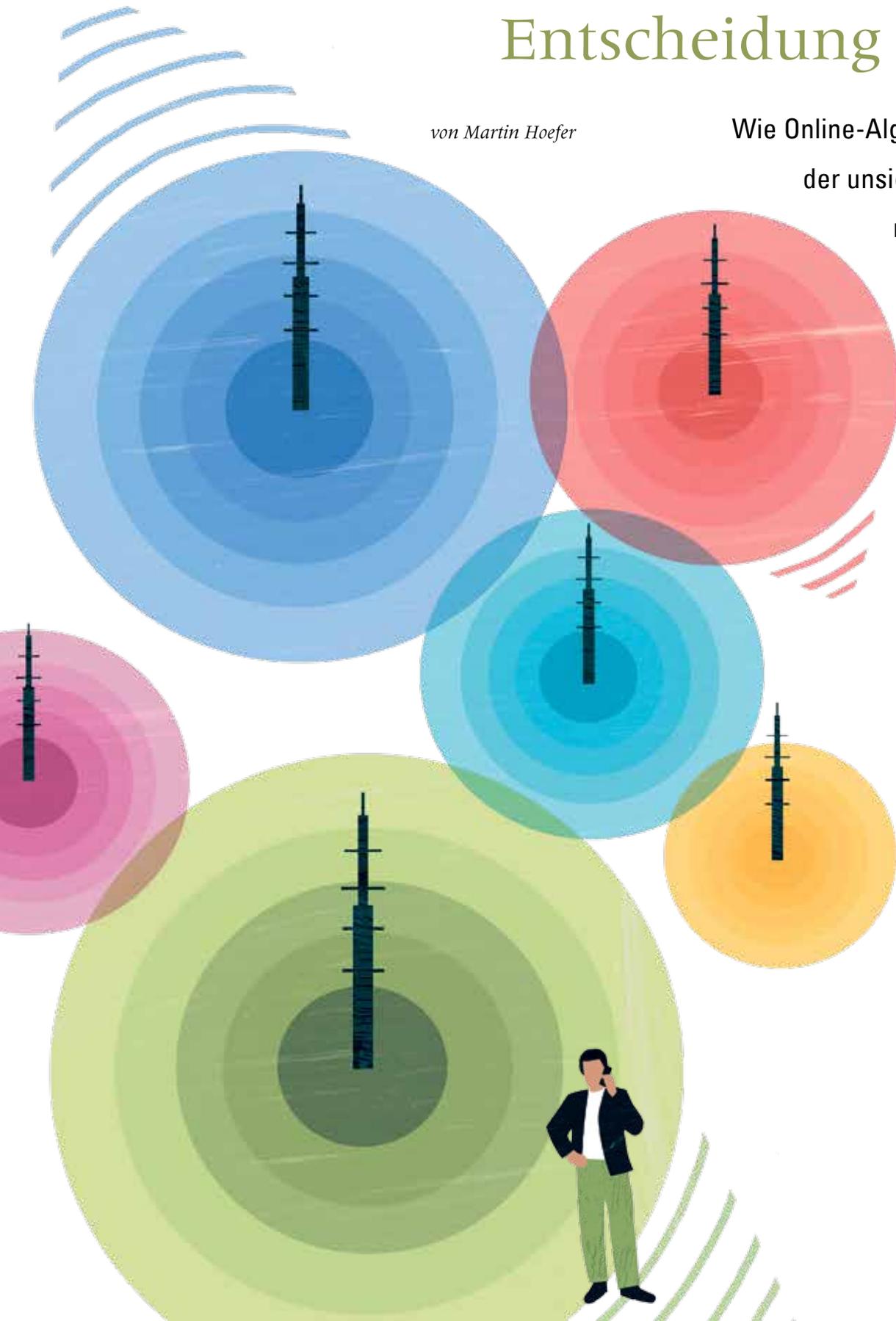


Die wahrscheinlich beste Entscheidung

von *Martin Hoefler*

Wie Online-Algorithmen mit
der unsicheren Zukunft
rechnen



Lohnt es sich, als Skianfänger in einem schneeunsicheren Jahr Skier zu kaufen? Oder ist es günstiger, sie zu mieten? Oft müssen wir Entscheidungen treffen, ohne genügend Informationen über die Zukunft zu haben. Das gilt in noch größerem Maße für Rechnersysteme, die große Datenmengen verarbeiten und schnelle Entscheidungen treffen müssen. Damit sie trotz einer Vielzahl von Unsicherheiten erfolgreich arbeiten können, entwickeln Informatiker Online-Algorithmen.

Rent-or-buy-Probleme sind ein typisches Beispiel für die Aufgabe, Entscheidungen angesichts unsicherer Entwicklungen zu optimieren. Nehmen wir an, Andrea fährt nach Österreich in den Skiurlaub. Sie ist noch nicht sicher, ob sie längerfristig Wintersportlerin bleibt. Da das Skigebiet nicht mehr schneesicher ist, weiß sie auch nicht, wie viele Tage sie Skifahren wird. Skier zu mieten kostet 40 Euro pro Tag; für den Kauf müsste sie 400 Euro investieren. Wie lange sollte sie mieten, wann sollte sie kaufen?

Probleme dieser Art mit unsicheren Eingaben treten in fast allen Bereichen von Rechnersystemen und Optimierung auf, z.B. bei der Steuerung von Echtzeitsystemen, bei Servern im Internet, bei der internen Speicherverwaltung im Rechner oder bei der Interaktion mit Nutzern. Gute Algorithmen mit unsicherer Eingabe zu entwerfen und zu analysieren, ist das Forschungsgebiet der Online-Algorithmen.

Unsicherheit als Nullsummenspiel

Ein einfacher Online-Algorithmus für das Ski-problem besteht darin, direkt am ersten Tag die Skier zu kaufen. Das wäre bei vielen Skitagen optimal, bei wenigen aber viel zu teuer. Wie kann man so einen Online-Algorithmus also bewerten? Eine Antwort gibt die kompetitive Analyse, bei der wir die Lösung des Algorithmus mit einer optimalen Lösung vergleichen, die man mit vollständigem Wissen über die Zukunft erreichen könnte. Da eventuell wenig bis gar kein Wissen über die Zukunft vorliegt, geht man von einem schlimmsten Fall aus und macht aus der Optimierung ein Spiel: Andrea wählt einen Algorithmus A , um zu entscheiden, wie lange sie mietet und wann sie kauft. Ein Gegner wählt danach die Anzahl der Skitage auf die für den Algorithmus A ungünstigste Art. Algorithmus A erzielt für diese Skitage die Kosten c ; die optimale Lösung mit Kenntnis der Skitage hätte dagegen die Kosten c_{opt} . Als kompetitiven Faktor definiert man dann $r = c/c_{opt}$ und gibt an, um welchen Faktor die Kosten von A die optimalen

Kosten im schlimmsten Fall übersteigen. Andrea und der Gegner spielen ein Nullsummenspiel: Je näher r an 1 liegt, desto besser für Andrea und desto schlechter für ihren Gegner.

Mit Algorithmus A muss Andrea in jedem Fall höchstens r -mal so viel zahlen wie der optimale Ansatz, der die Anzahl Skitage kennt. Wenn wir die Anzahl kennen, ist klar was zu tun ist: bei höchstens $400:40=10$ Skitagen werden wir nur mieten, bei mindestens 11 Tagen am allerersten Tag kaufen. A kennt die Skitage nicht und kauft am Anfang von Tag x . Der Gegner legt die Eingabe dann natürlich auf x Skitage fest, damit A kauft, aber nicht vom Kauf profitiert. Die Kosten sind $c = 40(x-1) + 400$, und $c_{opt} = \min(400, 40x)$. Der kompetitive Faktor ist $r = (40(x-1) + 400) / \min(400, 40x)$. Die beste Wahl ist also am Tag $x = 10$ zu kaufen. Damit ergibt sich der Faktor $r = 1.9$. Andrea zahlt damit immer weniger als doppelt so viel wie bei perfekter Kenntnis der Skitage.

Die kompetitive Analyse ist ein attraktiver Ansatz, um Algorithmen zu entwerfen und zu bewerten. Sie liefert mathematisch beweisbare Aussagen bei weitreichenden Unsicherheiten in der Eingabe. Für eine Garantie auf den kompetitiven Faktor nutzen wir sehr wenig Information über die Unsicherheit in der Eingabe und erlauben dem Gegner eine »worst case«-Auswahl bei den möglichen Zukunftsszenarien. Damit ist so eine Garantie äußerst robust – sie bleibt gültig, auch wenn der Algorithmus für ein ähnliches Problem in einer anderen Anwendung zum Einsatz kommt. Z.B. könnte Andrea den gleichen Ansatz nutzen, um zu entscheiden, wie lange sie im Hotel auf den Aufzug warten soll, bevor sie die Treppe nimmt. Sie würde wieder höchstens doppelt so lange bis zu ihrem Zimmer brauchen wie eine optimale Strategie mit Kenntnis über die Zukunft.

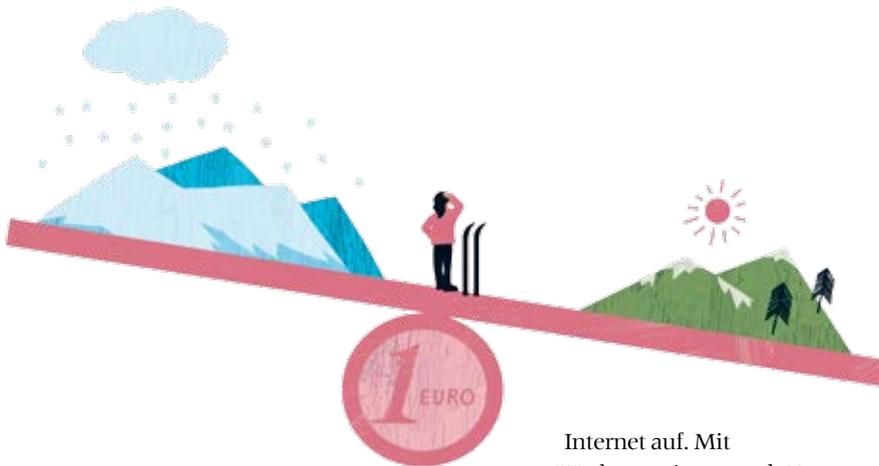
Von Sekretären und stochastischen Unsicherheiten

Online-Algorithmen werden seit über 30 Jahren erforscht. Sie sind aber für wichtige Probleme immer noch nicht gut verstanden und stellen weiterhin eine spannende Herausforderung für die aktuelle Forschung dar. In den letzten Jahren treten Probleme mit unsicherer Eingabe vermehrt bei ökonomischen Anwendungen im



AUF DEN PUNKT GEBRACHT

- Eine Strategie, Algorithmen für eine unsichere Zukunft zu optimieren, besteht darin, den gewählten Algorithmus mit einer optimalen Lösung zu vergleichen. Die Unsicherheiten fließen durch ein Nullsummenspiel mit einem Gegner ein, der den »worst case« des Algorithmus herbeiführt.
- Im »Secretary-Problem«, bei dem man den besten Bewerber auswählen soll, errechnet man anhand einer Stichprobe eine Schranke. Sie liefert einen gut austarierten Mittelwert für eine optimierte Entscheidung.
- Diese und ähnliche Strategien können für eine Vielzahl von wichtigen Problemstellungen genutzt werden. Gute Algorithmen sind aber in vielen Fällen noch unzureichend verstanden.



Internet auf. Mit Werbeanzeigen und Empfehlungssystemen werden heute Umsätze in Milliardenhöhe erzielt, insbesondere (aber nicht nur) von großen IT-Firmen und Suchmaschinen. Grundlegende Problemstellungen in diesem Bereich haben oft eine »Wähle-die-beste«-Eigenschaft, die im folgenden stark vereinfachten Beispiel deutlich wird:

Eine Suchmaschine möchte die Werbeanzeige einer Firma auf ihren Resultatseiten einblenden. Je besser der Suchbegriff zum Profil der Firma passt, desto mehr zahlt die Firma. Sie zahlt dabei höchstens für k Einblendungen, der Einfachheit halber sei hier $k = 1$. Die Suchanfragen kommen nacheinander an. Wir müssen sofort entscheiden, ob wir die Anzeige auf der Resultatseite einblenden oder nicht, bevor die nächste Suchanfrage bearbeitet wird. Das Problem besteht darin, den besten Suchbegriff zu finden und auszuwählen.

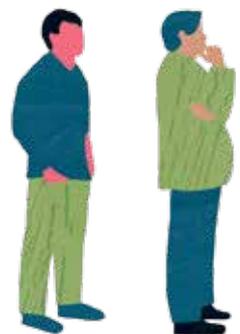
Dieses Problem lässt sich mit der klassischen kompetitiven Analyse nicht zufriedenstellend analysieren. Ein allmächtiger Gegner, der über Suchbegriffe und Ankunftsreihenfolge bestimmt, kann damit für jeden algorithmischen Ansatz leicht eine Eingabe erzeugen, bei der die Lösung des Algorithmus beliebig schlecht ist. Dies gilt in angepasster Form sogar, wenn der Algorithmus seine Wahl zufällig treffen kann und der Gegner die Suchbegriffe nicht auf den Zufall in den Entscheidungen des Algorithmus abstimmen

kann. Gemäß der Analyse sind damit alle möglichen Algorithmen für dieses Problem extrem schlecht. Aber natürlich gibt es auch hier bessere und schlechtere Algorithmen. Die komplett gegnerische Analyse ist nur viel zu pessimistisch und liefert keine brauchbare Unterscheidung.

In der Praxis sind Suchanfragen nicht völlig gegnerisch bestimmt, sondern werden von vielen Parametern beeinflusst, die man relativ gut durch stochastische Annahmen erfassen kann. Selbst wenn die Menge der Anfragen in einem Zeitintervall unbekannt ist, kann man sie durch Erfahrungswerte oft relativ genau einschätzen. Außerdem kommen die Anfragen selten in gegnerischer Reihenfolge an. Im sogenannten »Secretary-Modell« sind Anfragen und Profite gegnerisch bestimmt, ihre Anzahl n ist aber bekannt, und die Ankunftsreihenfolge wird als gleichmäßig zufällig angenommen. Im Grunde ergibt sich damit das klassische »Secretary-Problem«, den besten Bewerber für eine Stelle aus einer Menge von n Bewerbern zu finden. Die Bewerber kommen nacheinander in zufälliger Reihenfolge an, und man erfährt ihre Eignung erst, wenn sie eintreffen. Jeder Bewerber muss endgültig akzeptiert oder abgelehnt werden, bevor weitere Bewerber interviewt werden. Sobald ein Bewerber akzeptiert wird, kommen keine weiteren Bewerber mehr an.

Die optimale Lösung für diesen Ansatz ist ein einfaches Lernverfahren: Die Lernphase besteht aus den ersten n/e Anfragen/Bewerbern. Alle Anfragen in der Lernphase werden abgelehnt. Es sei P der maximale Profit einer Anfrage in der Lernphase. Akzeptiere von den verbleibenden Anfragen die erste, die mehr Profit als P liefert. Die Wahrscheinlichkeit, dass wir die Anfrage mit maximalem Profit auswählen, geht gegen $1/e \approx 36.79$ Prozent, wenn n groß wird. Im Sinne der kompetitiven Analyse erhält man somit einen sehr einfachen Algorithmus mit kompetitivem Faktor nahe $1/e$.

Hier wird mit stochastischem Teilwissen über die Zukunft eine Auswahlsschranke berechnet. Damit wird ein grundlegender Zielkonflikt gelöst, der sich aus der verbleibenden Unsicherheit über die Eingabe ergibt. Ist P zu niedrig, wählen wir sehr schlechte Anfragen und erhalten kaum Profit. Ist P zu groß, warten wir nur auf sehr gute Anfragen, die aber eventuell nie eintreffen, und erhalten dann gar keinen Profit. Die Wahl von P ist ein sorgfältig austarierter Mittelwert, der diese Extreme erfolgreich vermeidet. Im Erwartungswert über die zufällige



Der Autor

Prof. Dr. Martin Hoefer, Jahrgang 1978, ist Professor für Algorithmen und Komplexität. Seine Forschungsschwerpunkte sind Entwurf und Analyse von Algorithmen, Algorithmische Spieltheorie und Optimierung unter Unsicherheit. Sein Ziel ist dabei, durch geeignete Algorithmen das Beste aus unsicheren und komplexen Situationen herauszuholen, insbesondere bei Unsicherheit über die Zukunft oder in Systemen mit vielen Teilnehmern und Wettbewerb.

mhoefer@cs.uni-frankfurt.de

Ankunftsreihenfolge wird dadurch ein sehr viel besserer Profit erzielt.

Online-Algorithmen in der Praxis

Erweiterungen des oben skizzierten Lösungsansatzes finden in der aktuellen Forschung eine Vielzahl von Anwendungen. Dabei werden neben dem Secretary-Modell noch eine Reihe weiterer Varianten untersucht (z.B. IID-, Prophet-Inequality- oder Prophet-Secretary-Modelle), die unterschiedliche Mischungen aus gegnerischen und stochastischen Unsicherheiten über die Zukunft erfassen. Eine offensichtliche Erweiterung der Problemstellung sind mehrere Firmen, maximal $k > 1$ Einblendungen pro Firma und maximal $l > 1$ Einblendungen pro Suchanfrage. Dabei wird jede Firma und jede Suchanfrage als Knoten in einem Netzwerk aufgefasst. Eine Kante zwischen Firma und Anfrage stellt eine Einblendung der Anzeige dar und liefert als Wert den entsprechenden Profit. Gesucht ist ein gutes Matching, d.h. möglichst viele wertvolle Kanten, wobei für jeden Firmenknoten nur bis zu k und für jeden Anfrageknoten bis zu l Kanten anliegen dürfen. Dafür sind mittlerweile viele Algorithmen mit sehr guten kompetitiven Faktoren in diversen Modellen bekannt [1, 2, 4, 8], die auch erfolgreich in der Praxis eingesetzt werden.

Eine weitere Anwendung ist die Nutzung von Frequenzspektren in Funknetzwerken. Hier treffen Kommunikationsanfragen nacheinander über die Zeit ein und müssen unmittelbar akzeptiert oder abgelehnt werden. Akzeptierte Anfragen müssen konfliktfrei bleiben, d.h., die Störsignale von anderen akzeptierten Anfragen dürfen nicht zu groß werden. Dabei wird jede

Anfrage als ein Knoten in einem gerichteten Netzwerk aufgefasst. Eine Kante gibt das Ausmaß der Störung an, die die Geräte bei gleichzeitiger Nutzung des Kanals aufeinander ausüben. Gesucht ist ein gutes Independent Set, d.h. möglichst viele Anfragen, so dass für jede gewählte Anfrage das Gewicht der eingehenden Kanten von anderen gewählten Anfragen nicht zu groß wird. Hierfür haben wir Online-Algorithmen in einem allgemeinen stochastisch-gegnerischen Eingabemodell entwickelt. Die guten kompetitiven Faktoren orientieren sich an Modellparametern der Signalübertragung [6].

Verfahren dieser Art können für eine Vielzahl von Auswahlproblemen mit »Wähle-die-beste(n)«-Struktur genutzt werden (z.B. Auswahl von Kauf- oder Verkaufsangeboten in Märkten, Bewerberauswahl für Jobs, Partnerwahl auf Dating-Portalen). In diesen Anwendungen ergeben sich viele zusätzliche Aspekte, z.B. mehrere Verkäufer, die um die nacheinander eintreffenden Käufer konkurrieren [3]; Bewerber, die nicht unmittelbar akzeptiert oder abgelehnt werden müssen [7]; oder Experten, die nur im Rahmen von zeitlich begrenzten Projekten eingestellt werden [5]. Für Probleme mit diesen Eigenschaften existieren erst seit Kurzem Online-Algorithmen im Secretary-Modell mit beweisbaren Garantien für die kompetitiven Faktoren.

Es gibt noch viele weitere wichtige Modellaspekte, die bisher gar nicht oder nur sehr unzureichend verstanden sind. Für die aktuelle und zukünftige Forschung in der Algorithmik ist dies eine gute Nachricht – es gibt noch viele neue, spannende Online-Algorithmen, die nur darauf warten, von enthusiastischen Forschern entdeckt und analysiert zu werden. ●

Literatur

- 1 S. Alaei, M. Hajiaghayi, V. Liaghat: Online Prophet-Inequality Matching with Applications to Ad Allocation, in: Proc. EC, 2012, S. 18-35.
- 2 Y. Chen, P. Berkhin, B. Anderson, N. Devanur: Real-Time Bidding Algorithms for Performance-Based Display Ad Allocation, in: Proc. KDD, 2011, S. 1307-1315.
- 3 N. Chen, M. Hoefer, M. Künnemann, C. Lin, P. Miao: Secretary Markets with Local Information, Distributed Computing, zur Veröffentlichung angenommen, 2018.
- 4 N. Devanur, T. Hayes: The Adwords Problem: Online Keyword Matching with Budgeted Bidders under Random Permutations, in: Proc. EC, 2009, S. 71-78.
- 5 A. Fiat, I. Gurel, H. Kaplan, S. Novgorodov: The Temp Secretary Problem, in: Proc. ESA, 2015, S. 631-642.
- 6 O. Göbel, M. Hoefer, T. Kesselheim, T. Schleiden, B. Vöcking: Online Independent Set Beyond the Worst-Case: Secretaries, Prophets, and Periods, in: Proc. ICALP, 2014, S. 508-519.
- 7 M. Hoefer, L. Wilhelm: Packing Returning Secretaries, in: Proc. ISAAC, 2018.
- 8 T. Kesselheim, K. Radke, A. Tönnis, B. Vöcking: An Optimal Algorithm for Weighted Bipartite Matching and Extensions to Combinatorial Auctions, in: Proc. ESA, 2013, S. 589-600.

