. Dazu wird die innerhalb der Wissenschaftstheorie des 20. Jahrhunderts „strukturalistisch“ genannte Theorienauffassung“ durch die Konzepte der Fuzzy Set Theorie bereichert.

5 Hertz' Bilder, Wittgensteins beulen, Fuzzy Sets

Die zwischen der Realität und den exakt-mathematischen Theorien bestehende Kluft und die Suche nach Möglichkeiten, sie zu überbrücken, motivierten in den 1960-er Jahren den Elektrotechniker Lotfi A. Zadeh eine „mathematics of cloudy or fuzzy quantities“ zu erwägen, und sie schließlich als Theorie der *Fuzzy Sets and Systems* zu etablieren. Mittels dieser Theorie wurde in der zweiten Hälfte des 20. Jahrhunderts eine Fülle von wissenschaftlichen Begriffen, Methoden und Theorien bietet „fuzzifiziert“ – eine Transformation, die wissenschaftstheoretisch nachvollzogen und reflektiert werden kann, indem das Gerüst der *strukturalistischen Theorienauffassung* entsprechend ergänzt wird: Fuzzy Sets können dann als neues Modellierungswerkzeug in der Wissenschaftstheorie fungieren. (Siehe [16].)

Mit Fuzzy Sets können unscharf begrenzte Mengen und Strukturen erfasst werden. Sie können dazu dienen, die scharfen Begrenzungen unserer Begriffe zu auflösen, die der deutsche Mathematiker und Philosoph Gottlob Frege (1848-1925) stets verlangt hatte, denn „ohne vollständige und endgültige Definitionen hat man keinen festen Boden unter den Füßen, ist man der Geltung seiner Lehrsätze nicht sicher, kann man nicht zuversichtlich die logischen Gesetze anwenden, die ja die scharfe Begrenzung der Begriffe und also auch der Beziehungen zur Voraussetzung haben.“ ([15], S. 75)

Zadehs Theorie der Fuzzy Sets und Fuzzy Logik, das „Computing with Words“ und die „Computational Theory of Perceptions“ sind zudem aufeinander aufsetzende Methodologien, die zwischen die Ebene der realen Systeme und die der theoretischen Strukturen passen, um die menschlichen Wahrnehmungen (perceptions) zu repräsentieren, die mit den scharfen Begriffen der klassischen Logik nicht dargestellt werden können. In meinem Beitrag zum 16. Workshop „Computational Intelligence“ habe ich dies ausführlich dargestellt [16] und dabei die folgende Abb. 8 benutzt. Die hier zuunterst gezeichnete „Empirische Ebene“ entspricht der Ebene der äußeren Welt in Abb. 4, die „Theoretische Ebene“ in Abb. 8 der dortigen „Ebene der abstrakten Kerne der Vorstellungen“, und die Fuzzy-Ebene in Abb. 8 soll der „Ebene der Vorstellungen“ in Abb. 4 analog aufgefasst werden. Dies bedeutet, dass die „Scheinbilder“ oder Vorstellungen im Sinne der Hertzschen Erkenntnisphilosophie als Fuzzy Sets interpretiert werden können, während die „Kerne“ dieser Vorstellungen die scharfen Mengen sind, die nach der Defuzzifizierung dieser Fuzzy Sets zurück bleiben. Die Erkenntnisphilosophie von Heinrich Hertz und in deren Folge auch die Spätphilosophie von Ludwig Wittgenstein lassen sich auf diese Weise mit Hilfe Fuzzy Sets modellieren, während die davon abweichende Bornsche Interpretation der Hertzschen Theorie von auf die übliche Modellierung durch die gewöhnliche klassische Mathematik führt.

Diese beiden Modellierungswiesen entsprechen der in [16] vorgestellten strukturalistischen Theorienauffassung der modernen Wissenschaftstheorie bzw. seiner Erweiterung um Elemente der *Fuzzy Set Theorie*.

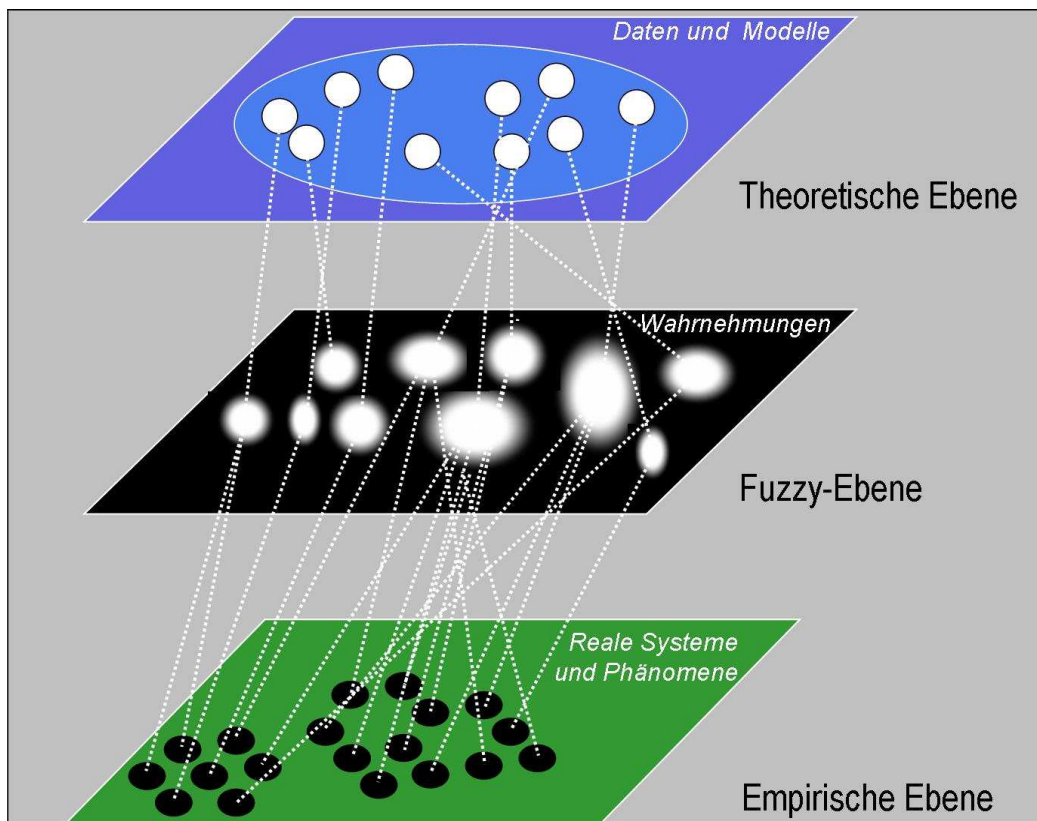


Abb. 8: Die drei Ebenen des Fuzzy-Strukturalismus und die Übergänge der Fuzzifizierung zwischen der Empirischen und der Fuzzy-Ebene bzw. der Defuzzifizierung zwischen der Fuzzy-Ebene und der Theoretischen Ebene aus [16].

6 Literatur

- [1] Sanchis-Segure, Carles und Spanagel, Rainer: Behavioural assessment of drugreinforcement and addictive features in rodents: an overview. *Addiction Biology*, 11, 2006, S. 2-38.
- [2] Die Comics sind unter <http://www.zi-mannheim.de/psychopharmakologie.html> abrufbar.
- [3] Pressemitteilung: Dr. med. Marina Martini, M. Sc., Referat Öffentlichkeitsarbeit Zentralinstitut für Seelische Gesundheit J 5, Mannheim http://www.innovationsreport.de/html/berichte/bildung_wissenschaft/bericht-66587.html
- [4] Siehe dazu die Homepage des Zentralinstituts für Seelische Gesundheit in Mannheim: <http://www.zi-mannheim.de/psychopharmakologie.html>
- [5] Aus: Andreas Cellarius: *Harmonia Macrocosmica*, 1660/61. Künstler: Loon, Johannes van: *Scenographia systematis mvndani Ptolemaici* [cartographic material]. Amsterdam 1660. National Library of Australia, MAP NK 10241.
- [6] Aus: Andreas Cellarius: *Harmonia Macrocosmica*, 1708.
- [7] Heinrich Hertz: *Die Constitution der Materie*. Eine Vorlesung über die Grundlagen der Physik aus dem Jahre 1884. Herausgegeben von Albrecht Fölsing. Berlin, Heidelberg [u.a.]: Springer 1999.

- [8] Albrecht Fölsing: *Heinrich Hertz. Eine Biographie*. Hamburg: Hoffman und Campe 1997.
- [9] Max Born: Geleitwort zu W. R. Fuchs: *Exakte Geheimnisse*. München, Zürich 1965. Zitiert nach Anm. 35 von J. Kuczera in [10].
- [10] Heinrich Hertz: *Die Prinzipien der Mechanik in neuen Zusammenhänge dargestellt*. Drei Beiträge. (Ostwalds Klassiker der exakten Wissenschaften Bd. 263), Frankfurt am Main: Verlag Harri Deutsch 1996, 2. Auflage 2002).
- [11] Ludwig Wittgenstein: *Tractatus logico-philosophicus, Logisch-philosophische Abhandlung*. Edition Suhrkamp Nr. 12, Frankfurt am Main 1963, 2003.
- [12] Ludwig Wittgenstein: Tagebucheintrag in: *Werkausgabe* in acht Bänden. Frankfurt am Main: Suhrkamp, Bd. 8, 1984 ff.
- [13] Ludwig Wittgenstein: *Das Blaue Buch*. In: *Werkausgabe* in acht Bänden. Frankfurt am Main: Suhrkamp, Bd. 5, 1984 ff.
- [14] Ludwig Wittgenstein: *Philosophische Untersuchungen*. Auf der Grundlage der kritisch-genetische Edition neu herausgegeben von Joachim Schulte. Mit einem Nachwort des Herausgebers. Frankfurt am Main: Suhrkamp Verlag 2003.
- [15] Gottlob Frege: *Grundgesetze der Arithmetik*, Bd. II Jena 1903, S. 75.
- [16] Rudolf Seising: *Fuzzy Sets als Werkzeug der Wissenschaftstheorie*. In: R. Mikut, M. Reischl (Hrsg.), Schriftenreihe des Instituts für Angewandte Informatik / Automatisierungstechnik Universität Karlsruhe (TH) Band 15: Proceedings 16. Workshop „Computational Intelligence“ des GMA-FA 5.14 „Computational Intelligence“ (Fachausschuss der VDI/VDE Gesellschaft für Mess- und Automatisierungstechnik) und der GI-FG „Fuzzy-Systeme und Soft-Computing“ (Fachgruppe der Gesellschaft für Informatik) 29. November – 1. Dezember 2006, Universitätsverlag Karlsruhe 2006, S. 262-276.

Sequenzvergleiche im Fuzzy-Hypercube

Julia Limberg und Rudolf Seising

Medizinische Statistik und Informatik, Medizinische Universität Wien
Spitalgasse 23, A-1090 Wien, Österreich
Tel. (+43) 1 404006664
Fax (+43) 1 404006667
E-Mail: rudolf.seising@meduniwien.ac.at

1 Einführung

In einem Vortrag, den Lotfi A. Zadeh (Abb. 1, links) im Frühjahr 1965 auf dem *Symposium on Systems Theory* am Polytechnic Institute in Brooklyn über “A New View on System Theory” hielt, sprach er erstmals vor größerem Publikum über „Fuzzy Sets and Systems“, und unter dieser Überschrift wurde sein Beitrag auch im Tagungsband abgedruckt (Abb. 1, rechts). Er schlug damals die Fuzzy Sets als mathematisches Werkzeug der Systemtheorie vor, denn diese Konzeption “provide a way of treating fuzziness in a quantitative manner” ([2], p. 29). Zadeh definierte damals: “A system S is a fuzzy system if (input) $u(t)$, output $y(t)$, or state $x(t)$ of S or any combination of them ranges over fuzzy sets. ([1], p. 33) Damals erklärte er auch, dass “these concepts relate to situations in which the source of imprecision is not a random variable or a stochastic process but rather a class or classes which do not possess sharply defined boundaries.” ([2], p. 29)

Zu dieser Zeit wurde der Artikel “Fuzzy Sets” [3] bereits bei der Zeitschrift *Information and Control* begutachtet, im oben genannten Vortrag übernahm Zadeh allerdings zentrale Aussagen über seinen dort eingeführten neuen “way of dealing with classes in which there may be intermediate grades of membership.”¹

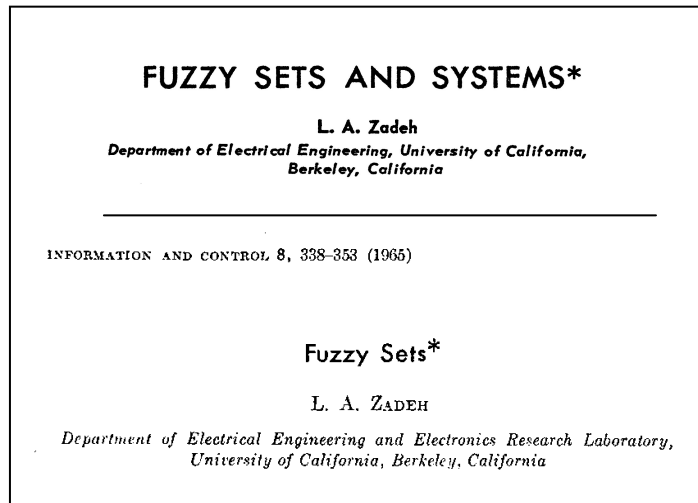
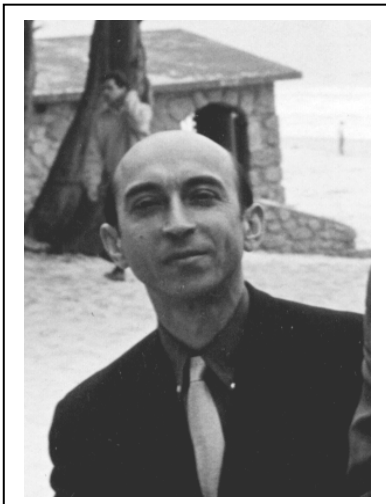


Abb. 1: Links: Lotfi A. Zadeh in den 1960-er Jahren. Rechts, oben: Überschrift der Druckversion seines Vortrags “A New View on System Theory” [2], unten: Überschrift des Zeitschriftenartikels „Fuzzy Sets“ [3].

¹ Für eine ausführliche historische Betrachtung der Geschichte der Fuzzy Set Theorie siehe [1].

Den Begriff des Fuzzy Sets führte er folgendermaßen ein: “a class in which there may be a continuous infinity of grades of membership, with the grade of membership of an object x in a fuzzy set A represented by a number $\mu_A(x)$ in the interval $[0,1]$.” ([2], S. 29.)

Zadeh betonte, dass mit diesem Konzept ein oft benötigter Abstraktionsprozeß geeignet dargestellt werden könne: a “convenient way of defining abstraction – a process which plays a basic role in human thinking and communication.” Um die Eigenschaften der Mengen in der gewöhnlichen Mengentheorie in die Theorie der Fuzzy Sets übertragen zu können, definierte Zadeh Gleichheit (*equality*), Enthaltensein (*containment*), Komplement (*complementation*), Durchschnitt (*intersection*) und Vereinigung (*union*) für fuzzy sets A, B in einer beliebigen Grundmenge X auf folgende Weise (für alle $x \in X$):

- $A = B$ gilt genau dann, wenn gilt: $\mu_A(x) = \mu_B(x)$,
- $A \subseteq B$ genau dann, wenn gilt: $\mu_A(x) \leq \mu_B(x)$,
- $\neg A$ is the complement of A genau dann, wenn gilt: $\mu_{\neg A}(x) = 1 - \mu_A(x)$,
- $A \cup B$ genau dann, wenn gilt: $\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$,
- $A \cap B$ genau dann, wenn gilt: $\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$.

In den späten 1960-er Jahren entwickelte sich dann ein “Fuzzifizierungsprogramm”, als Zadeh und andere Autoren zahlreiche mathematische Begriffe und Theorien „fuzzifizierten“; damals entstanden z. B. die Fuzzy-Topologie [4], die Fuzzy-Wahrscheinlichkeitstheorie [5] und Fuzzy-Algorithmen [6].

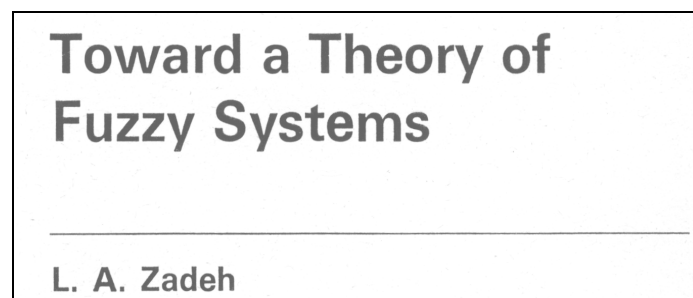


Abb. 2: Überschrift des Buchkapitels von Zadeh aus dem Jahre 1972 (geschrieben 1969) [7].

Zur Fuzzifizierung von Algorithmen erklärte Zadeh in seinem Buchbeitrag “Toward a Theory of Fuzzy Systems” [7], (Abb. 2): “Roughly speaking, a fuzzy algorithm is an algorithm in which some of the instructions are fuzzy in nature. Examples of such instructions are:

- (a) increase x slightly if y is slightly larger than 10;
- (b) decrease u until it becomes much smaller than v ;
- (c) reduce speed if the road is *slippery*.”

Allgemein ausgedrückt plädierte Zadeh dafür, einen Fuzzy-Algorithmus als ein Fuzzy-System A aufzufassen, das durch folgende Gleichungen charakterisiert ist:

$$X^{t+1} = F(X^t, U^t), \quad U^t = H(X), \quad (1)$$

Dabei ist X^t ein Fuzzy-Zustand (*fuzzy state*) des Fuzzy-Systems A zum Zeitpunkt t , U^t ist ein Fuzzy-Input (*fuzzy input*) zum Zeitpunkt t (der eine Fuzzy-Instruktion darstellt), und X^{t+1} ist der Fuzzy-Zustand zum Zeitpunkt $t+1$, der sich nach der Ausführung der durch U^t dargestellten Fuzzy-Instruktion für das Fuzzy-System A ergibt. Die Funktion F definiert die Abhängigkeit des Fuzzy-Zustandes zum Zeitpunkt $t+1$ vom Fuzzy-Zustand zum Zeitpunkt t und vom Fuzzy-Input zum Zeitpunkt t , während die Funktion H die Abhängigkeit des Fuzzy-Inputs zum Zeitpunkt t vom Fuzzy-Zustand zum Zeitpunkt t beschreibt. Nach diesen Definitionen zur Einführung einer Theorie der Fuzzy-Systems betrachtet Zadeh ein sehr einfaches Beispiel zur Illustration, das später allerdings zu einer völlig neuen Anschauung der Fuzzy Sets führen sollte:

“Suppose that X^t is a fuzzy subset of a finite set $X = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ and U^t is a fuzzy subset of a finite set $U = \{\beta_1, \beta_2\}$. Since the membership functions of X^t and U^t are mappings from, respectively, X and U to the unit interval, these functions can be represented as points in unit hypercubes R^4 and R^2 , which we shall denote for convenience by C^4 and C^2 . Thus, F may be defined by a mapping from $C^4 \times C^2$ to C^4 and H by a mapping from C^4 to C^2 . For example, if the membership function of X^t is represented by the vector $(0.5, 0.8, 1, 0.6)$ and that of U^t by the vector $(1, 0.2)$, then the membership function of X^{t+1} would be defined by F as a vector—say $(0.2, 1, 0.8, 0.4)$ —whereas that of U^t would be defined by H as a vector $(0.3, 1)$, say.“ ([7], S. 485f)

Es gibt in Zadehs Werk aus den ersten Jahrzehnten der Fuzzy Set Theorie unseres Wissens keine weitere Bemerkung zur Darstellung der Zugehörigkeitsfunktionen von Fuzzy Sets als Punkte im Hyperwürfel! Dieser Zusammenhang wurde erst mehr als 15 Jahre später wieder aufgegriffen und bekannt gemacht, und zwar durch die Arbeiten von Bart Kosko (Abb. 4), einem damaligen Doktoranden in der elektrotechnischen Fakultät an der University of California at Irvine, der in den 1980-er Jahren bei Zadeh in Berkeley die Fuzzy Sets kennen lernte. Kosko promovierte 1987 mit der Dissertation *Foundation of Fuzzy Estimation Theory* [8]. Zuvor hatte er in mehreren Zeitschriftenartikeln Ergebnisse seiner Grundlagenarbeiten vorgestellt: In „Counting with Fuzzy Sets“ [9] verallgemeinerte er das Maß der Kardinalität für gewöhnliche Mengen (Anzahl der Elemente einer Menge) auf das der Kardinalität von Fuzzy Sets (Summe der Zugehörigkeitsfunktionswerte aller Elemente ihres Definitionsbereichs) – eine Größe, die von Zadeh schon 1983 als so genannter *sigma count* eingeführt wurde [10]. In „Fuzzy Entropy and Conditioning“ [11] führte er ein so genanntes Maß der „Fuzzy Entropie“ ein, das die Fuzzi-ness von Fuzzy Sets angibt, ohne die Methoden der Wahrscheinlichkeitstheorie zu nutzen. Er definierte dieses Entropiemaß als rationalen Bruch der zwei Abstände, den das jeweilige Fuzzy Set zur nächst bzw. weitest entfernten gewöhnlichen (also nicht fuzzy) Menge einnimmt. Diese Abstände entsprechen den Distanzen zwischen Vektoren in einem Raum, deren Koordinaten die Zugehörigkeitswerte der entsprechenden Fuzzy Sets sind. Ausgehend von Shannons Entropiebegriff, interpretierte Kosko die Fuzzy Sets in dieser Arbeit als Fuzzy-Nachrichten (*fuzzy messages*) innerhalb eines neuen Gerüsts für eine Fuzzy-Informationstheorie (*fuzzy information theory*). [11] Möglicherweise verwies Kosko aufgrund der informationstheoretischen Ausrichtung dieser Texte nicht auf den deutlichen Zusammenhang mit der Darstellung von Fuzzy Sets im Hyperwürfel. Vielleicht erkannte er diese Darstellungsmöglichkeit auch erst etwas später, denn erst 1990 führte er die Ergebnisse der genannten Arbeiten in seinem Artikel „Fuzziness vs. Probability“ [12], (Abb. 3) zur „Geometrie der Fuzzy Sets“ im Hyperwürfel zusammen:

“It helps to see the geometry of fuzzy sets when discussing fuzziness. To date this visual property has been overlooked. The emphasis has instead been on interpreting fuzzy sets as membership functions, mappings μ_A from domain X to range $[0, 1]$. But functions are hard to visualize. Membership functions are often pictured as two-dimensional graphs, with the domain X misleadingly represented as one-dimensional. The geometry of fuzzy sets involves both the domain $X = \{x_1, \dots, x_n\}$ and the range $[0, 1]$ of mappings $\mu_A: X \rightarrow [0, 1]$. The geometry of fuzzy sets is a great aid in understanding fuzziness, defining fuzzy concepts, and proving fuzzy theorems. Visualizing this geometry may by itself be the most powerful argument for fuzziness.

The geometry of fuzzy sets is revealed by asking an odd question: What does the fuzzy power set $F(2^X)$, the set of all fuzzy subsets of X , look like? Answer: A cube. What does a fuzzy set look like? A point in a cube. The set of all fuzzy subsets is the unit hypercube $I^n = [0, 1]^n$. A fuzzy set is a point in the cube I^n .” ([12], S. 216)

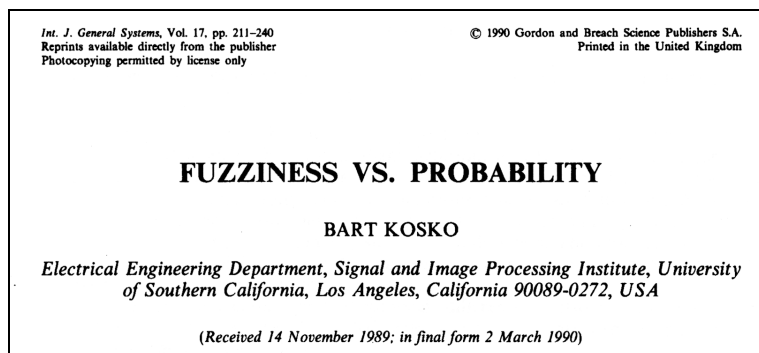


Abb. 3: Überschrift des Artikels von Kosko im Jahre 1990 [12].

2 Der Fuzzy Hyperwürfel

In diesem Abschnitt wollen wir in aller gebotenen Kürze die Grundlagen der „Geometrie der Fuzzy Sets“ von Kosko zusammenfassend darstellen. Dabei orientieren wir uns an seinem Artikel “Fuzziness vs. Probability” [12].

Kosko setzte gegen Zadehs “sets-as-functions definition of fuzzy sets” einen „sets-as-points“-Definition im „Fuzzy Hyperwürfels“ ([12], S. 216). Er begründete dies damit, dass die Interpretation der “fuzzy sets as membership functions, mappings μ_A from domain X to range $[0, 1]$ ” sehr schwierig zu visualisieren sei: Zugehörigkeitsfunktionen würden oft als zweidimensionale Graphen dargestellt, wobei der Definitionsbereich X irreführenderweise dann als eindimensional erscheine. Kosko betonte, dass seine „Geometrie der Fuzzy Sets” als Punkte im Hyperwürfel sowohl den Definitionsbereich $X = \{x_1, \dots, x_n\}$ als auch den Wertebereich $[0, 1]$ der Abbildungen $\mu_A: X \rightarrow [0, 1]$ umfasst. Seine „Geometrie der Fuzzy Sets” sei daher eine große Hilfe für das Verständnis der Fuzziness, die Definition der Fuzzy Sets und die Beweise von Theoremen der Theorie von Fuzzy Sets und Fuzzy Systemen. Die Visualisierung dieser „Geometrie“ selbst sei eines der mächtigsten Argumente für die Fuzziness.

Die Geometrie der Fuzzy Sets offenbare sich mit der ungewöhnlichen Frage Wie sieht die Fuzzy-Potenzmenge $F(2^X)$ aus, also die Menge aller Fuzzy-Teilmengen von X ? –

Die Antwort ist: $F(2^X)$ ist ein Würfel. Fragen wir weiter: Wie sieht ein Fuzzy Set aus? – Die Antwort ist: Es ist ein Punkt in einem Würfel!

Die Menge aller Fuzzy-Teilmengen ist der Einheits-Hyperwürfel $I^n = [0,1]^n$. Ein Fuzzy Set ist ein Punkt in diesem Würfel I^n . ([10], p. 216) Zur Illustration dieser Sachverhalte siehe die Abbildungen 4 (rechts) und 5.

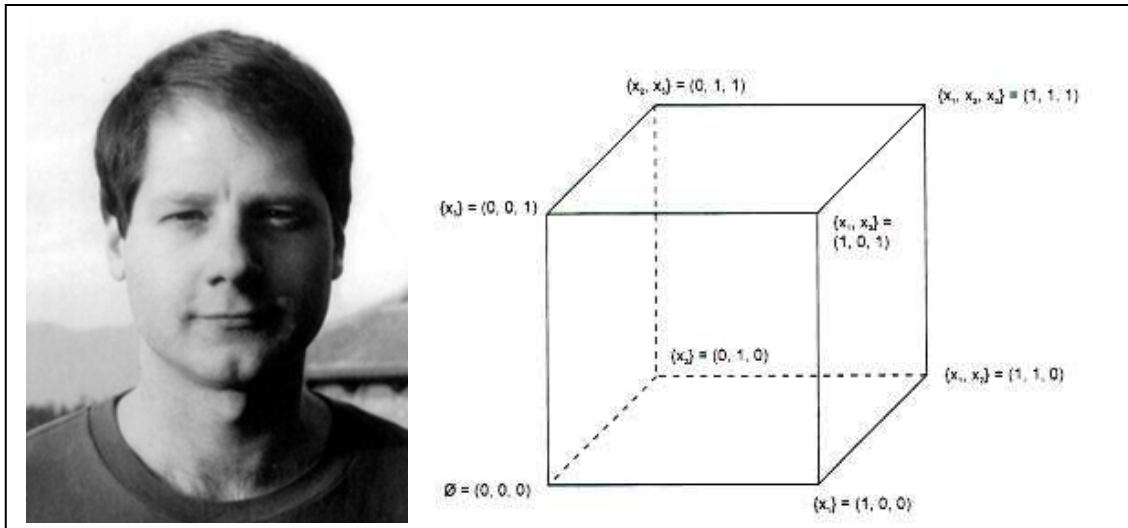


Abb. 4: Links: Bart Kosko in den 1990-er Jahren. Rechts: Der dreidimensionale Würfel I^3 .

Die Ecken des Würfels I^n sind gewöhnliche Mengen, folglich ist die gewöhnliche Potenzmenge 2^X , die Menge aller 2^n gewöhnlichen (also nicht fuzzy) Teilmengen von X , identisch mit dem Booleschen n -Würfel B^n : $2^X = B^n$. Die Fuzzy Sets füllen somit den Verband B^n zum ganzen Würfel aus: $I^n: F(2^X) = I^n$. Dementsprechend kann ein Fuzzy Set $A = \{(x_1, a_1), \dots, (x_n, a_n)\}$ durch den n -dimensionalen Vektor (a_1, \dots, a_n) dargestellt werden, und alle Koordinaten a_i sind Elemente in $[0, 1]$. Konsequenterweise ist A ein Punkt im n -dimensionalen Einheits-Hyperwürfel $[0, 1]^n$.

Wir betrachten das Beispiel einer Menge mit nur zwei Elementen: $X = \{x_1, x_2\}$. Die Potenzmenge 2^X aller Teilmengen von X ist $\{\emptyset, \{x_1\}, \{x_2\}, \{x_1, x_2\}\}$. Diese vier Mengen korrespondieren den Vektoren (bits): $(0,0)$, $(1,1)$, $(1,0)$, $(0,1)$. Die Einsen und Nullen bezeichnen dabei die An- bzw. Abwesenheit des i -ten Elements x_i in der Teilmenge. Abstrakter heißt das: Jede Teilmenge A ist eindeutig durch eine der zweiwertigen Zugehörigkeitsfunktionen $\mu_A: X \rightarrow [0,1]$ definiert.

Betrachten wir nun die Fuzzy-Teilmengen von X . – Eine Fuzzy-Teilmenge von X ist beispielsweise $A = \{0.5, 0.4, 0.7\}$. – Sie kann als eine der unendlich vielen ∞ -wertigen Zugehörigkeitsfunktionen $\mu_A: X \rightarrow [0,1]$ aufgefasst werden, und diese Sichtweise entspricht in der Tat Zadehs „sets-as-functions-Definition“ von Fuzzy Sets.

In diesem Beispiel gehört Element x_1 zum Grad 0.5, Element x_2 zum Grad 0.4 und Element x_3 zum Grad 0.7 zur Teilmenge A und entsprechend der fuzzy-geometrischen „sets-as-points“-Ansicht ist die Fuzzy-Teilmenge A der Punkt $\{0.5, 0.4, 0.7\}$ im Einheitswürfel I^3 , siehe Abbildung 5 (links).

Zur leichteren Zeichnung betrachten wir in diesem Abschnitt nur noch zweidimensionale Beispiele. Bei der Fuzzy-Teilmenge $A = \{1/3, 1/3\}$ von X gehören Elemente x_1 und x_2 jeweils zum Grad $1/3$ zur Teilmenge A . Die Fuzzy-Teilmenge A ist also der Punkt $\{1/3, 1/3\}$ im Einheitswürfel I^2 , siehe Abbildung 5, (rechts).

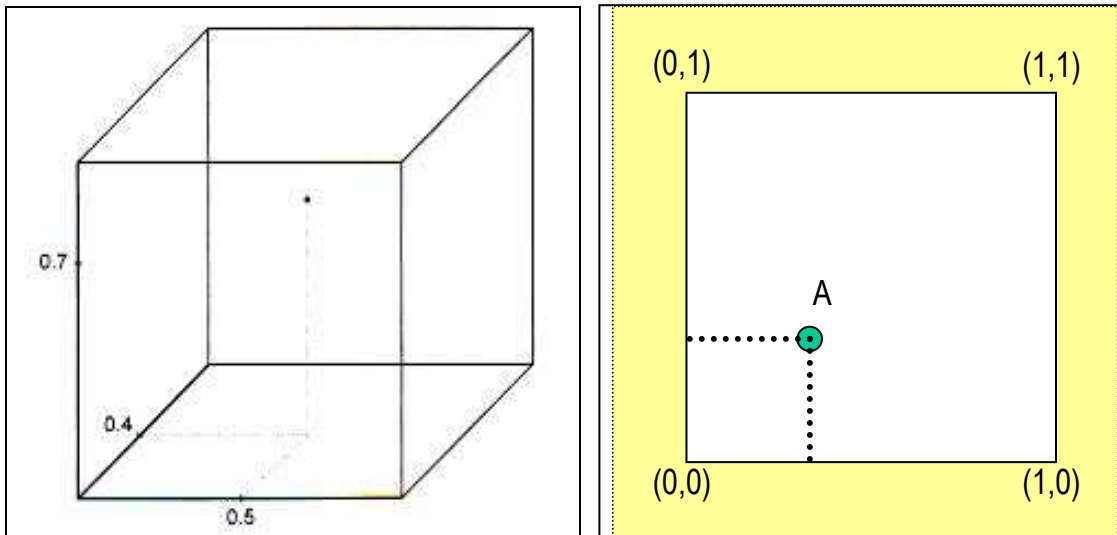


Abb. 5: Links: Punkt (0.5, 0.4, 0.7) im Würfel \mathcal{P} . Rechts: Punkt (1/3, 1/3) im Würfel \mathcal{P} .

In Abbildung 6 (links) ist der zweidimensionale Einheitswürfel mit den Punkten für das Fuzzy Set $A = \{1/4, 2/3\}$ sein Komplement $A^c = \{1-1/4, 1-2/3\} = \{3/4, 1/3\}$ und Mittelpunkt $M = \{1/2, 1/2\}$ des Einheitswürfels dargestellt. Man erkennt leicht, dass der Punkt für das Komplement eines Fuzzy Sets durch Spiegelung am Mittelpunkt erhalten wird. Abbildung 6 (rechts) zeigt neben den Punkten für die Fuzzy Sets $A = \{1/4, 2/3\}$ und dessen Komplement $A^c = \{3/4, 1/3\}$ auch die für Vereinigung und Durchschnitt dieser beiden Fuzzy Sets. Es ist offensichtlich, dass diese vier Punkte ein Rechteck im Einheitswürfel bilden.² Kosko konstatierte: „The set-as-point view shows that these four points in the unit square hang together, indeed move together, in a very natural way.“ ([12], S. 218)

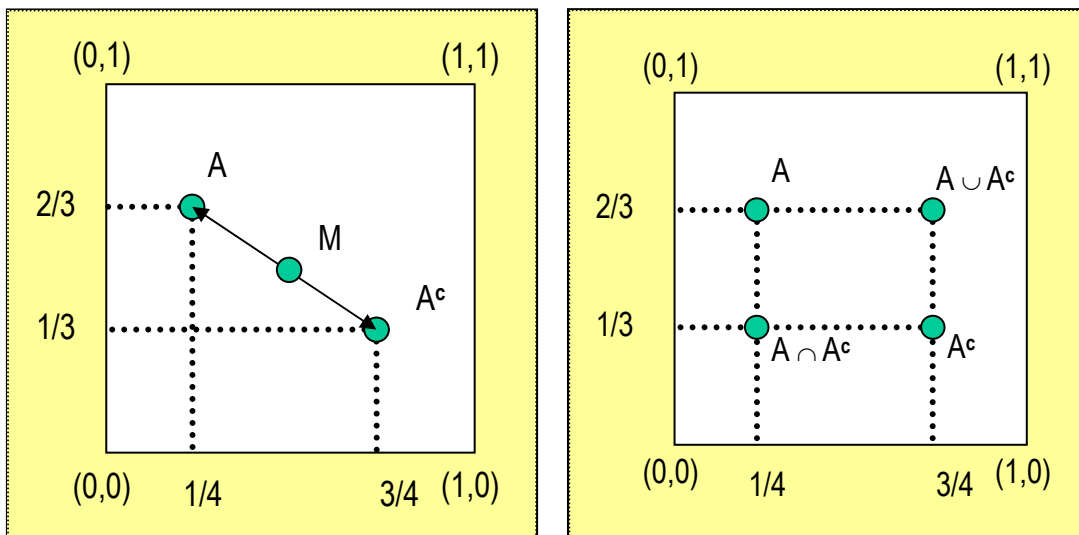


Abb. 6: Links: Punkte A, A^c und der Mittelpunkt im Einheitswürfel \mathcal{P} .
Rechts: A, A^c sowie die Punkte für die Fuzzy Sets $A \cap A^c$ und $A \cup A^c$ im Einheitswürfel \mathcal{P} .

² Dies und die folgenden Gleichungen bzw. Verhältnisse im Einheitswürfel gelten nur mit Zadehs ursprünglichen Definition des Durchschnitts (der Vereinigung) von Fuzzy Sets als Minimum (Maximum) der Zugehörigkeitsfunktionen. Für andere t-Normen und t-Conormen stimmt dies nicht. Wir danken Prof. Dr. Erich Peter Klement für diesen Hinweis.

Der Mittelpunkt ist für alle Einheitswürfel ($n \in \mathbb{N}$) eindeutig und es gilt: „The midpoint of the cube I^n is maximally fuzzy.“ Kosko erläutert: “All its membership values are 1/2. The midpoint is unique in two respects. First, the midpoint is the only set A that not only equals its own opposite A^c but equals its own overlap and underlap as well ... Second, the midpoint is the only point in the cube I^n that is equidistant to each of the 2^n vertices of the cube. The nearest corners are also the farthest.” ([12], S. 216 f)

Das bedeutet, dass 1) folgende Gleichungsreihe gilt:

$$M = M^c = M \cap M^c = M \cup M^c, \quad (2)$$

und 2) dass der Mittelpunkt (siehe Abb. 6 (links) und 7) von allen vier Eckpunkten des Einheitswürfels gleich weit entfernt ist.

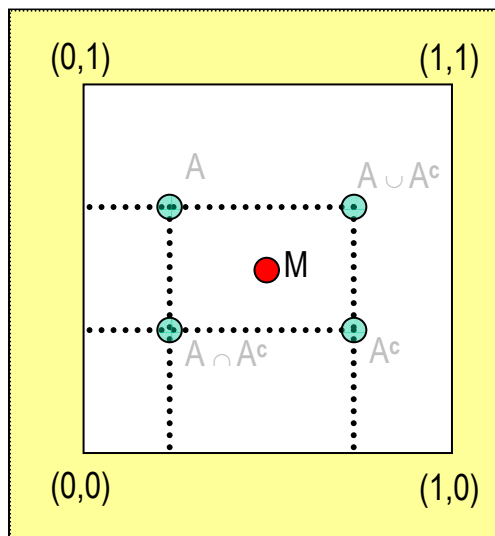


Abb. 7: Der Mittelpunkt im Einheitswürfel I^2 .

In den beiden folgenden Abschnitten werden Anwendungen von Koskos „Geometrie der Fuzzy Sets“ als Punkte im Hyperwürfel auf zwei Probleme der Lebenswissenschaften thematisiert:

- Eines der wichtigsten Probleme in der Medizin ist die Diagnose von Krankheiten aufgrund der Symptome und Zeichen beim Patienten (siehe dazu [13-15] und Kapitel 7 in [1]). Infolge des hohen Wachstums des medizinischen Wissens arbeitet man seit Jahrzehnten an einer einheitlichen Klassifikation von Krankheitsbegriffen doch eine einzige verbindliche Terminologie der Krankheitsbegriffe gibt es nicht.³ In der Medizin beschäftigt man sich in der *Nosologie* (griechisch: *nosos* – Krankheit) mit dieser Problematik. Krankheiten können durch Aufzählung aller Symptome und Zeichen dargestellt werden, die für sie mit einem gewissen Grad hinweisend sind. Somit besteht die Möglichkeit sie als Sequenzen

³ 1893 wurde die von Jacques Bertillon (1851-1922), dem Leiter des Pariser Statistischen Amtes im Auftrag des Internationalen Statistischen Instituts aufgestellte „Todesursachenklassifikation“ zur Grundlage des International *Statistical Classification of Diseases and Related Health Problems* (ICD) der Weltgesundheitsorganisation (WHO). Diese erweiterte die Systematik 1948 in der 6. Revision auf Krankheiten und Verletzungen. Aufgrund der medizinischen Fortschritte wurden immer wieder Änderungen und Ergänzungen notwendig, daher gab es bis 1976 im Abstand von ca. 10 Jahren Revisionen bis zu ICD-9. Die 1992 verabschiedete ICD-10 wird von der WHO jährlich aktualisiert. Seit 1986 ist ICD-9 in Deutschland verpflichtend zur Diagnosenverschlüsselung in Krankenhäusern eingesetzt.

von bewerteten Symptomen und Zeichen – als Fuzzy Sets, also als Punkte im Hyperwürfel zu repräsentieren.

- Seit der Entschlüsselung des genetischen Codes in den 1960-er Jahren untersucht und behandelt man in der Genetik die aus Molekülsequenzen bestehenden Nucleinsäuren DNA (Desoxyribonucleinsäure) und RNA (Ribonucleinsäure). In der Sequenzanalyse werden die Grundbestandteile und die Ordnung des genetischen Materials untersucht, mittels Sequenzvergleichen versucht man strukturelle Beziehungen (Identität, Differenz, Ähnlichkeit) von Nucleinsäureketten zu solchen zu erkennen die bereits analysiert und bekannt sind. Solche Sequenzen lassen sich offensichtlich als Fuzzy Sets, als Punkte im Hyperwürfel darstellen.

3 Fuzzy-Gesundheit

Der an der Universität in Münster lehrende iranisch-deutsche Mediziner und Medizinteoretiker Kazem Sadegh-Zadeh (Abb. 7) diskutiert seit Jahrzehnten die Begriffe „Gesundheit“ und „Krankheit“ [16, 17], die er 1980 schließlich fuzzifizierte, nachdem er sich mit der damals schon 15 Jahre alten Theorie seines früheren Landsmannes Zadeh beschäftigte, und ein erstes Gerüst für eine neue Medizinteorie publizierte: “health is a matter of degree, illness is a matter of degree, and disease is a matter of degree” [18].

Schon zu Beginn der 1980-er Jahre hatte Sadegh-Zadeh den Begriff des „Leidens“ (malady) betrachtet, den er – damals durchaus unüblich – als den Gegenbegriff zur „Gesundheit“ identifizierte: „The opposite of health is *malady* (German *Leiden*) that is a much broader class than disease, containing besides disease also injury, wound, lesion, defect, deformity, disorder, syndrome, disability, impairment, and the like“ [19]. Als von einem Leiden geplagt – „being afflicted by a malady“ identifizierte Sadegh-Zadeh damals anstelle von „krank“ das Gegenteil von „gesund“. Später ersetzte er diesen sperrigen Ausdruck (auch wegen seiner Vagheit und „semantischen Verzerrung“) durch den Ausdruck „patienthood“ [18], den er in seiner „Fuzzy-Medizinteorie“ als das additive Inverse des Begriffs “Gesundheit” (*health*) definierte:

$$\text{Gesundheit} = 1 - \text{patienthood.} \quad (3)$$

Im Jahre 2000 veröffentlichte Sadegh-Zadeh den Artikel “Fuzzy Health, Illness, and Disease” [18], (Abb. 8), in dem er ein Grundgerüst für seine neue Theorie präsentierte:

“With Ω being the set of human beings at a particular time we have a fuzzy subset P (the set of patients) of Ω “whose members are to various extents characterized by discomfort, pain, endogenously threatened life, loss of autonomy, loss of vitality, and loss of pleasure. The extent to which an individual is a member of this fuzzy set P is called the degree of patienthood.” [18]

Die Zugehörigkeitsfunktion $\mu_p(x) \in [0, 1]$ gibt dabei den Grad der “patienthood“ eines Individuums $x \in \Omega$ an. H – die Menge aller gesunden Individuen ist dann das Komplement von P , der Menge aller von einem Leiden geplagten Individuen. Weiterhin gilt dann:

$$\mu_p(x) = \text{Grad der patienthood von } x, \quad (4)$$

$$\mu_h(x) = \text{Grad der Gesundheit von } x. \quad (5)$$

Dabei gilt per definitionem: $\mu_h(x) = 1 - \mu_p(x)$, $H = \{(x, \mu_h(x)) \mid x \in \Omega\}$.

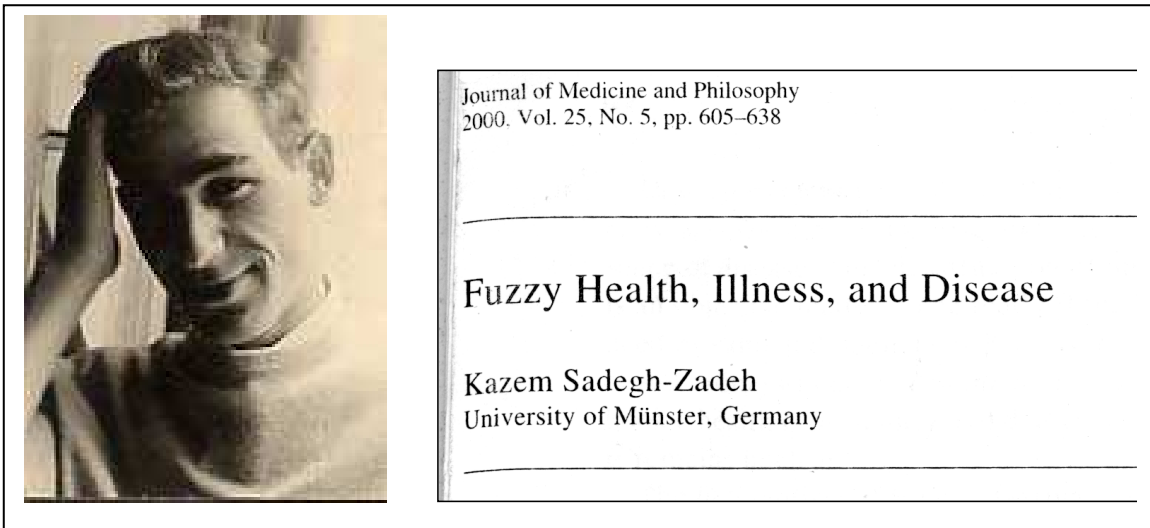


Abb. 6: Links: Kazem Sadegh-Zadeh in den 1970-er Jahren.
Rechts: Überschrift seines Artikels aus dem Jahre 2000 [19].

In Sadegh-Zadehs Vorschlag zur Nosologie werden Krankheitseinheiten (*disease entities*) – „such as myocardial infarction, gastric ulcer, and multiple sclerosis” – durch Fuzzy Sets „auf einen Begriff gebracht“: disease entities “may be conceptualized as fuzzy sets”. Das bedeutet, dass die auf eine Krankheit hinweisenden Kriterien, Symptome und Zeichen zu einem gewissen Grad zu einer individuellen Krankheit gehören. „Thus, an individual disease would appear as a multidimensional cloud rather than a clear-cut phenomenon” ([19], S. 629).⁴ Eine einzelne Krankheit würde sich dann völlig unterschiedlich in verschiedenen Patienten und in unterschiedlichem Zustand ausprägen. Die unzähligen Patienten mit einer Krankheit, wie z. B. einem Myocardinfarkt sind dann nur verstreute punktuelle Erscheinungen einer solchen multidimensionalen Wolke.

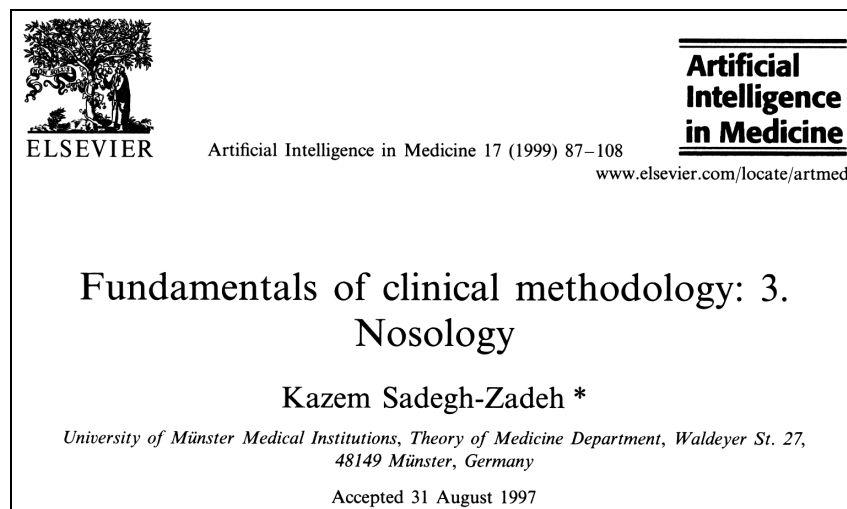


Abb. 9: Überschrift des Artikels von Sadegh-Zadeh aus dem Jahre 1999 [15]

⁴ Für eine ausführliche Darstellung dieses Teils von Sadegh-Zadehs Fuzzy-Medizinphilosophie und eine entsprechende Überführung dieser Konzepte in ein Java-Programm siehe [20, 21].

In seinem dritten, speziell der Nosologie gewidmeten Teil seiner „Fundamentals of Clinical Methodology“ – einer vierteiligen Aufsatzserie, die 1994 bis 2000 erschien [22-25] – verwies Sadegh-Zadeh darauf, dass die konventionellen nosologischen Systeme, die Krankheiten bezüglich ihrer Ursache (*Etiologie*), ihrer Entwicklung (*Pathogenese*, *Epidemiologie*) oder ihrer Symptome klassifizieren, einige Probleme aufwerfen und es deshalb nötig sei, sie zu verbessern. Effektive nosologische Systeme sollten seiner Auffassung nach nicht nur Datensammlungen klassifizierter Krankheiten umfassen sondern auch klinische Diagnose- und Therapiemöglichkeiten bereitstellen, so dass sie dann auf ihre Eignung im klinischen Einsatz und für die Entscheidungsunterstützung des Arztes evaluiert werden können ([24], S. 102).

Wir hatten Krankheiten oben schon als Aufzählungen – Mengen von Symptomen und Zeichen – charakterisiert. Allerdings sind die Symptome und Zeichen in den allermeisten Fällen mit Unsicherheit behaftet. Diese Unsicherheit beabsichtigte Sadegh-Zadeh mit Hilfe der Fuzzy Sets zu modellieren. Er griff er auf Koskos Darstellung der Fuzzy Sets als Punkte im Hyperwürfel zurück und etablierte eine „Nosologie in the unit hypercube“ denn: “We can in this fashion represent an individual disease entity as a point in a unit hypercube.” ([24], S. 102).

Eine zweite Krankheitseinheit mit ähnlichen Symptomen und Zeichen, die wir ebenfalls in diesem Hyperwürfel darstellen, können wir dann nicht nur als ebenfalls klassifiziert ansehen, sondern wir können sie sogar mit der ersten oder auch vielen anderen als Punkte des Hyperwürfels repräsentierten Krankheiten vergleichen. Dazu müssen wir lediglich ihren Abstand voneinander bestimmen. Im Hyperwürfel lässt sich leicht der euklidische Abstand oder die Hamming-Distanz zwischen zwei Punkten berechnen, und auf diese Weise lassen sich gegebenenfalls auch Aussagen über Beziehungen zu anderen Erkrankungen und zu deren Diagnose formulieren. Dies könnte auch für ein unbekanntes Krankheitsbild nützlich sein. Darüber hinaus ist es möglich, die Entwicklung einer Krankheit als „Bahn“ des sie repräsentierenden Punktes im Hyperwürfel zu verfolgen: Jeder Punkt dieser Trajektorie entspricht dann der Krankheit zu einem bestimmten Zeitpunkt und so können dann auch Behandlungsergebnisse, z. B. Erfolge von Therapien reproduziert und die Wiederholung eines Misserfolgs verhindert werden.

4 Fuzzy-Genome

„Medicine at the turn of the century is characterized by the deepest change it has ever been subject in its history, i.e. its transformation from healing profession to a branch of biotechnology. Viewed from an evolutionary perspective, this transformation appears as an aspect of a Darwin-Lamarckian autoevolution of life on earth.” ([26], S. 1)

Mit diesen kritischen Worten leitete Sadegh-Zadeh kurz vor dem Jahrtausendende seine Wende von den Arbeiten zur Fuzzifizierung von Gesundheit, patienthood und Krankheiten, zu einer der interessantesten Errungenschaften in der Geschichte der Lebenswissenschaften zu: dem Erbgut von Lebewesen, seiner Analyse und seiner Modifikationsmöglichkeiten [26] (Abb. 10).

Nachdem das genetische Material des Menschen analysiert ist, wurde die Biotechnologie ein wichtiger Faktor für Medizinische Diagnostik und Therapie. Entscheidende Fragen wie „Ist diese RNA-Sequenz ein HIV?“ können durch Sequenzvergleichsverfahren beantwortet werden. Zur Beantwortung von Fragen dieser Art ist allerdings eine zuverlässige Technologie für den Vergleich von bekannten mit unbekanntem Nukleinsäurese-

quenzen nötig. Sadegh-Zadeh (Abb. 11) stellte im Jahre 2000 eine Methode vor, um RNA- oder DNA-Sequenzen als Fuzzy Sets bzw. als Punkte im Hyperwürfel darzustellen und somit vergleichbar zu machen, indem ihr Abstand voneinander im Hyperwürfel bestimmt wird:

„DNA and RNA are linear polymers of nucleotides and are therefore called polynucleotides. We will formalize polynucleotide sequences and will then show, first, that a polynucleotide is an ordered fuzzy set and as such it has, second, an unequivocal *fuzzy code*.“ ([26], S. 2)

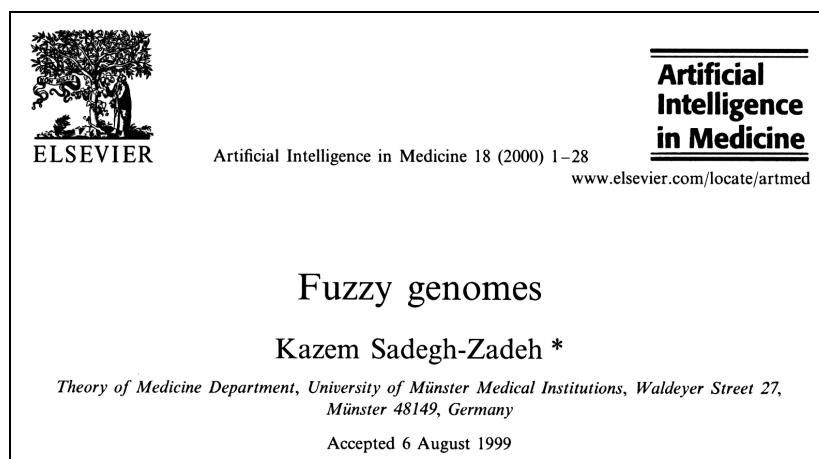


Abb. 10: Überschrift eines Artikels von Sadegh-Zadeh aus dem Jahre 1999 [19].

Gene sind Abschnitte auf der DNA. Sie enthalten die Information zur Herstellung von RNA und sind damit Träger der Erbinformation des Lebewesens. Die Sequenzen der DNA oder RNA bestehen vier verschiedenen Basen: DNA enthält die Basen Thymin (T), Cytosin (C), Adenin (A) und Guanin (G), während RNA zwar auch die letzten drei Basen enthält, anstelle von Thymin aber Uracil (U) steht. Basen können defekt sein und Gene sind oft sehr schwer auf der DNA-Sequenz zu lokalisieren. Beispielsweise gibt es so genannte „Springende Gene“ (Transposons), die ihren Ort im Genom verändern können. Gene sind daher Kandidaten für „unscharfe Begriffe“, und dies war möglicherweise eine Motivation für Sadegh-Zadeh, DNA/RNA-Sequenzen als Fuzzy Sets darzustellen. Da die Grundbausteine dieser Sequenzen jeweils vier Basen sind – A, C, G und T bzw. U –, die jeweils zu einem gewissen Grad gegeben sein können, repräsentiert in Sadegh-Zadehs Theorie jeweils ein Punkt im $4n$ -dimensionalen Hyperwürfel eine Sequenz der DNA oder RNA.

Im n -dimensionalen Hyperwürfel können nun wieder durch die Berechnung der Abstände der Punkte voneinander Unterschiede oder Ähnlichkeiten erkannt werden.

Der Buchstabe c bezeichne für alle $n \in \mathbb{N}$ die Summe aller Zugehörigkeitswerte eines Fuzzy Set (*sigma count*), $c = \sum_{i=1}^n \mu(x_i)$, dann ergibt sich folgende Gleichung zur Berechnung der Differenz zweier Fuzzy Sets A, B :

$$\text{differ}(A, B) = \frac{\sum_i \max(0, \mu_A(x_i) - \mu_B(x_i)) + \sum_i \max(0, \mu_B(x_i) - \mu_A(x_i))}{c(A \cup B)} \quad (6)$$

Nachdem wir schon in Kapitel 3 darauf hingewiesen haben, dass die Differenz im Hyperwürfel z. B. als Hamming-Distanz definiert werden kann, läßt sich die Differenz (*difference*) zweier Polynukleotide A und B leicht berechnen, und die Ähnlichkeit (*similarity*) zweier Sequenzen ergibt sich dann als das Inverse dieser Differenz:

$$\text{similar}(A,B) = \frac{c(A \cap B)}{c(A \cup B)} \quad (7)$$

Den Grad der Fuzziness einer Menge bezeichnet Sadegh-Zadeh mit *ent* (*entropy*), dem Maß, das wir schon als Koskos Entropie in Kapitel 2 erwähnten. Dieses Maß bildet den gesamten Hyperwürfel Ω in das Einheitsintervall ab:

$$\text{ent} : F(2^\Omega) \rightarrow [0, 1] \quad (8)$$

Um die Entropie einer Menge betrachten zu können, müssen (siehe Kapitel 2) die nächste und die weitest entfernte gewöhnliche Menge (als Eckpunkt im Hyperwürfel) ermittelt werden. Diese beiden Punkte A_{near} und A_{far} existieren immer im Hyperwürfel! Beispielsweise sei $A = (0.2, 0.8, 0.6)$ der unsere Menge repräsentierende Punkt im Hyperwürfel. Der nächste und der weitest entfernte Punkt sind dann: $A_{\text{near}} = (0, 1, 1)$ und $A_{\text{far}} = (1, 0, 0)$. Die Fuzzy Entropie einer Menge A ist dann der rationale Bruch der Hamming-Distanzen von A zur Ecke A_{near} und von A zu Ecke A_{far} (vgl. Kapitel 2):

$$\text{ent}(A) = \frac{l^1(A, A_{\text{near}})}{l^1(A, A_{\text{far}})} \quad (9)$$

Die Klarheit (*clarity*) einer Menge A , $\text{clar}(A)$, definierte Sadegh-Zadeh dann als das Komplement der Fuzzy Entropie von A :

$$\text{clar}(A) = 1 - \text{ent}(A) \quad (10)$$

In den Eckpunkten des Hyperwürfels gilt $\text{clar}(A) = 1$, während für den Mittelpunkt des Hyperwürfels gilt: $\text{clar}(A) = 0$.

Daraus können wir folgern, dass ein real existierendes Polynukleotid wie z. B. UAC die Fuzzy-Entropie 0 hat, und deshalb die Klarheit 1, während ein Fuzzy-Polynukleotid von einem realen Polynukleotid nah bzw. weit entfernt ist – zu einem gewissen Grad! [26].



Abb. 11: Kazem Sadegh-Zadeh um 2006.

5 Die Definition des Gens – ein historischer Ausblick

Eine „Geschichte des Genbegriffs“ sollte folgende wichtige Eckdaten zur Orientierung nennen:

- 1865 formulierte Gregor Mendel die Vererbungsregeln. Als erster ging er damals von der Annahme aus, dass es Faktoren – er nannte sie „Elemente“ – gebe, die von den Eltern auf die Nachkommen weitergegeben werden. Mendel nahm an, dass es sich um messbare Größen handelt.
- 1868 ging Charles Darwin von Vererbungs-Partikel aus, die er „gemmules“ nannte. Diese würden in den Blutkreislauf eindringen, um die Gameten zu erreichen. Dort kommt es zur Vererbung von Eigenschaften. Da alle Teile des Organismus daran beteiligt seien, sprach Hugo de Vries von „pangenes“.

Keiner dieser Wissenschaftler dachte nach den Studien des Molekularbiologen und Wissenschaftshistorikers Hans-Jörg Rheinberger bei diesen Partikeln an eine besondere „Vererbungs-substanz“. Vielmehr gingen sie davon aus, dass sie aus dem Körpermaterial des jeweils betrachteten Organismus bestünden. [27]. Der Name „Gen“ (griechisch: *gignesthai*, *genésthai* = entstehen, werden) ist nicht aus dem 19. Jahrhundert.

- 1909 wählte Wilhelm Johannsen die Bezeichnung „Gen“, um einen Operanden zu benennen, der alle für die Vererbung nötigen Objekte bereitstellt. Viele Eigenschaften des Organismus würden durch spezifische und unterscheidbare Zustände und Basen festgelegt, und diese Zustände und Basen kombinierte Johannsen zu einem Gen.
- 1928 fand Frederick Griffith im so genannten „Griffiths Experiment“ Anhaltspunkte dafür, dass Gene von einem Organismus zum anderen transportiert werden.
- 1941 erzielten George Wells Beadle und Edward Lawrie Tatum experimentell das Ergebnis, dass Mutationen in Genen für Defekte im Stoffwechsel verantwortlich sind. Sie gingen nun von einer direkten Verbindung zwischen Genen und der enzymatischen Reaktion aus, die so genannte „one gene, one enzyme Hypothese“.
- 1944 entdeckten Oswald Avery, Colin MacLeod und Maclyn McCarty, dass die DNA die genetische Information enthält und auch das Material ist, aus dem Gene und Chromosomen bestehen.
- 1953 fanden James D. Watson und Francis Crick – aufbauend auf Ergebnissen von Rosalind Franklin – die Doppelhelixstruktur der DNA.
- 1969 konnte Jonathan Beckwith ein einzelnes Gen isolieren.

Die „Geschichte des Gens“ ist ein umfangreiches Gebiet, und die Definition des Gens variierte im 20. Jahrhundert beträchtlich, vor allem auch in Abhängigkeit von zeitgenössischen wissenschaftlichen und technologischen Errungenschaften. Um sich auf die heute weitgehend akzeptierte Definition des *Sequence Ontology Consortium* zu einigen, was als „Gen“ bezeichnet werden soll, brauchte das aus 25 Wissenschaftlern zusammengesetzte Gremium etwa zwei Tage:

“a locatable region of genomic sequence, corresponding to a unit of inheritance, which is associated with regulatory regions, transcribed regions and/or other functional sequence regions”. [28].

In seinem Beitrag über “Gene” in der *Stanford Encyclopedia of Philosophy* schrieb H.-J. Rheinberger:

“There has never been a generally accepted definition of the 'gene' in genetics. There exist several, different accounts of the historical development and diversification of the gene concept as well. Today, along with the completion of the human genome sequence and the beginning of what has been called the era of postgenomics, genetics is again experiencing a time of conceptual change, voices even being raised to abandon the concept of the gene altogether.” [27]

Sadegh-Zadehs Vorschlag zum “Fuzzy-Genom”, die Repräsentation von Erbinformationssequenzen als Punkte im Hypercube und ihre Differenzen bzw. Übereinstimmungen als Funktionen eines Pairs solcher Punkte zu interpretieren ist ein Ansatz, den Begriff vom Gen in konzeptuell fließenden oder unscharfen Grenzen zu halten.

Acknowledgment

Die AutorInnen danken Professor Dr. Kazem Sadegh-Zadeh für hilfreiche Diskussionen, sowie ihm und seiner Frau für ihre große Gastfreundschaft in Tecklenburg.

6 Literatur

- [1] Seising, R.: *Die Fuzzifizierung der Systeme. Die Entstehung der Fuzzy Set Theorie und ihrer ersten Anwendungen – Ihre Entwicklung bis in die 70er Jahre des 20. Jahrhunderts*. Stuttgart: Franz Steiner Verlag (Boethius: Texte und Abhandlungen zur Geschichte der Mathematik und der Naturwissenschaften Band 54) 2005. Englischsprachige Ausgabe: *The Fuzzification of Systems. The Genesis of Fuzzy Set Theory and Its Initial Applications – Developments up to the 1970s* (Studies in Fuzziness and Soft Computing Vol. 216) Berlin [u.a.]: Springer 2007.
- [2] Zadeh, L. A.: Fuzzy Sets and Systems, in: J. Fox, (ed.): *System Theory*. Microwave Research Institute Symposia Series XV, Brooklyn, New York: Polytechnic Press 1965, S. 29-37.
- [3] Zadeh, L. A.: Fuzzy Sets, *Information and Control*, Vol. 8, 1965, S. 338-353.
- [4] Chang, C.-L.: Fuzzy Topological Spaces, *Journal of Mathematical Analysis and Applications*, 24, S. 182-190, 1968.
- [5] Zadeh, L. A.: Probability Measures of Fuzzy Events, *Journal of Mathematical Analysis and Applications*, 10, S. 421-427, 1968.
- [6] Zadeh, L. A.: Fuzzy Algorithms, *Information and Control*, Vol. 12, 1968, S. 99-102.
- [7] Zadeh, L. A. Towards a Theory of Fuzzy Systems: In: R. E. Kalman and N. DeClaris, (Eds.): *Aspects of Network and System Theory*. New York [u.a.]: Holt, Rinehart and Winston 1971, S. 469-490. Auch: *Toward a Theory of Fuzzy Systems*, Electronic Research Laboratory, University of California, Berkeley 94720, Report No. ERL-69-2, June 1969, S. 1.
- [8] Kosko, B.: *Foundations of Fuzzy Estimation Theory*. Ph. D. Dissertation, Department of Electrical Engineering, University of California at Irvine, June 1987; Order Number 8801936, University Microfilms International, 300 N. Zeeb Road, Ann Arbor, Michigan 48106.
- [9] Kosko, B.: Counting with fuzzy sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8, July 1986, S. 556-557.
- [10] Zadeh, L. A.: A Computational Approach to Fuzzy Quantifiers in Natural Languages. *Computers and Mathematics*, 9, 1, 1983, S. 149-184.
- [11] Kosko, B.: Fuzzy Entropy and Conditioning. *Information Sciences*, 40, 1986, S. 165-174.

- [12] Kosko, B.: Fuzziness vs. Probability, *International Journal of General Systems*, 17, S. 211-240.
- [13] Seising, R.: *Eine kleine Geschichte der Fuzzy-Systeme in der Medizintechnik*. In: R. Mikut, M. Reischl (Hrsg.): Proceedings 12. Workshop „Fuzzy Systeme“ des GMA-FA 5.22 „Fuzzy Control“ (Fachausschuss der VDI/VDE Gesellschaft für Mess- und Automatisierungstechnik) und der GI-FG „Fuzzy-Systeme und Soft-Computing“ (Fachgruppe der Gesellschaft für Informatik) am 14./15. November 2002 in Dortmund, Gästehaus der Universität Dortmund „Haus Bommerholz“. Forschungszentrum Karlsruhe: 2002, S. 27-42.
- [14] Seising, R.: Zwischen Symptomen und Diagnosen: vom Raum medizinischen Wissens, seiner unscharfen Struktur und deren technologischer Bewältigung in der zweiten Hälfte des 20. Jahrhunderts. *Technikgeschichte* Bd. 71, Heft 1, 2004, S. 19-57.
- [15] Seising, R.: From Vagueness in Medical Thought to the Foundations of Fuzzy Reasoning in Medical Diagnosis. *Artificial Intelligence in Medicine*, 38, 2006, S. 237-256.
- [16] Sadegh-Zadeh, K.: Zur Logik und Methodologie der ärztlichen Urteilsbildung. *Methods of Information in Medicine*, 11, 4, 1972, S. 203-212.
- [17] Sadegh-Zadeh, K.: Concepts of Disease and Nosological Systems. *Metamed*, 1, 1977, S. 4-41.
- [18] Sadegh-Zadeh, K.: Fuzzy Health, Illness, and Disease, *The Journal of Medicine and Philosophy*, 25, 2000, S. 605-638.
- [19] Sadegh-Zadeh, K.: *Organism and disease as fuzzy categories*, presented at the conference on Medicine and Philosophy, Humboldt University of Berlin, July 2, 1982.
- [20] Limberg, J., Seising, R.: *Fuzzy-Konzepte für die Begriffe Gesundheit, Kranksein und Krankheit. Darstellung und Implementierung eines Ansatzes von Kazem Sadegh-Zadeh*. In: R. Mikut, M. Reischl (Hrsg.), Schriftenreihe des Instituts für Angewandte Informatik / Automatisierungstechnik Universität Karlsruhe (TH) Band 15: Proceedings 16. Workshop „Computational Intelligence“ des GMA-FA 5.14 „Computational Intelligence“ (Fachausschuss der VDI/VDE Gesellschaft für Mess- und Automatisierungstechnik) und der GI-FG „Fuzzy-Systeme und Soft-Computing“ (Fachgruppe der Gesellschaft für Informatik) 29. November – 1. Dezember 2006, Universitätsverlag Karlsruhe 2006, S. 248-261.
- [21] Limberg, J., Seising, R.: *Fuzzy Health, Illness, and Disease – Sadegh-Zadeh’s Framework and a program to identify diseases*. Proceedings of the 2007, IEEE International Conference on Fuzzy Systems :: Intelligence is Fuzzy ::, Imperial College, London, UK, July 23-26, 2007, S. 792-797..
- [22] Sadegh-Zadeh, K.: Fundamentals of clinical methodology: 1. Differential indication, *Artificial Intelligence in Medicine*, 6, 1994, S. 83-102.
- [23] Sadegh-Zadeh, K.: Fundamentals of clinical methodology: 2. Etiology, *Artificial Intelligence in Medicine*, 12, 1998, S. 227-270.
- [24] Sadegh-Zadeh, K.: Fundamentals of clinical methodology: 3. Nosology, *Artificial Intelligence in Medicine*, 17, 1999, S. 87-108.
- [25] Sadegh-Zadeh, K.: Fundamentals of clinical methodology: 4. Diagnosis, *Artificial Intelligence in Medicine*, 20, 2000, S. 227-241.
- [26] Sadegh-Zadeh, K.: Fuzzy Genomes, *Artificial Intelligence in Medicine*, 18 (2000) S. 1-28.
- [27] Hans-Jörg Rheinberger: *Gene*. Stanford Encyclopedia of Philosophy. Available at: <http://plato.stanford.edu/entries/gene/>
- [28] Neurophilosophy. *The Concept Of A Gene Has Evolved*. Available at: <http://neurophilosophy.wordpress.com/2006/05/30/the-concept-of-a-gene-has-evolved/>

Visualisierung temporaler Abhängigkeiten in Bayesschen Netzen

Matthias Steinbrecher and Rudolf Kruse

Institut für Wissens- und Sprachverarbeitung
Otto-von-Guericke-Universität Magdeburg
Universitätsplatz 2, 39106 Magdeburg
{msteinbr, kruse}@iws.cs.uni-magdeburg.de

1 Zusammenfassung

Eine für heutige Industrieunternehmen permanente Herausforderung besteht in der wettbewerbsbestimmten Notwendigkeit, Innovationen und Weiterentwicklungen immer schneller zu marktreifen Produkten entwickeln zu können. Um dennoch zeitnah auf Produktionsprobleme oder Qualitätsmängel zu reagieren, sind speziell zugeschnittene Data-Mining-Lösungen notwendig, die den Hersteller in die Lage versetzen, umfangreiche Daten auf Anomalien oder Störzusammenhänge hin zu untersuchen.

Da viele Problemarten nicht sofort, sondern allmählich während des Einsatzes auftreten, sind Analysetechniken vonnöten, die es erlauben, temporale Veränderungen innerhalb der Daten aufzudecken.

Bayessche Netze [3,4] stellen eine bekannte Methode dar, die auf sehr intuitive Weise Abhängigkeiten zwischen Einflussgrößen und anderen Variablen darstellen kann. Diese Zusammenhänge können sowohl von Experten vorgegeben als auch automatisch aus vorhandenen Daten extrahiert werden [2,1].

Während die Graphenstruktur die qualitative Information trägt (welche Attribute besitzen eine statistische Abhängigkeit mit der Klassenvariable?), kodieren die Parametertabellen eines solchen Netzes die quantitative Information, die verwendet werden kann, um dem Experten wertvolle lokale Zusammenhänge aufzuzeigen [5] und graphisch intuitiv aufzubereiten.

In dieser Arbeit wird eine Erweiterung vorgestellt, die in der Lage ist, die temporalen Zusammenhänge in den Daten zu visualisieren und so dem Experten ein Werkzeug in die Hand gibt, zeitliche Trends (z. B. das schleichende Auftreten eines Fehlers) zu lokalisieren.

Literatur

1. Christian Borgelt and Rudolf Kruse. *Graphical Models — Methods for Data Analysis and Mining*. John Wiley & Sons, United Kingdom, 2002.
2. David Heckerman. A Tutorial on Learning with Bayesian Networks. Technical Report MSR-TR-95-06, Microsoft Research, Advanced Technology Division, Redmond, WA, 1995. Revised June 96.
3. J. Pearl. Aspects of Graphical Models Connected with Causality. In *49th Session of the International Statistics Institute*, 1993.
4. Judea Pearl and Stuart Russel. Bayesian Networks, 1994.
5. Matthias Steinbrecher and Rudolf Kruse. Visualization of local dependencies of possibilistic network structures. In Rafael Bello, Rafael Falcón, and Yudel Gómez, editors, *Proceedings of the International Symposium of Fuzzy and Rough Sets (ISFUROS), UCLV, Santa Clara, Cuba*, pages 77–80, 2006.

Künstlich Neuronales Netz zur Analyse der Geometrie von großflächig gekrümmten Bauteilen

M. Schneider¹, R. Tillmann¹, U. Lehmann¹, J. Krone¹, P. Langbein¹,
U. Stark², J. Schrickel², Ch. Ament³, P. Otto³

¹Fachhochschule Südwestfalen, Frauenstuhlweg 31, 58644 Iserlohn

Tel. (02371) 566-303 Fax (02371) 566-209

E-Mail: {MSchneider, RTillmann, Lehmann, Krone, Langbein}@fh-swf.de

²Airbus Deutschland GmbH, Kreetslag 10, 21129 Hamburg

Tel. (040) 743-81268 Fax (040) 743-74794

E-Mail: {ulrich.stark, joerg.schricketl}@airbus.com

³TU Ilmenau, Fachgebiet, Postfach 10 55 65, 98684 Ilmenau

Tel. (03677) 69-2815 Fax (03677) 69-1434

E-Mail: {christoph.ament, peter.otto}@tu-ilmenau.de

Zusammenfassung

Im Rahmen des Projektes in Kooperation mit der Fa. Airbus wird das Materialverhalten der Bauteile einer Flugzeugrumpfssektion mit Künstlich Neuronalen Netzen (KNN) von der Kompetenzplattform Computer Vision based on Computational Intelligence an der Fachhochschule Südwestfalen analysiert, modelliert und simuliert.

1 Motivation und Zielsetzung

Großflächig gekrümmte Bauteile, wie sie z. B. bei der Flugzeugrumpfmontage verwendet werden, lassen sich aufgrund ihrer konstruktiven Eigenschaften nur schwer mit klassischen Methoden wie z. B. Finite-Element-Methode kostengünstig modellieren. Ein Flugzeugrumpf besteht aus mehreren Sektionen (siehe Abb. 1.1).

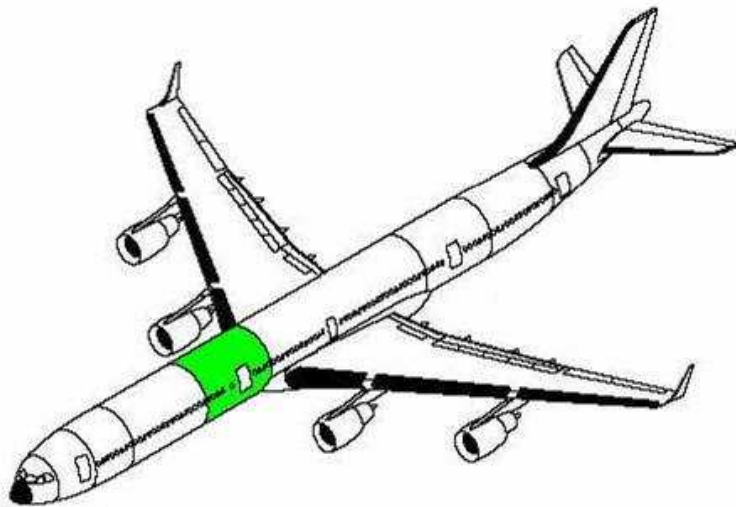


Abbildung 1.1: Rumpfssektionen des Airbus A340-600

Solch eine Flugzeugrumpfsektion besteht aus mehreren Schalen, welche während der Montage in einem Bauplatz mit beweglichen Aktoren eingehängt sind (siehe Abb. 1.2). Mit Hilfe dieser Aktoren kann das Bauteil im Bauplatz bewegt werden. Die Aktoren können vom Bauplatzbediener sehr genau und individuell auf eine vorgegebene Position innerhalb eines bestimmten Intervalles gefahren werden. Das Materialverhalten großflächig gekrümmter Bauteile mit Künstlich Neuronalen Netzen zu modellieren, erscheint aufgrund bereits erfolgreich durchgeführter Projekte [1], [2] aussichtsreich.

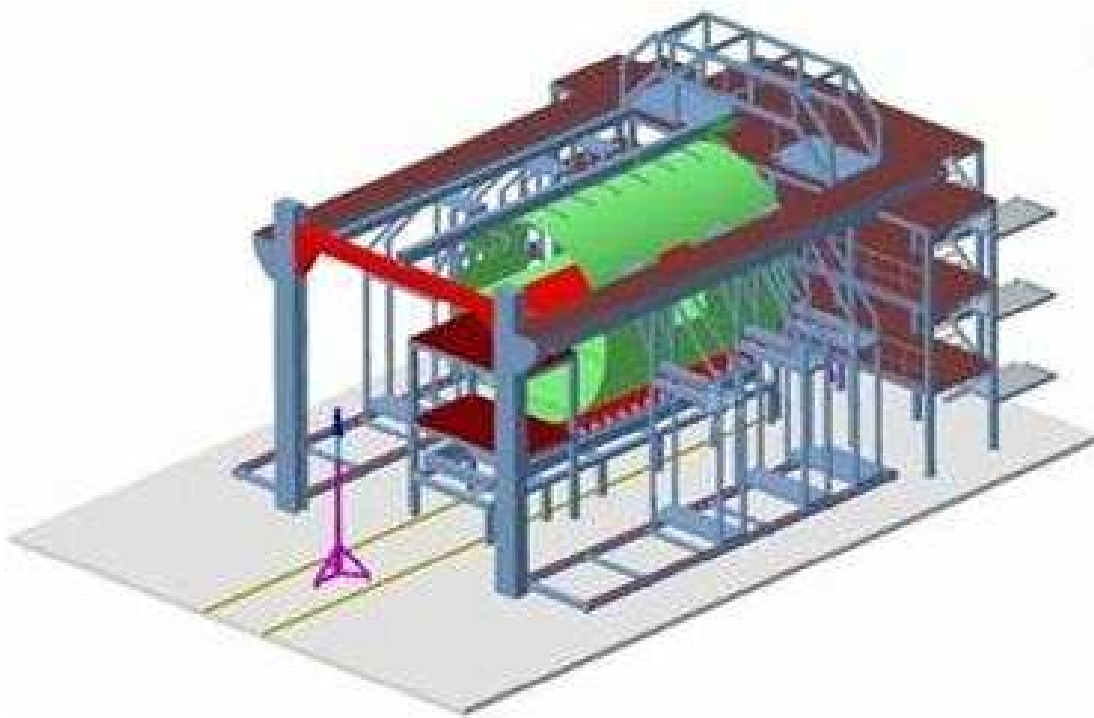


Abbildung 1.2: Bauplatz mit einer Flugzeugrumpfsektion

2 Datengenerierung

Die Messdaten wurden in einer Versuchsreihe an gleichen Rumpfbauteilen verschiedener Produktionen durchgeführt, um den Zusammenhang zwischen der Bewegung der Aktoren für die Positionierung und der sich ergebenden Bauteilgeometrieänderung zu modellieren. Jeder Aktor kann sich in x-, y-, z-Richtung unabhängig voneinander bewegen. Ein hochgenaues Messsystem liefert die Koordinaten der Messpunkte, welche in den Rumpfbauteilen planmäßig integriert sind. Anhand dieser Messpunkte wird die Geometrie der Seitenschale und ihre Lage im Raum ermittelt. Die Datensätze aus den einzelnen Messungen bilden die Grundlage für die Entwicklung eines Simulationsmodelles zur Abbildung des Materialverhaltens einer Seitenschale in einem Künstlich Neuronalen Netz. Sie wurden vorher entsprechend aufbereitet und ausgewertet, bevor diese für das Training genutzt werden konnten. Das Training erfolgte nicht mit den absoluten Koordinaten, sondern mit den relativen Werten. Die relativen Werte entsprechen dabei der Abweichung zur Nullposition.

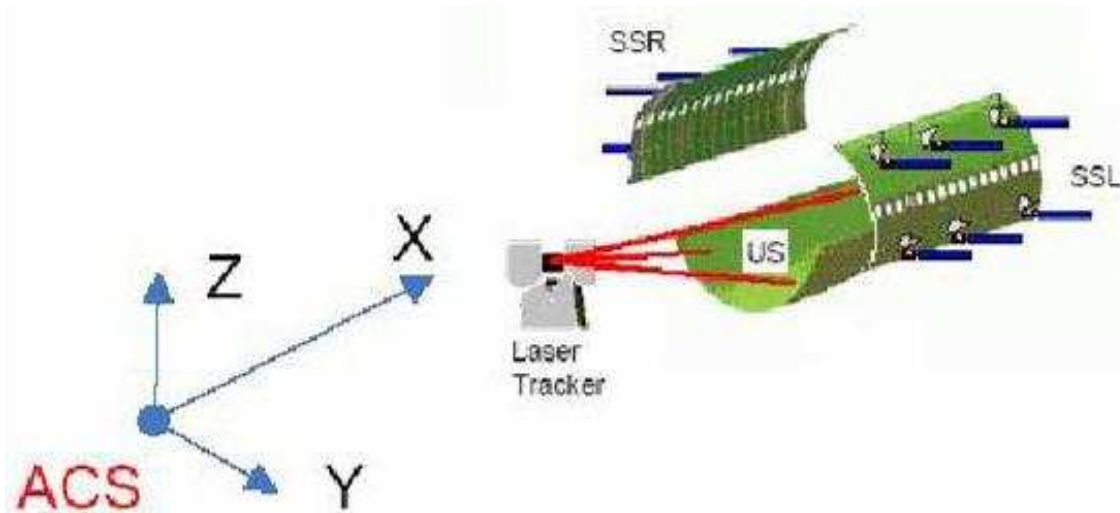


Abbildung 2.1: Koordinatensystem und Messsystem zur Bestimmung der Geometrie der Seitenschale

3 Künstlich Neuronale Netze (KNN)

Künstlich Neuronale Netze können anhand von Input- und Outputdaten mathematisch physikalische Zusammenhänge lernen. Durch das Training des Künstlich Neuronales Netzes entsteht ein mathematisches Modell, welches auch in der Lage ist zu generalisieren, d. h. es funktioniert auch für Daten, die dem Netz vorher noch nicht präsentiert wurden. Die Datensätze sollten den gesamten Arbeitsbereich bis in seine Extreme ausgewogen abdecken. Es ist auch auf eine ausreichende Anzahl von Datensätzen zu achten, damit ein stabiles und genaues Trainingsergebnis erzielt werden kann. Die Datensätze sollen ausserdem im laufenden Betrieb generiert werden, damit nicht ein Prozessverhalten gelernt wird, welches nicht dem realen entspricht.

3.1 Backpropagation

Der Backpropagation-Algorithmus (traingd) ist ein sehr gutes Trainingsverfahren, welches im Regelfall immer zum Einsatz kommen kann. Das Training dauert allerdings im Vergleich zu anderen Trainingsverfahren sehr lange, bis ein gutes Ergebnis mit dem trainierten Künstlich Neuronales Netz erzielt werden kann. Ein großer Vorteil ist, dass mit diesem Trainingsverfahren „praktisch alles“ trainiert werden kann, was sich mathematisch abbilden läßt. Es ist somit universell einsetzbar im Gegensatz zu anderen Trainingsverfahren, welche manchmal nur für spezielle Anwendungen geeignet sind. Diese universelle Einsatzmöglichkeit geht aber auf Kosten der Trainingsdauer, da eine hohe Genauigkeit bei komplexen Zusammenhängen meist erst nach sehr vielen Trainingsepochen erzielt wird. Bei dieser Untersuchung wurde der Batchmodus für das Training der Künstlich Neuronales Netze verwendet. Beim Batch-Training werden die Verbindungsgewichte zwischen den einzelnen Neuronen erst nach einem kompletten Trainingsdurchlauf aktualisiert [3].

3.2 Levenberg-Marquardt

Das Levenberg-Marquardt-Verfahren (trainlm) zeichnet sich durch eine stufenlose Interpolation zwischen dem Gradientenverfahren aus, welches sich durch eine gute Konvergenz fernab vom Minimum auszeichnet, und dem Verfahren der quadratischen Minimierung, welches besonders in der Nähe des Minimums geeignet ist. Das Konvergenzverhalten ist sehr gut, d. h. nach einer kurzen Epochenanzahl ist der mittlere quadratische Fehler des Trainings meist schon sehr klein. Der Levenberg-Marquardt-Trainingsalgorithmus stößt jedoch bei einer größeren Architektur des Künstlich Neuronalen Netzes und Trainingsdatenmenge auf einem Standard-PC an seine Grenzen, da er sehr speicherintensiv ist [4].

In Abb. 3.1 ist der Verlauf des mittleren quadratischen Fehlers des Backpropagation-Algorithmus (gestrichelt) für Training, Validierung und Test des Levenberg-Marquardt-Algorithmus (durchgezogene Linie) zu sehen. Der Levenberg-Marquardt-Trainingsalgorithmus erzielt nach wenigen Epochen einen wesentlich kleineren mittleren quadratischen Fehler als der Backpropagation-Trainingsalgorithmus (siehe Abb. 3.1).

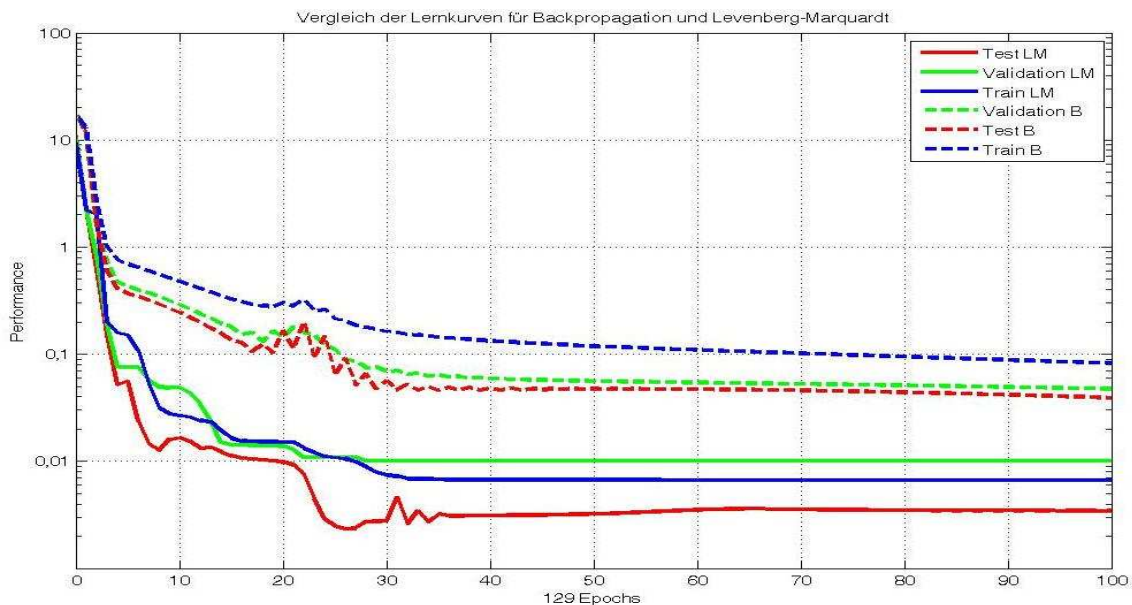


Abbildung 3.1: Vergleich der Fehlerkurve des Backpropagation und des Levenberg-Marquardt Algorithmus

3.3 Training

Für das Training der Künstlich Neuronalen Netze wird die Neural Network Toolbox (nntool) von MATLAB [5] genutzt. Als Input des Künstlich Neuronalen Netzes dienen die Bewegungen der Positionierer und als Output die x-, y- und z- Koordinate jedes einzelnen Messpunktes. Aufgrund einer geringen Anzahl von Messdaten wurde pro Messpunkt ein Künstlich Neuronales Netz entwickelt (siehe Abb. 3.2). Insgesamt standen 98 Datensätze zur Verfügung, welche folgendermaßen aufgeteilt wurden:

Trainingsdatensätze	74
Validierungsdatensätze	15
Testdatensätze	9
Gesamt	98

Diese Künstlich Neuronale Netze wurden parallel angeordnet und es konnte das Materialverhalten eines Bauteils einer Flugzeugrumpfssektion entsprechend der Genauigkeitsanforderung der Fa. Airbus abgebildet werden. Bei einer größeren Anzahl von Messdaten wäre es besser, das Materialverhalten in nur einem Künstlich Neuronale Netz zu trainieren, da dadurch ein noch besseres Simulationsergebnis zu erwarten wäre. Die Interaktion der Messpunkte untereinander kann in einem Neuro-Materialmodell besser abgebildet werden, so dass das Materialverhalten der Seitenschale noch realistischer simuliert werden kann. Aufgrund des besseren Verhältnisses von Verbindungsgewichten zu den Trainingsdaten wurde die Variante mit den 7 parallelen Künstlich Neuronale Netzen zu Gunsten der Generalisierungsleistung gewählt.

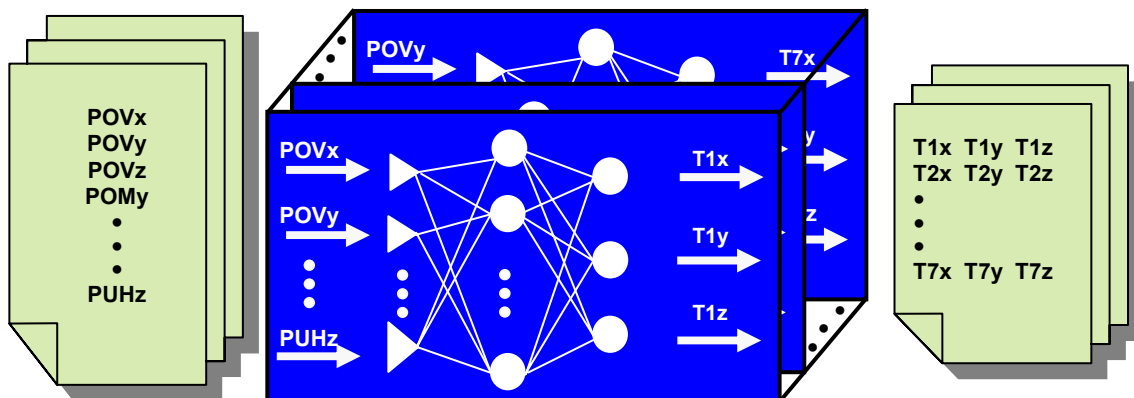


Abbildung 3.2: parallele Anordnung der Künstlich Neuronale Netze zur Simulation des Materialverhaltens

3.4 Initialisierung

Bevor Künstlich Neuronale Netze trainiert werden, sollen sie initialisiert werden. Dabei werden die Verbindungsgewichte mit zufälligen Startwerten belegt. Es ist darauf zu achten, dass die Startwerte der Verbindungsgewichte pro Layer nicht identisch sind. Durch das Durchbrechen dieser Symmetrie ist ein besseres Trainingsergebnis zu erzielen. Verschiedene Initialisierungen haben unterschiedliche Trainingsergebnisse zur Folge (siehe Abb. 3.3). Das Künstlich Neuronale Netz, welches mit der Initialisierung 1 (durchgezogene Linie) trainiert wurde, erzielte bessere Ergebnisse als das mit der Initialisierung 2 (gestrichelte Linie) trainierte Künstlich Neuronale Netz. Es ist deshalb empfehlenswert, mehrere Trainingseinheiten mit unterschiedlichen Initialisierungen bei gleicher Netzarchitektur und Parametrierung durchzuführen, um das beste Ergebnis zu erreichen.

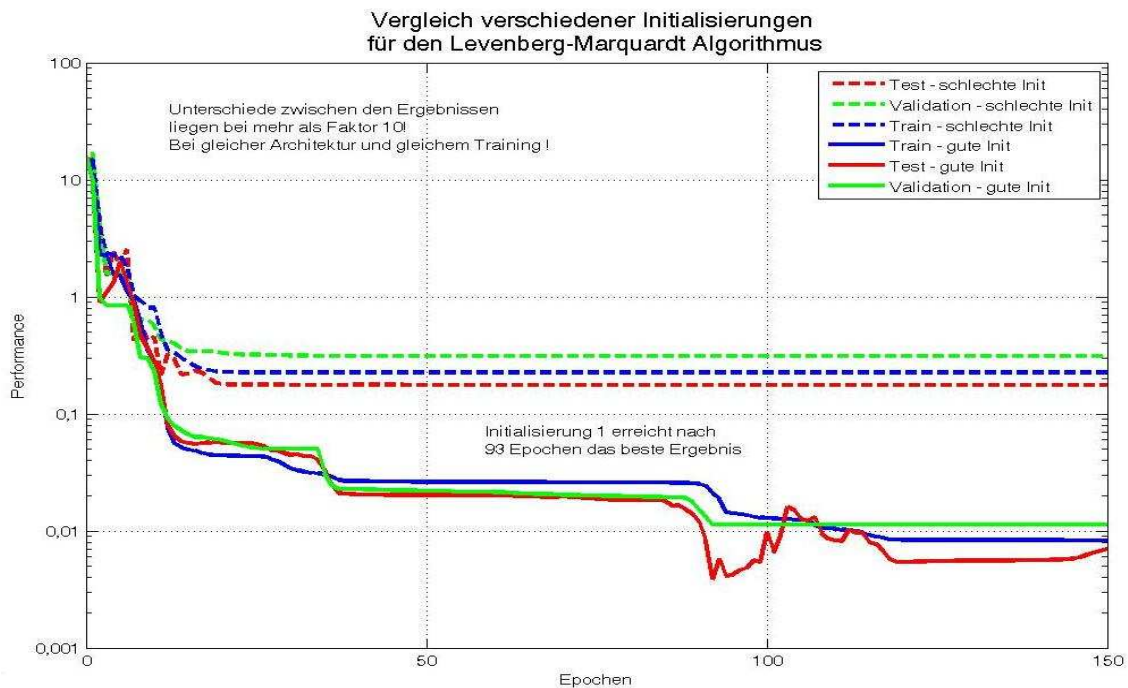


Abbildung 3.3: mittlerer quadratischer Fehler unterschiedlicher Initialisierungen für das Levenberg-Marquardt Trainingsverfahren

4 Ergebnisse

Künstlich Neuronale Netze sind gut geeignet das nichtlineare Verhalten von großflächig gekrümmten Bauteilen einer Flugzeugrumpfschale zu simulieren. Die Ergebnisse sind auf unbekannte Rumpfschalen gleichen Typs übertragbar. Der Vergleich der beiden Trainingsverfahren lieferte folgendes Ergebnis:

Das mit dem Levenberg-Marquardt-Algorithmus trainierte Künstlich Neuronale Netz lieferte einen durchschnittlich halb so großen Testfehler über alle Messpunkte (sog. Targets) wie das mit dem Backpropagation-Algorithmus trainierte Künstlich Neuronale Netz. Einzelne Targetkomponenten ließen sich mit dem Levenberg-Marquardt Algorithmus bis zu 70% besser vorhersagen (siehe Tab. 4.1).

	max Test	Ø Test
SSR_TOV	-40,86%	-39,41%
SSR_TOM	-67,64%	-39,17%
SSR_TOH	-60,08%	-35,40%
SSR_TUV	-40,55%	-12,35%
SSR_TUM	-61,43%	-43,75%
SSR_TUH	-21,36%	-26,37%
SSR_Mitte	-71,43%	-28,67%
Ø	-51,91%	-32,16%

Tabelle 4.1: Vergleich der Testergebnisse der 7 Netze, die mit dem Backpropagation-Algorithmus und dem Levenberg-Marquardt-Algorithmus trainiert wurden

In Tab. 4.1 wird das Testergebnis des mit dem Backpropagation-Algorithmus trainierte Künstlich Neuronale Netz mit dem mit Levenberg-Marquardt trainierten verglichen. Erklärung der Tab. 4.1 anhand eines Beispiels: Der maximale Testfehler der x-,y- oder z-Komponente eines Targets (SSR_TOV) mit dem Levenberg-Marquardt-Algorithmus war um 40% kleiner, bezogen auf alle Testdatensätze, als der maximale Testfehler der x-, y- oder z-Komponente des gleichen Targets mit dem Backpropagation-Algorithmus.

5 Ausblick

Im weiteren Verlauf des Projektes soll ein CI-Regler entwickelt werden, der auf Bauteilbewegungen aufgrund äusserer Einflüsse korrigierend einwirkt und das Bauteil wieder in seine Sollposition bringt. Durch dieses Regelungskonzept soll außerdem die Sollposition nach dem Eintakten des Bauteils in einem Bedienungsschritt unter Einhaltung vorgegebener Kraftgrenzen erzielt werden.

6 Danksagung

Die Kompetenzplattform Computer Vision und Computational Intelligence (<http://www.fh-swf.de/cv-ci>) wird vom Ministerium für Innovation, Wissenschaft, Forschung und Technologie des Landes Nordrhein-Westfalen gefördert.

Die Autoren wollen ihren aufrichtigen Dank an Herrn Dipl.-Ing. Ulrich Stark und Herrn Dipl.-Ing. Jörg Schrickel von der Airbus Deutschland GmbH für die Unterstützung dieser Forschungsarbeit und das zur Verfügung stellen realer Messdaten ausdrücken.

Literatur/Veröffentlichungen

- [1] Lehmann, U.; Johannes, H.; Hohage, S.; Woestmann, S.: Analyse eines Umformprozesses mit KNN und Data Mining. GI/VDI-Computational-Intelligence-Kongress, Baden-Baden im Mai 2000.
- [2] R. Tillmann, M. Schneider, J. Krone, U. Lehmann, J. Schrickel, U. Stark, D. Welbhoff: „Airbus - KNN zur fertigungsbegleitenden Analyse und Prozessregelung“, Ausstellung "Lebendige Forschung an Fachhochschulen im Landtag NRW", Düsseldorf, 24.01.-02.02.2007
- [3] A. Zell: Simulation neuronaler Netze. Oldenburg Verlag, 1994
- [4] Simon Haykin: Neural Networks: A Comprehensive Foundation. Prentice Hall; 2nd edition, 1998
- [5] <http://www.mathworks.com>

Prozessdiagnose und Monitoring feldbusbasierter Automatisierungsanlagen mittels selbstorganisierender Karten

Christian W. Frey

Fraunhofer-Institut IITB
Fraunhoferstraße 1, 76131 Karlsruhe
Tel.: +49-721-6091-332
Fax: +49-721-6091-413
E-Mail: christian.frey@iitb.fraunhofer.de

1. Motivation

Der Technologiewandel in der Automatisierungstechnik stellt die Anlagenbetreiber und vor allem das Inbetriebnahme- und Wartungspersonal vor große Herausforderungen. Funktionen, die bisher durch Mechanik oder Elektromechanik erbracht wurden, werden zunehmend durch Software-basierte mechatronische Systeme ersetzt. Bisher sichtbare und nachvollziehbare Funktionalität wird nun durch Software und digitale Kommunikationssysteme realisiert. Informationen, die z. B. bisher als analoges elektrisches Signal übertragen wurden und dadurch einfach zu prüfen waren, werden als digitale Telegramme in schnellen Datennetzen übertragen. Durch die Dezentralisierung der Intelligenz und den Übergang zu verteilten Automatisierungsarchitekturen werden auch das Engineering und die Inbetriebnahme von Anlagen immer komplexer. Die Kosten hierfür übertreffen mittlerweile die reinen Hardwarekosten für die Komponenten um ein Mehrfaches. Hier eröffnen wirksame Diagnosekonzepte ein großes Potential zur Kostenersparnis. Die Überprüfung der korrekten Funktionalität einer Anlage, die Fehlersuche, das frühzeitige Erkennen von drohenden Komponentenausfällen und insbesondere das Monitoring des Prozesses sind von entscheidender Bedeutung für das wirtschaftliche Betreiben komplexer Automatisierungsprozesse.

In dem vom BMBF geförderten Verbundvorhaben „Feldbus“ wurden am Fraunhofer-Institut IITB Softwarekomponenten zur agentenbasierten Diagnose von feldbusbasierten Automatisierungsanlagen entwickelt. Das entwickelte Diagnosekonzept stellt eine integrale Komponente der automatisierungstechnischen Anlage dar und ermöglicht eine permanente Diagnose des kommunikationstechnischen sowie physikalischen Verhaltens der Automatisierungsanlage. Im vorliegenden Beitrag wird die Architektur des entwickelten verteilten Diagnosekonzepts basierend auf Softwareagenten skizziert sowie die darin eingebettete Funktionalität zur Diagnose des unbekanntem Prozessverhaltens der zugrunde liegenden Automatisierungsanlage erläutert.

2. Diagnosekonzept basierend auf Softwareagenten

Der Feldbus bildet das zentrale „Nervensystem“ verteilter Automatisierungssysteme und stellt zum einen Gegenstand der Diagnose als auch Zugangspunkt für die Diagnose bzw. das Monitoring der Anlagenkomponenten und der Anlagenfunktionalität dar.

Die Diagnose technischer Systeme kann als zweistufiger hierarchischer Prozess aufgefasst werden, in dem quantitative Information in Form von Sensorsignalen in qualitative Information d. h. Diagnoseereignisse überführt wird [1]. Ziel der ersten Stufe, der Merkmalgenerierung, ist es, die erfassbaren Zustandsgrößen des Systems durch Transformation in eine geeignete komprimierte Repräsentationsform zu überführen, so dass sich die möglichen Diagnoseergebnisse zuverlässig widerspiegeln. Der zweite Schritt des Diagnoseprozesses, die Merkmalauswertung, stellt einen logischen Entscheidungsprozess dar, in dem das komprimierte quantitative Wissen, in Form der Merkmale, in qualitatives Wissen transformiert wird.

Das entwickelte Diagnosekonzept, dargestellt in Bild 1, basiert auf der Idee, kontinuierlich die auf den Feldbus übertragenen Telegramme zu analysieren und charakteristische Merkmale zu extrahieren, welche das Verhalten des jeweiligen Feldgerätes bzw. dessen Interaktion mit der Gesamtanlage in komprimierter Form beschreiben. Dabei können aus den Feldbustelegrammen sowohl Merkmale hinsichtlich des Kommunikationsverhaltens, z. B. durch Analyse der Antwortdauer eines Feldgerätes, wie auch Merkmale hinsichtlich des tatsächlichen physikalischen Prozessverhaltens, z. B. durch Analyse einer Stellgröße oder eines Messwertes, erzeugt werden. Die Merkmale stellen eine komprimierte Beschreibung der Anlagenfunktionalität auf Kommunikations- und Prozessebene dar, welche wiederum durch einen logischen Entscheidungsprozess in der übergeordneten Merkmalauswertungsstufe zu einem ganzheitlichen Diagnoseergebnis miteinander verknüpft werden können.

Die permanente Analyse des Feldbusdatenstroms erfordert eine effektive Verdichtung der auf dem Feldbus übertragenen Daten, da sonst die generierte Datenmenge nicht mehr handhabbar ist. So werden beispielsweise in einem PROFIBUS-Netz mit 12 Mbit/s Übertragungsgeschwindigkeit bereits 1 MByte Daten pro Sekunde übertragen - in Ethernet basierten Systemen mit Übertragungsraten von bis zu einem Gbit/s können dies über 100 MByte pro Sekunde sein. Es ist klar, dass unter diesen Randbedingungen auch eine selektive Aufzeichnung vereinzelter Feldbusdatentelegramme mit nachfolgender off-line Analyse an der Verfügbarkeit entsprechend hoher Rechenkapazität bzw. großer Speichermedien scheitert. Die Generierung aussagekräftiger Merkmale basierend auf dem immensen Datenstrom des Feldbusses stellt höchste Anforderungen an die Effizienz und Intelligenz der verwendeten Softwarekonzepte. Im Hinblick auf den Entwurf des Diagnosesystems muss zudem in Anbetracht der in einer Automatisierungsanlage zum Einsatz kommenden heterogenen Kommunikationsnetzwerke bzw. der verschiedenen Softwareplattformen auf Steuerungsebene ein besonderes Augenmerk auf die effektive Koordination und Kooperation der verteilten Softwarekomponenten gelegt werden. Unter Berücksichtigung dieser Anforderungen wurde am IITB ein hierarchisches Diagnose- und Monitoringkonzept für feldbusbasierte Anlagen basierend auf Softwareagenten entwickelt.

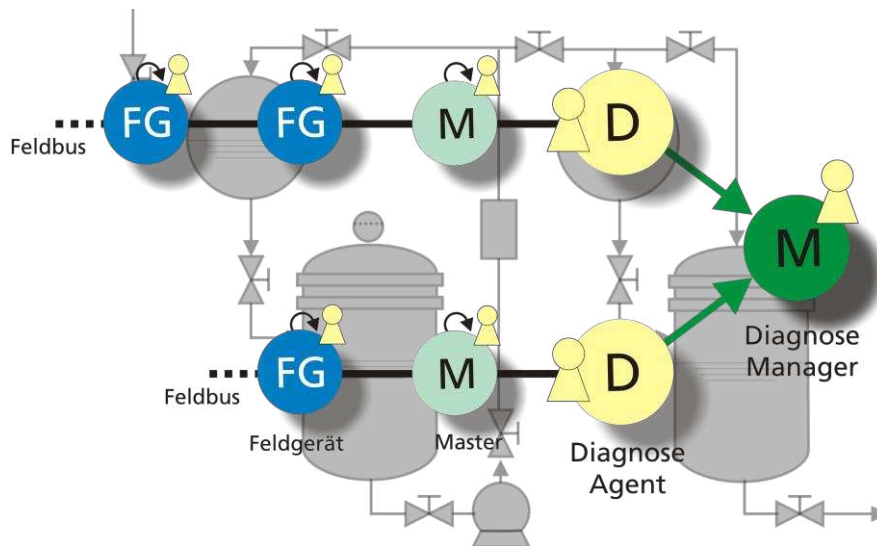


Bild 1: Schema des agentenbasierten Diagnosekonzeptes.

Die Diagnoseagenten, in der untersten Hierarchieebene, sind Software-Komponenten welche auf dedizierten eingebetteten Systemen, den Feldbusgateways, weitestgehend autonom ablaufen. Die Aufgabe der verschiedenen Diagnoseagenten ist es, die Feldbustelegramme der dem Feldbusabschnitt zugeordneten Feldgeräte zu analysieren und aussagekräftige Merkmale zu erzeugen. Der Diagnosemanager in der übergeordneten Ebene ist ebenfalls als Softwareagent realisiert und beinhaltet die eigentliche Diagnosefunktionalität. Realisiert als lauffähige Einzelapplikation bzw. als Komponente des Leitsystems ruft der Diagnosemanager zyklisch die in den Diagnoseagenten der jeweiligen Feldbusstränge generierten Merkmale ab und führt sie der Merkmalauswertungsstufe zu.

Im Hinblick auf die Integration in bestehende Automatisierungsanlagen stellt das entwickelte Konzept weitere Softwareagenten z. B. für die Anbindung an verschiedene physikalische Ausführungen des Feldbusses (Profibus, Foundation Fieldbus), für die Visualisierung der Diagnoseergebnisse (GUIAgent) oder für die Archivierung der Diagnoseergebnisse (ArchiveAgent) bereit.

Zur Realisierung und Umsetzung von Multi-Agenten-Systemen existieren verschiedene Entwicklungsumgebungen, welche dem Entwickler Basisfunktionalitäten zur Entwicklung von Softwareagenten zur Verfügung stellen – für die Implementierung des vorgeschlagenen Diagnosekonzeptes wurde die Entwicklungsumgebung JADE (Java Agent Development Framework) ausgewählt [2,3].

3. Diagnose und Monitoring des Prozessverhaltens

Im Hinblick auf eine ganzheitliche Diagnose industrieller Prozesse stellt der Feldbus als zentrales Kommunikationsnetzwerk der Anlagenkomponenten sowohl Gegenstand der Diagnose in kommunikationstechnischem Sinne dar und zudem bietet der Feldbus den idealen Zugangspunkt für die Diagnose des der Anlage zugrunde liegenden physikalischen Prozessverhaltens. Um eine möglichst transparente und robuste Diagnosefunktionalität zu gewährleisten wird in dem entwickelten Konzept die Diagnoseaufgabe in die Diagnose des Kommunikationsverhaltens des Feldbusses und

die Diagnose bzw. Monitoring des zu Grunde liegenden physikalischen Prozessverhaltens der Anlage unterteilt.

Für die Diagnose des Kommunikationsverhaltens werden aus den Feldbustelegrammen kommunikationsrelevante Merkmale, wie z. B. die Antwortdauer eines Feldgerätes, häufigkeitsbasierte Fuzzy-Zugehörigkeitsfunktionen generiert und wissensbasiert in Form von Expertenregeln (*wenn ... dann ...*) ausgewertet [5]. Eine detaillierte Vorstellung dieses Neuro-Fuzzy basierten Konzeptes wird in [3,4] gegeben.

Im Gegensatz zu der Diagnose der Kommunikationsebene, bei der ausschließlich signalgestützte Merkmalgenerierungsverfahren genutzt werden, stützt sich die Diagnose des Prozessverhaltens der Anlage auf modellbasierte Methoden. Das Grundprinzip der modellgestützten Diagnose technischer Systeme basiert auf einem quantitativen mathematischen Modell des zu überwachenden Prozesses. Durch ein geeignetes Abstandsmaß werden die gemessenen Prozessgrößen mit den berechneten Prozessgrößen verglichen – je größer der Abstand zwischen gemessenen und modellierten Prozessgrößen ist, desto wahrscheinlicher liegt eine Abweichung vom in der Modellierung angenommenen Normalverhalten des Prozesses vor [1,6]. Die Schwierigkeit bei der Realisierung solcher modellbasierten Diagnosesysteme liegt in der Erstellung des Prozessmodells insbesondere wenn analytische Modellansätze zur Verwendung kommen. So ist es z. B. bei verfahrenstechnischen industriellen Prozessen oft nur mit großem Entwicklungsaufwand unter Einbezug von detailliertem Expertenwissen über die physikalischen Zusammenhänge des Prozesses möglich, ein robustes Modell zur Merkmalgenerierung aufzustellen.

Eine Alternative zu den analytisch modellbasierten Methoden bieten datengetriebene lernfähige Modellansätze. Methodisch stehen dabei die künstlichen neuronalen Netze im Vordergrund. Auf Basis von gemessenen Prozessgrößen (Lerndaten) sind neuronale Netze unter Anwendung eines Lernalgorithmus in der Lage, das statische und / oder das dynamische Übertragungsverhalten des Prozesses zu erfassen [7]. Der Vorteil hierbei liegt darin, dass a priori kein analytisch formuliertes Prozessmodell von Nöten ist und somit der Entwickler von „unsicheren“ Annahmen über die physikalischen Zusammenhänge des Prozesses bei der Modellierung entbunden wird. Häufig zum Einsatz kommende Verfahren bei der so genannten Black-Box Modellierung sind Backpropagation Netze. Diese neuronalen Strukturen basieren auf einer überwachten Lernaufgabe, d. h. zum „Trainieren“ solcher Netze müssen die Eingangs- und Ausgangsgrößen des Prozesses bekannt sein.

Im Zusammenhang mit der Diagnose eines unbekanntem physikalischen Verhaltens von feldbusbasierten Automatisierungsanlagen stellt sich jedoch das Problem, dass im Feldbusdatenstrom prinzipiell alle Ein- und Ausgangsgrößen der physikalischen Teilprozesse in den Telegrammen ablesbar sind, jedoch erfordert die Zuordnung hinsichtlich Ein- bzw. Ausgangsgröße wiederum eine sehr detaillierte, aus der Anlagendokumentation abzuleitende, Kenntnis über die zugrundeliegende Anlage. Basierend auf diesen Randbedingungen nutzt das entwickelte Konzept zur Diagnose des unbekanntem physikalischen Verhaltens die Eigenschaften von selbstorganisierenden Karten zur datengetriebenen Modellierung des Prozessverhaltens.

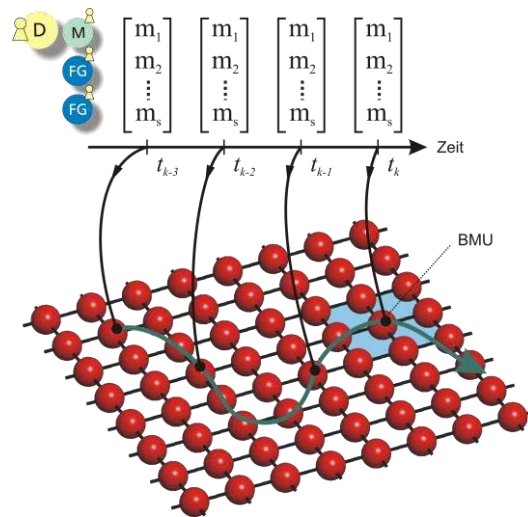


Bild 2: Selbstorganisierende Karte in zweidimensionaler Gittertopologie.

Bei den so genannten selbstorganisierenden Merkmalskarten (SOM) handelt es sich um ein spezielles neuronales Netz, welches geeignet ist, eine topologieerhaltende Abbildung eines hochdimensionalen Merkmalsraumes in einen Ausgaberaum niedrigerer Dimensionalität zu erzeugen [8]. Diese, auch Kohonen-Karten genannten, neuronalen Modelle sind in der Lage, unbekannte Strukturen in der zu analysierenden Datenbasis ohne a priori Informationen „unüberwacht“ zu extrahieren (Strukturentdeckung) und zu visualisieren.

Die Grundbausteine der SOMs werden als Neuronen bezeichnet, sie unterscheiden sich aber in ihrer grundsätzlichen Funktion von dem Neuronenmodell der Backpropagation Netze. Die Neuronen fungieren nicht als Verarbeitungseinheiten, die auf bestimmte Eingaben mit bestimmten Ausgaben reagieren, sondern übernehmen die Funktion von einfachen Speichern, deren Inhalt, gespeicherte Mustervektoren oder auch Prototypen genannt, ausgelesen und wieder neu geschrieben werden können. Die Neuronen der SOMs sind, wie in Bild 2 dargestellt, in einer definierten Topologie angeordnet, d. h. die Neuronen stehen in einer Beziehung zueinander. Diese Qualität des Netzes wird im Allgemeinen mit dem Begriff der Nachbarschaft beschreiben. Selbstorganisierende Karten basieren auf einem unüberwachten, konkurrierenden Lernvorgang - während des Lernvorganges wird dem Netz ein Merkmalvektor M aus der Lernaufgabe präsentiert und dessen Distanz zu den in den Neuronen gespeicherten Mustervektoren W berechnet. Das Neuron mit der geringsten Distanz zum Eingabevektor, die so genannte Best-Matching-Unit (BMU), wird dann nach folgender Gleichung modifiziert:

$$W_j^{k+1} = W_j^k + \eta \mu (M - W_j^k)$$

Neben „Gewinnerneuron“ bezieht die Kohonen-Lernregel zudem noch Nachbarneuronen in den Lernvorgang ein. Durch das Nachbarschaftslernen wird erreicht, dass „ähnliche“ Merkmalsvektoren in topographisch ähnliche Bereiche der Karte projiziert werden. Durch den Nachbarschaftskoeffizienten η kann während des Lernvorganges diese Eigenschaft verstärkt bzw. abgeschwächt werden. Als Maß wie gut die Karte einen Lerndatensatz abbildet, dient der so genannte Quantisierungsfehler,

welcher z. B. mittels Euklidischer Distanz zwischen Merkmaleingabevektor M und Mustervektor W des Gewinnerneurons berechnet wird:

Durch die so genannte UMatrix-Darstellung einer selbstorganisierenden Karte ist es möglich, eine Klassifizierung bzw. ein Clustering des Merkmalsraumes vorzunehmen [7]. Als Beispiel hierfür ist in Bild 5 eine solche UMatrix dargestellt. Die Täler in dieser UMatrix-Darstellung (blau) stellen Bereiche dar, in denen sich die gespeicherten Mustervektoren sehr „stark“ ähneln, die farblich rot dargestellten Bereiche kennzeichnen einen Übergang von einem Merkmalraumcluster zum nächsten Cluster.

Wie bereits angedeutet sind im Datenstrom des Feldbusses, respektive in den Telegrammen der Feldgeräte, prinzipiell alle Prozessgrößen ablesbar und können durch die Diagnoseagenten ohne detailliertes Expertenwissen extrahiert werden. Je nach Komplexität der betrachteten Anlage und der Übertragungsgeschwindigkeit des Feldbusses werden die Zustandsgrößen des Prozesses zeitlich aggregiert und in einem ganzheitlichen Zustandsgrößenvektor M zusammengefasst (vgl. Bild 2). Dieser hochdimensionale Zustandsgrößenvektor beschreibt die Interaktion aller Feldgeräte mit dem Gesamtsystem und stellt die Basis für die modellbasierte Diagnose dar. In der anfänglichen Trainingsphase des Diagnosekonzeptes wird mittels des Merkmalvektors das datengetriebene Modell der Anlage in Form einer SOM erzeugt. Die Dauer des Trainingsvorganges hängt im Wesentlichen von der Komplexität des Prozesses (Anzahl der Feldgeräte) bzw. bei zyklischem Prozessverhalten von der Batchdauer ab. Der Erfolg des Lernvorganges kann anhand des sich einstellenden Quantisierungsfehlers der Karte bewertet werden. Klar ist, dass bei dieser datengestützten Vorgehensweise möglichst alle Betriebszustände der Anlage erfasst werden sollten, um eine robuste Diagnosefunktionalität zu erzielen. Basierend auf dieser trainierten Karte kann dann eine Online-Diagnose des Prozessverhaltens vorgenommen werden. Durch Analyse des Quantisierungsfehlers der Karte können Abweichungen vom eintrainierten Normverhalten erkannt werden und somit auf Fehler in der Anlage geschlossen werden.

4. Erprobung des entwickelten Diagnosekonzeptes

In dem vom BMBF geförderten Verbundprojekt „Feldbus“ wurde zur Erprobung des entwickelten agentenbasierten Diagnosekonzeptes eine verfahrenstechnische Demonstrationsanlage aufgebaut [3].

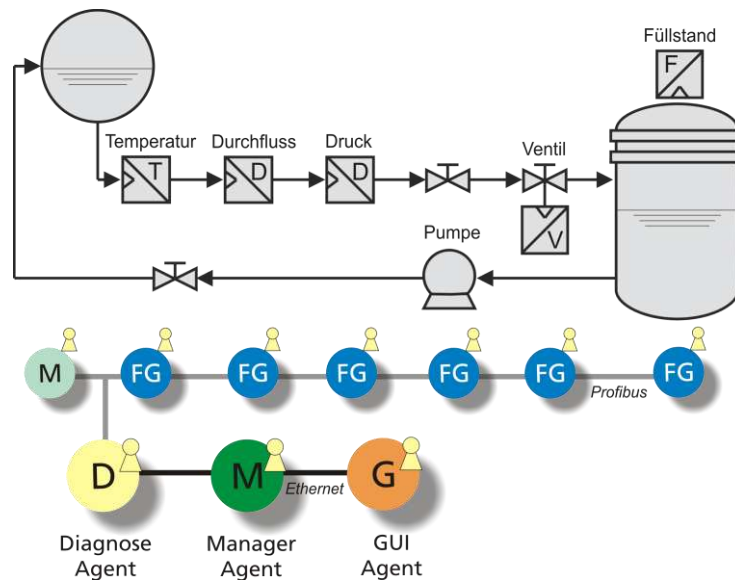


Bild 3: Installierte Demoanlage des Verbundprojektes „Feldbus“.

Die Versuchsanlage, dargestellt in Bild 3, besteht im Wesentlichen aus zwei Behältern, zwischen denen zyklisch Flüssigkeit bei variierender Pumpenleistung und Ventilstellung umgepumpt wird. Angemerkt sei hier, dass im Falle der aufgebauten Demoanlage die Art der Feldgeräte (Mess- bzw. Stellgröße), die physikalische Bedeutung der erfassten Prozessgrößen (z. B. Durchflussmesser, Temperatursensor) und prinzipiell das durch die Steuerung realisierte Prozessverhalten bekannt sind, sich das entwickelte Diagnosekonzept aber nicht auf diese Zusatzinformation stützt.

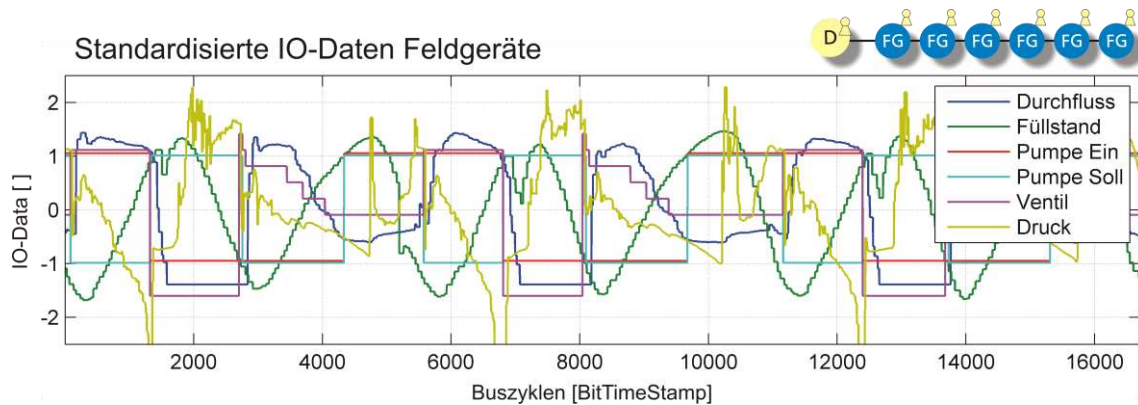


Bild 4: Verlauf der aus den Feldbustelegrammen extrahierten Zustandsgrößen.

Bild 4 zeigt beispielhaft den zeitlichen Verlauf der aus dem Feldbusdatenstrom respektive den Feldbustelegrammen gewonnenen standardisierten Prozessdaten. Anhand des Verlaufes ist augenscheinlich das zyklische sich wiederholende Verhalten des Prozesses zu erkennen. Auf Basis dieser Trainingsdaten wurde ein datengetriebenes Modell der Anlage in Form einer selbstorganisierenden Karte erzeugt. Anhand der UMatrix Darstellung der Karte, dargestellt in Bild 5, sind deutlich die verschiedenen Prozessphasen bzw. Betriebszustände der Anlage (z. B. Entleeren Behälter oder Auffüllen Behälter) zu erkennen, welche sich als blau kolorierte Täler in der 3D Darstellung ausbilden. Die farblich rot dargestellten Bereiche in der UMatrix

Darstellung stellen Übergänge zwischen den verschiedenen Prozessphasen, so genannte transiente Prozessphasen, dar. Basierend auf dieser trainierten Merkmalkarte kann dann eine Online-Diagnose des Prozessverhaltens vorgenommen werden. Dazu werden aus dem Feldbusdatenstrom durch die Diagnoseagenten die IO-Daten der Feldgeräte online herausgefiltert und der Merkmalkarte präsentiert. Durch Analyse des Quantisierungsfehlers der Karte können so Abweichungen vom eintrainierten Normverhalten erkannt werden. Als Beispiel zeigt Bild 5 den Verlauf des Quantisierungsfehlers über der Prozessdauer bei gestörtem Prozessverhalten. Im Fehlerfall 1 wurde zur Störung des Prozessverhaltens die Entlüftung des Systems verringert, im Fehlerfall 2 hingegen der Durchflussquerschnitt verringert. Diese Eingriffe in das Prozessverhalten der Anlage zeigen sich deutlich im Verlauf des Quantisierungsfehlers der Karte.

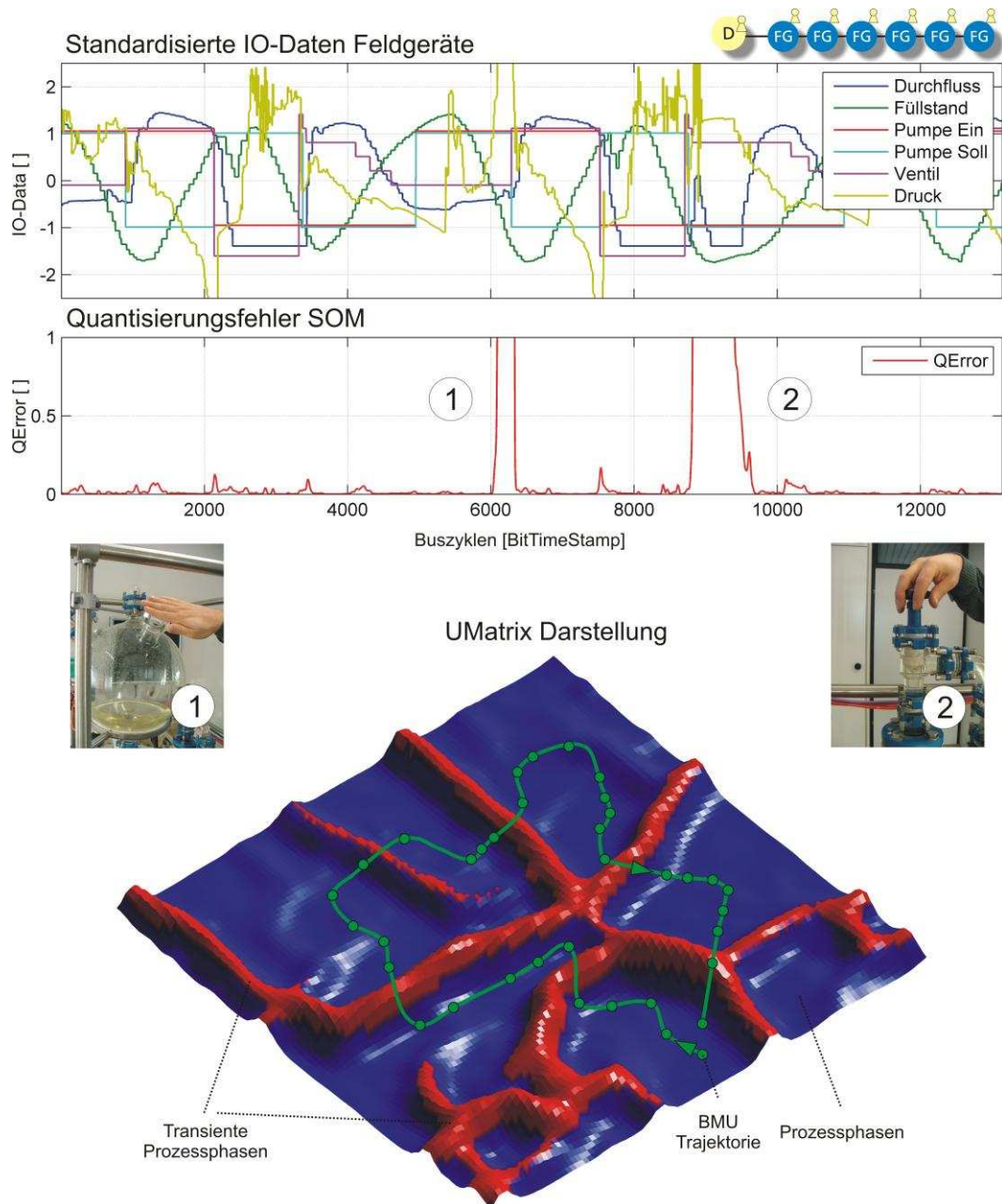


Bild 5: Verlauf des Quantisierungsfehlers der Karte bei gestörtem Prozessverhalten

5. Zusammenfassung

Das entwickelte agentenbasierte Diagnosekonzept stellt eine integrale Komponente der automatisierungstechnischen Anlage dar und ermöglicht eine permanente ganzheitliche Diagnose des kommunikationstechnischen sowie physikalischen Verhaltens der Automatisierungsanlage. Im vorliegenden Beitrag wurde die Architektur des entwickelten verteilten Diagnosekonzepts basierend auf Softwareagenten skizziert sowie die darin eingebettete Funktionalität zur Diagnose des unbekanntes Prozessverhaltens mittels selbstorganisierender Merkmalskarten erläutert. Die Funktionalität und Leistungsfähigkeit des entwickelten Diagnosekonzeptes wurde anhand einer verfahrenstechnischen Demoanlage validiert. Das Konzept wird derzeit auf einen industriellen verfahrenstechnischen Batch-Prozess übertragen.

Literatur

- [1] Frank, P.M.: Diagnoseverfahren in der Automatisierungstechnik. at-Automatisierungstechnik 42 (1994) Nr. 2, S. 47-64.
- [2] Bradshaw, J.M.: Software Agents. AAAI Press, April 1997 ISBN 0-262-52234-9
- [3] Endl, H.: FELDBUS - Erhöhung der Sicherheit und Zuverlässigkeit von eingebetteten, echtzeitfähigen Softwaresystemen durch die Entwicklung von Methoden und Werkzeugen zur agentenbasierten Diagnose von Feldbusnetzen in verteilten Automatisierungssystemen. Schlussbericht BMBF Förderprojekt, Förderkennzeichen 01 IS C 16 A
- [4] Bischoff, T.: Gewinnung statistisch gestützter Zugehörigkeitsfunktionen zur Neuro-Fuzzy Diagnose von feldbusbasierten Automatisierungsanlagen. Diplomarbeit Hochschule für Technik Karlsruhe 07.03.2005
- [5] Nauck, D.; Klawonn, F.; Kruse, R.: Neuronale Netze und Fuzzy Systeme Vieweg Verlag, Braunschweig/Wiesbaden 1994
- [6] Frey, C.; Kuntze, H.-B.: A neuro-fuzzy supervisory control system for industrial batch processes. IEEE Transactions on Fuzzy Systems 9 (2001), 4, S. 570-577
- [7] Haykin, S.: Neural Networks a comprehensive foundation. 2nd Edition. Prentice Hall, Upper Saddle River, New Jersey 1999
- [8] Kohonen, T.: Self-Organizing Maps. 3. Auflage Springer, Berlin, 1995

**Bereits veröffentlicht wurden in der Schriftenreihe des
Instituts für Angewandte Informatik / Automatisierungstechnik im
Universitätsverlag Karlsruhe:**

- Nr. 1: BECK, S.: Ein Konzept zur automatischen Lösung von Entscheidungsproblemen bei Unsicherheit mittels der Theorie der unscharfen Mengen und der Evidenztheorie, 2005
- Nr. 2: MARTIN, J.: Ein Beitrag zur Integration von Sensoren in eine anthropomorphe künstliche Hand mit flexiblen Fluidaktoren, 2004
- Nr. 3: TRACHEL, A.: Neue Verfahren zur Modellierung nichtlinearer thermodynamischer Prozesse in einem Druckbehälter mit siedendem Wasser-Dampf Gemisch bei negativen Drucktransienten, 2005
- Nr. 4: LOOSE, T.: Konzept für eine modellgestützte Diagnostik mittels Data Mining am Beispiel der Bewegungsanalyse, 2004
- Nr. 5: MATTHES, J.: Eine neue Methode zur Quellenlokalisierung auf der Basis räumlich verteilter, punktwiser Konzentrationsmessungen, 2004
- Nr. 6: MIKUT, R.; REISCHL, M.: Proceedings - 14. Workshop Fuzzy-Systeme und Computational Intelligence: Dortmund, 10. - 12. November 2004, 2004
- Nr. 7: ZIPSER, S.: Beitrag zur modellbasierten Regelung von Verbrennungsprozessen, 2004
- Nr. 8: STADLER, A.: Ein Beitrag zur Ableitung regelbasierter Modelle aus Zeitreihen, 2005
- Nr. 9: MIKUT, R.; REISCHL, M.: Proceedings - 15. Workshop Computational Intelligence: Dortmund, 16. - 18. November 2005, 2005
- Nr. 10: BÄR, M.: μ FEMOS - Mikro-Fertigungstechniken für hybride mikrooptische Sensoren, 2005
- Nr. 11: SCHAUDEL, F.: Entropie- und Störungssensitivität als neues Kriterium zum Vergleich verschiedener Entscheidungskalküle, 2006
- Nr. 12: SCHABLOWSKI-TRAUTMANN, M.: Konzept zur Analyse der Lokomotion auf dem Laufband bei inkompletter Querschnittlähmung mit Verfahren der nichtlinearen Dynamik, 2006
- Nr. 13: REISCHL, M.: Ein Verfahren zum automatischen Entwurf von Mensch-Maschine-Schnittstellen am Beispiel myoelektrischer Handprothesen, 2006
- Nr. 14: KOKER, T.: Konzeption und Realisierung einer neuen Prozesskette zur Integration von Kohlenstoff-Nanoröhren über Handhabung in technische Anwendungen, 2007
- Nr. 15: MIKUT, R.; REISCHL, M.: Proceedings - 16. Workshop Computational Intelligence: Dortmund, 29. November - 1. Dezember 2006
- Nr. 16: LI, S.: Entwicklung eines Verfahrens zur Automatisierung der CAD/CAM-Kette in der Einzelfertigung am Beispiel von Mauerwerksteinen, 2007
- Nr. 17: BERGEMANN, M.: Neues mechatronisches System für die Wiederherstellung der Akkommodationsfähigkeit des menschlichen Auges, 2007
- Nr. 18: HEINTZ, R.: Neues Verfahren zur invarianten Objekterkennung und -lokalisierung auf der Basis lokaler Merkmale, 2007
- Nr. 19: RUCHTER, M.: A New Concept for Mobile Environmental Education, 2007

Die Schriften sind als PDF frei verfügbar, eine Nachbestellung der Printversion ist möglich.
Nähere Informationen unter www.uvka.de.

Nähere Informationen zum Fachausschuss finden Sie auf unserer Webseite

<http://www.iai.fzk.de/medtech/biosignal/gma/index.html>

Dort erhalten sie u.a.:

- Informationen zu aktuellen Tagungen und Workshops,
- Zugang zum Tutorial für Industrieanwendungen von Fuzzy-Logik und Fuzzy-Control,
- Verlinkungen zur VDI/VDE-Richtlinie 3550 (Computational Intelligence) mit drei Blättern (unter Publikationen):
 - Blatt 1: Künstliche Neuronale Netze in der Automatisierungstechnik - Begriffe und Definitionen, 2001
 - Blatt 2: Fuzzy-Logik und Fuzzy Control - Begriffe und Definitionen, 2002
 - Blatt 3: Evolutionäre Algorithmen - Begriffe und Definitionen, 2003,
- Möglichkeiten zum Download vieler Volltexte und Präsentationen der Workshopbeiträge (unter Publikationen: komplette Volltexte seit 2000, ausgewählte Beiträge seit 1994) sowie
- Downloadmöglichkeiten von im Fachausschuss entwickelter Software zum Thema Computational Intelligence (z. Zt. MATLAB-Toolboxen Gait-CAD und SPOT).

VDI VEREIN DEUTSCHER INGENIEURE
Bitte klicken Sie auf das Logo um zur Homepage zurück zu gelangen
VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik (GMA)

Fachausschuss 5.14 Computational Intelligence

Ziele der Arbeiten

Der **Fachausschuss** bietet ein Forum zur Diskussion neuer methodischer Ansätze und industrieller Anwendungen auf dem Gebiet der Computational Intelligence. Er entstand 2005 aus den bisherigen Fachausschüssen Neuronale Netze und Evolutionäre Algorithmen (FA 5 21) sowie Fuzzy Control (FA 5 22). Besondere Schwerpunkte sind automatisierungstechnische Anwendungen, z.B. in der Verfahrenstechnik, der Energietechnik, der Automobiltechnik, der Robotik und der Medizintechnik; aber auch Lösungen aus anderen Anwendungsgebieten sind jederzeit von Interesse. Die Ergebnisse werden auf jährlich stattfindenden Workshops präsentiert und in einer offenen Atmosphäre intensiv diskutiert. Dabei ist es gute Tradition, auch neue methodische Ansätze und Ideen bereits in einem frühen Entwicklungsstadium vorzustellen, in dem sie noch nicht vollständig ausgereift sind.

Die Ziele des Fachausschusses sind

- ein Gesprächsforum zwischen Industrie, Hochschulen und Forschungsinstituten zu bieten,
- den Erfahrungsaustausch der Mitglieder zu fördern,
- die industrielle Umsetzung entwickelter Methoden voranzutreiben,
- Tagungen und Workshops zu veranstalten,
- als neutrales Fachgremium Veröffentlichungen und Richtlinien zu erarbeiten und
- die Kommunikation mit anderen Fachgremien (insbesondere GI-FG 1.2.4 Fuzzy-Systeme und Soft Computing) zu koordinieren.

Zur Förderung des wissenschaftlichen Nachwuchses wird jährlich ein Young Author Award vergeben, der Kreativität und Originalität methodischer Ansätze und besonders gelungene Anwendungen prämiert.

Optimiert für MS-IE 7.0 und Mozilla Firefox 2.0.
Letzte Änderung: 11.10.07
Markus Ratsch

Fertig Lokales Intranet 100%

Institut für Angewandte Informatik / Automatisierungstechnik Universität Karlsruhe (TH)

Dieser Tagungsband enthält die Beiträge des 17. Workshops „Computational Intelligence“ des Fachausschusses 5.14 der VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik (GMA) und der Fachgruppe „Fuzzy-Systeme und Soft-Computing“ der Gesellschaft für Informatik (GI), der vom 5.-7. Dezember 2007 im Haus Bommerholz bei Dortmund stattfindet.

Der GMA-Fachausschuss 5.14 „Computational Intelligence“ entstand 2005 aus den bisherigen Fachausschüssen „Neuronale Netze und Evolutionäre Algorithmen“ (FA 5.21) sowie „Fuzzy Control“ (FA 5.22). Der Workshop steht in der Tradition der bisherigen Fuzzy-Workshops, hat aber seinen Fokus in den letzten Jahren schrittweise erweitert.

Die Schwerpunkte sind Methoden, Anwendungen und Tools für

- Fuzzy-Systeme,
- Künstliche Neuronale Netze,
- Evolutionäre Algorithmen und
- Data-Mining-Verfahren

sowie der Methodenvergleich anhand von industriellen und Benchmark-Problemen.

Die Ergebnisse werden von Teilnehmern aus Hochschulen, Forschungseinrichtungen und der Industrie in einer offenen Atmosphäre intensiv diskutiert. Dabei ist es gute Tradition, auch neue Ansätze und Ideen bereits in einem frühen Entwicklungsstadium vorzustellen, in dem sie noch nicht vollständig ausgereift sind.

ISSN: 1614-5267

ISBN: 978-3-86644-191-0

www.uvka.de