Johann Wolfgang Goethe–Universität
Frankfurt am Main

Kettenbruchentwicklungen in beliebiger Dimension, Stabilität und Approximation

von

Carsten Rössner
aus Frankfurt am Main

Dissertation
zur Erlangung des Doktorgrades
der Naturwissenschaften

Fachbereich Mathematik

Kettenbruchentwicklungen in beliebiger Dimension, Stabilität und Approximation

•

Dissertation
zur Erlangung des Doktorgrades
der Naturwissenschaften

•

vorgelegt beim Fachbereich Mathematik
der Johann Wolfgang Goethe–Universität

in Frankfurt am Main

von

CARSTEN RÖSSNER[1]

aus Frankfurt am Main

Frankfurt am Main 1996
(D F 1)

[1] e-mail: roessner@cs.uni-frankfurt.de, URL: http://www.uni-frankfurt.de/~roessner/homepage

vom Fachbereich Mathematik der

Johann Wolfgang Goethe — Universität als Dissertation angenommen.

Dekan : Professor Dr. Götz Kersting

Gutachter : Professor Dr. Claus Peter Schnorr, Professor Dr. Johannes Buchmann

Datum der Disputation : 5. Juli 1996

Für Carmen,
meine Eltern,
meine Großeltern
und Ihn

'That day, for no particular reason, I decided to go for a little run. So, I ran to the end of the road, and when I got there, I thought maybe I'd run to the end of the town. And when I got there, I thought maybe I'd just run across Greenbow County. And I figured since I'd run this far, maybe I'd just run across the great state Alabama, and that's what I did. I ran clear 'cross Alabama. For no particular reason I just kept on going. I ran clear to the ocean. And when I got there, I figured since I'd gone this far, I might as well turn around just keep on going. And when I got to another ocean, I figured since I'd gone this far, I might as well just turn back keep right on going. When I got tired I slept. When I got hungry I ate. When I had to go, you know, I went.' 'And so, you just ran !?' 'Yeah !'

Paramount Pictures, 'Forrest Gump'

Zusammenfassung

Wir behandeln Kettenbruchentwicklungen in beliebiger Dimension. Wir geben einen Kettenbruchalgorithmus an, der für beliebige Dimension n simultane diophantische Approximationen berechnet, die bis auf den Faktor $2^{(n+2)/4}$ optimal sind. Für einen reellen Eingabevektor $x := (x_1, \ldots, x_{n-1}, 1)$ berechnet der Algorithmus eine Folge ganzzahliger Vektoren $((p_1^{(k)}, \ldots, p_{n-1}^{(k)}, q^{(k)}))_{k \in \mathbb{N}}$, so daß für $i = 1, \ldots, n-1 \;:\; |q^{(k)} x_i - p_i^{(k)}| \leq 2^{(n+2)/4} \sqrt{1 + x_i^2} / q^{\frac{1}{n-1}}$. Nach Sätzen von Dirichlet und Borel ist die Schranke optimal in dem Sinne, als daß der Exponent $\frac{1}{n-1}$ im allgemeinen nicht erhöht werden kann. Der Algorithmus konstruiert eine Folge von Gitterbasen des $\mathbb{Z}^n$, welche die Gerade $x\,\mathbb{R}$ approximieren. Für gegebenes $\epsilon > 0$ findet der Algorithmus entweder eine *Relation* zu x, das heißt einen ganzzahligen zu x orthogonalen Vektor (ungleich Null), mit euklidischer Länge kleiner oder gleich $2^{n/2-1}\,\epsilon^{-1}$, oder er schließt Relationen zu x mit euklidischer Länge kleiner als ϵ^{-1} aus. Der Algorithmus führt in der Dimension n und $|\log \epsilon|$ polynomial viele arithmetische Operationen auf rellen Zahlen in exakter Arithmetik aus. Für rationale Eingaben $x := (p_1, \ldots, p_n)/p_n$, $\epsilon > 0$ mit $p_1, \ldots, p_n \in \mathbb{Z}$ besitzt der Algorithmus polynomiale Bitkomplexität in $O(\sum_{i=1}^{n} \lceil \log |p_i| \rceil + \lceil |\log \epsilon| \rceil)$.

Eine Variante dieses Algorithmus konstruiert für Eingabevektoren x einen (von x nicht notwendigerweise verschiedenen) Nahebeipunkt x' zu x und eine kurze Relation zu x'. Im Falle $x' \neq x$ können wir die Existenz von Relationen kleiner als $(2\,\epsilon)^{-1}$ für Punkte in einer kleinen offenen Umgebung um x' ausschließen. Wir erhalten in diesem Sinne eine stetige untere Schranke für die Länge der kürzesten Relation zu Punkten in dieser Umgebung. Die für x' berechnete Relation ist bis auf einen in der Dimension n exponentiellen Faktor kürzeste Relation für x'.

Zur Implementierung des Kettenbruchalgorithmus stellen wir ein numerisch stabiles Verfahren vor und berichten über experimentelle Ergebnisse.

Wir geben untere Schranken für die Approximierbarkeit kürzester Relationen in der Maximum–Norm und minimaler diophantischer Approximationen an:
Unter der Annahme, daß die Klasse **NP** nicht in der deterministischen Zeitklasse $O(n^{poly \log n})$ enthalten ist, zeigen wir: Es existiert kein Algorithmus, der für rationale Eingabevektoren x polynomial in der Bitlänge $bin(x)$ von x ist und die in der Maximum–Norm kürzeste Relation bis auf einen Faktor $2^{\log^{0.5-\zeta} bin(x)}$ approximiert. Dabei ist ζ eine beliebig kleine positive Konstante.
Wir übertragen dieses Resultat auf das Problem, zu gegebenen rationalen Zahlen $x_1, \ldots, x_{n-1}$ und einem rationalen $\epsilon > 0$ gute simultane diophantische Approximationen zu finden, das heißt rationale Zahlen $\frac{p_1}{q}, \ldots, \frac{p_{n-1}}{q}$ mit möglichst kleinem Hauptnenner q zu konstruieren, so daß $\max_{1 \leq i \leq n-1} |q\,x_i - p_i| \leq \epsilon$. Wir zeigen unter obiger Annahme, daß kein Algorithmus existiert, der für gegebene rationale Zahlen $x_1, \ldots, x_{n-1}$ und natürlicher Zahl N polynomial–Zeit in der Bitlänge $bin(x)$ von x ist und simultane diophantische Approximationen berechnet, so daß $\max_{1 \leq i \leq n-1} |q\,x_i - p_i|$ für $q \in [1, N]$ bis auf den Faktor $2^{\log^{0.5-\zeta} bin(x)}$ minimal ist. Hierbei ist ζ wieder eine beliebig kleine positive Konstante.

Inhaltsverzeichnis

Einleitung

Der Kettenbruchalgorithmus berechnet zur reellen Zahl x eine Folge von Approximationen $\frac{p}{q}$ mit teilerfremden ganzen Zahlen p, q, die sogar *Bestapproximationen* sind. Der Kettenbruchalgorithmus bricht genau dann ab, wenn x rational ist. In diesem Fall ist $(-q, p)$ ein zu $(x, 1)$ orthogonaler ganzzahliger Vektor.

Der Kettenbruchalgorithmus löst somit folgende Probleme in Dimension $n = 2$:

1. Approximiere gegebene reelle Zahlen $x_1, \ldots, x_{n-1}$ durch rationale Zahlen $\frac{p_1}{q}, \ldots, \frac{p_{n-1}}{q}$.
2. Falls $x_1, \ldots, x_{n-1}, 1$ über $\mathbb{Z}$ linear abhängig sind, finde eine kurze ganzzahlige *Relation* $(m_1, \ldots, m_n)$ zu $(x_1, \ldots, x_{n-1}, 1)$ mit $\sum_{i=1}^{n-1} m_i\, x_i + m_n = 0$.

In dieser Arbeit behandeln wir Algorithmen, die diese beiden Probleme für beliebige Dimensionen n lösen. Jacobi [Ja1868], Poincaré [Po1884], Minkowski [Mi05], Perron [Pe07], Bernstein [Bern71], Brun [Bru20], Szekeres [Sz70], unter vielen anderen, übertrugen den 2–dimensionalen Kettenbruchalgorithmus auf höhere Dimensionen. Ziel der als *Jakobi–Perron–Algorithmen* bekannten Verallgemeinerungen des 2–dimensionalen Kettenbruchalgorithmus, ist es, reelle Zahlen $x_1, \ldots, x_{n-1}$ durch rationale Zahlen zu approximieren oder eine Relation für $x_1, \ldots, x_{n-1}, 1$ zu finden. Jakobi–Perron Algorithmen approximieren für Eingaben $x_1, \ldots, x_{n-1} \in \mathbb{R}$ die Gerade $x\,\mathbb{R}$ mit $x := (x_1, \ldots, x_{n-1}, 1)$ durch Folgen von Gitterbasen des Gitters $\mathbb{Z}^n$. Sie terminieren, sobald sie eine Relation für x gefunden haben. Für einige der Jakobi–Perron–Algorithmen gibt es jedoch Beispiele reeller Eingabevektoren $x := (x_1, \ldots, x_{n-1}, 1)$, deren Koordinaten über $\mathbb{Z}$ linear abhängig sind und für die der Algorithmus keine Relation findet [Bru20, Berg80, FF82]. Die von Jakob–Perron–Algorithmen konstruierten Folgen von Gitterbasen des $\mathbb{Z}^n$ approximieren die Gerade $x\,\mathbb{R}$ in dem Sinne, daß die Winkel zwischen der Geraden $x\,\mathbb{R}$ und allen Basisvektoren des $\mathbb{Z}^n$ gegen 0 konvergieren. Man spricht in diesem Falle auch von *schwacher Konvergenz.* Perron führte für Folgen von Basen des Gitters $\mathbb{Z}^n$, welche gegen eine Gerade $x\,\mathbb{R}$ konvergieren, die *starke* bzw. *ideale Konvergenz* ein. Die Abstände der Basisvektoren ideal konvergenter Folgen von Gitterbasen des $\mathbb{Z}^n$ zur Geraden $x\,\mathbb{R}$ konvergieren gegen 0; dies bedeutet, daß die zu x orthogonalen Projektionen der Basisvektoren des $\mathbb{Z}^n$ beliebig klein werden. Algorithmen, für die die zu x orthogonalen Projektionen der Basisvektoren des $\mathbb{Z}^n$ beliebig klein werden, finden stets Relationen zu x, sofern solche existieren. Sind die Koordinaten von x linear unabhängig über $\mathbb{Z}$, konstruieren solche Algorithmen eine ideal konvergente Folge von Gitterbasen des $\mathbb{Z}^n$ an die Gerade $x\,\mathbb{R}$. Dirichlet bewies die Existenz von guten (ganzzahligen) Approximationen: Es gibt unendlich viele ganzzahlige Vektoren

$b =: (p_1, \ldots, p_{n-1}, q)$, die $\max_{1 \le i \le n-1} |q\, x_i - p_i| < 1/q^{\frac{1}{n-1}}$ erfüllen. Die von einem Approximationsalgorithmus konstruierte Folge von Gitterbasen des $\mathbb{Z}^n$ an die Gerade $x\,\mathbb{R}$ ist genau dann ideal konvergent, wenn in der Folge der Approximationen $(p_1, \ldots, p_{n-1}, q)$ $\max_{1 \le i \le n-1} |q\, x_i - p_i|$ gegen 0 konvergiert.

Ferguson und Forcade [FF79] gaben 1979 als erste einen Approximationsalgorithmus an, der für über $\mathbb{Z}$ linear abhängige Eingaben $x_1, \ldots, x_{n-1}, 1$ stets mit einer Relation für $(x_1, \ldots, x_{n-1}, 1)$ anhält. Sind die Eingaben $x_1, \ldots, x_{n-1}, 1$ über $\mathbb{Z}$ linear unabhängig, so konstruiert der Algorithmus von Ferguson und Forcade eine ideal konvergente Folge von Gitterbasen des $\mathbb{Z}^n$ an die Gerade $(x_1, \ldots, x_{n-1}, 1)\,\mathbb{R}$. Bergman [Berg80] stellte 1980 eine iterative Version des rekursiven Algorithmus von Ferguson und Forcade vor. Von diesem Verfahren, dem Bergman–Ferguson–Algorithmus, bewiesen Bergman und Ferguson eine in der Dimension n der reellen Eingaben exponentielle Rechenzeitschranke [FF82]. Hastad, Just, Lagarias und Schnorr [HJLS89] formulierten 1989 den Bergman–Ferguson–Algorithmus in der Sprache der Reduktionstheorie von Gitterbasen nach Lenstra, Lenstra, Lovász [LLL82] und analysierten seine Laufzeit. Der nach [HJLS89] benannte HJLS–Algorithmus findet für Eingaben $x_1, \ldots, x_{n-1}, 1 \in \mathbb{R}$ und $\epsilon > 0$ in polynomialer Zeit in der Dimension n und $|\log \epsilon|$ entweder eine Relation für $(x_1, \ldots, x_{n-1}, 1)$ mit euklidischer Länge kleiner als $2^{n/2}\, \epsilon^{-1}$ oder schließt die Existenz einer Relation kürzer als ϵ^{-1} aus. Just [Ju92] erkannte 1992, daß der Bergman–Ferguson–Algorithmus eine Folge von (ganzzahligen) Approximationen $b =: (p_1, \ldots, p_{n-1}, q)$ an die reellen Zahlen $x_1, \ldots, x_{n-1}$ berechnet, die $\max_{1 \le i \le n-1} |q\, x_i - p_i| \le 2^{(n+2)/4} \max_{1 \le i \le n-1} \sqrt{1 + x_i^2}/|q|^{1+\frac{1}{2\,n\,(n-1)}}$ erfüllen. Die Approximation b ist jeweils der erste Basisvektor in der Folge der konstruierten Gitterbasen des $\mathbb{Z}^n$ jeweils vor Austauschen der letzten beiden Basisvektoren. Just gab hierzu ein Verfahren an, bei dem vor Austauschen der letzten beiden Basisvektoren des $\mathbb{Z}^n$ die zum Eingabevektor $(x_1, \ldots, x_{n-1}, 1)$ orthogonalen Projektionen der Basisvektoren des $\mathbb{Z}^n$ im Sinne von Lenstra, Lenstra und Lovász [LLL82] reduziert sind.

Hauptergebnis der vorliegenden Arbeit ist eine verbesserte Analyse des Relationen– und Kettenbruchalgorithmus von Just und dessen Implementierung. Wir zeigen, daß dieser Algorithmus Approximationen liefert, die bis auf einen nur von der Dimension abhängigen Faktor die Schranke der Existenzaussage von Dirichlet erreichen. Die verbesserte Analyse geht von den dualen Basen aus.

Approximiert eine Folge von Gitterbasen die Gerade $x\,\mathbb{R} \subseteq \mathbb{R}^n$, so approximiert die Folge der dualen Gitterbasen die $(n-1)$–dimensionale Hyperebene $(x\,\mathbb{R})^\perp$. Der entscheidende Punkt der verbesserten Analyse ist, daß die Längen der dualen Basisvektoren des Gitters $\mathbb{Z}^n$ im Relationen– und Kettenbruchalgorithmus von Just sehr klein bleiben. Die verbesserte Schranke für die Länge der dualen Basisvektoren verbessert auch die Schranke für die primären Basisvektoren. Die für die obere Schranke der primären Basisvektoren entscheidende Rolle der dualen Basisvektoren wird zum ersten Mal deutlich. Für die Folge der Approximationen $(p_1, \ldots, p_{n-1}, q)$ an reelle Zahlen $x_1, \ldots, x_{n-1}$ erhalten wir als obere Schranke für $\max_{1 \le i \le n-1} |q\, x_i - p_i|$ bis auf den Faktor $2^{(n+2)/4} \max_{1 \le i \le n-1} \sqrt{1 + x_i^2}$ die Schranke des Existenzsatzes von Dirichlet.

Der Relationenalgorithmus von [HJLS89] führt arithmetische Operationen auf reellen Zahlen in exakter reeller Arithmetik aus. Computer erhalten als Eingaben jedoch nur rationale Approximationen von reellen Zahlen. Es stellt sich die Frage, ob die Nichtexistenz kurzer Relationen für rationale Approximationen an reelle Eingabevektoren x die Nichtexistenz kurzer Relationen für x beweist. Buchmann und Kessler [BK93] behandelten dieses Problem unter der Annahme, daß eine untere Schranke für die Länge der kürzesten Relation zum Eingabevektor x bekannt ist. Wir ändern den Relationenalgorithmus von Just dahingehend ab, daß der Algorithmus für rationale n–dimensionale Eingabevektoren x einen (von x nicht notwendigerweise verschiedenen) Nahebeipunkt x' zu x und eine kurze Relation zu x' konstuiert. Diese Modifikation stammt aus [RSc95]. Im Falle $x' \neq x$ läßt sich die Existenz sehr kurzer Relationen für Punkte in einer kleinen offenen Umgebung um x' ausschließen. Wir zeigen in diesem Sinne eine stetige untere Schranke für die kürzeste Relation zu Punkten in dieser Umgebung. Die für x' berechnete Relation ist bis auf einen in der Dimension n exponentiellen Faktor kürzeste Relation für x'.

Bei der Implementierung des Relationenalgorithmus von Just treten numerische Stabilitätsprobleme auf. Die im Relationenalgorithmus von Just konstruierte, zu $x\,\mathbb{R}$ ideal konvergente Folge von Gitterbasisvektoren des $\mathbb{Z}^n$ ist durch eine Folge elementarer Basistransformationen gegeben. Eine elementare Basistransformation überführt eine Gitterbasis des $\mathbb{Z}^n$ in eine andere, indem entweder zwei Basisvektoren vertauscht werden oder ein ganzzahliges Vielfaches eines Basisvektors zu einem anderen Basisvektor addiert wird. Für die Berechnung jeder neuen elementaren Basistransformation wird im Relationenalgorithmus von Just das linear abhängige System des Eingabevektors x und der aktuellen Basisvektoren des $\mathbb{Z}^n$ orthogonalisiert und die Quadratlängen der Höhenvektoren des Orthogonalsystems berechnet. Wegen der linearen Abhängigkeit der zu x orthogonalen Projektionen der aktuellen Basisvektoren des $\mathbb{Z}^n$ können die Höhenvektoren des berechneten Orthogonalsystems beliebig klein werden. Bei der Implementierung des Relationenalgorithmus müssen daher die Quadratlängen der Höhenvektoren derart genau approximiert werden, daß das Computerprogramm mit den rationalen Approximationen der aktuellen Quadratlängen der Höhenvektoren die neue elementare Basistransformation noch korrekt berechnet. Das Verfahren der Gram–Schmidt Orthogonalisierung ist numerisch zu instabil. Zur Implementierung des Relationenalgorithmus führen wir die Orthogonalisierung mit der Givens Rotation durch. Beim Verfahren der Givens Rotation ist der numerische Fehler im Gegensatz zur Gram–Schmidt Orthogonalisierung unabhängig von der Länge der Höhenvektoren des Orthogonalsystems. Heckler, Thiele [HT93] und Joux [Jo93] verwendeten das Verfahren der Givens Rotation für die Orthogonalisierung der Gitterbasis schon in der Implementierung der parallelen Versionen des L^3–Gitterbasenreduktionsalgorithmus. Ferguson und Bailey [FB92] benutzten 1992 in Verbindung mit der Implementierung des HJLS–Algorithmus ein numerisch stabiles, der Givens Rotation verwandtes Verfahren, die Householder Reflektion.

Wir beweisen die numerische Stabilität des implementierten Relationenalgorithmus für Rechnerarchitekturen bei fester Genauigkeit der Gleitpunktarithmetik. Die praktischen Versuche zeigen, daß der implementierte Relationenalgorithmus bei 53–Bit genauer Gleitpunktarithmetik bis Dimension 120 auf 45–Bit langen zufälligen Eingabevektoren numerisch stabil arbeitet.

Die letzten beiden Kapitel behandeln die Resultate der Arbeiten [RSe96a] und [RSe96b]:

Der HJLS–Algorithmus approximiert die euklidische Länge der kürzesten Relation für einen n–dimensionalen Vektor x in polynomialer Zeit in der Eingabelänge bis auf einen in n exponentiellen Faktor oder zeigt, daß keine Relation mit kurzer euklidischer Länge zu x existiert. Wir untersuchen die Frage, bis auf welchen Faktor die Länge der kürzesten Relation für rationale Eingabevektoren in polynomialer Zeit in der binären Eingabelänge approximiert werden kann. Unter der Annahme, daß die Klasse **NP** nicht in der deterministischen Zeitklasse $O(n^{poly \log n})$ enthalten ist, zeigen wir: Es existiert kein Algorithmus, der für rationale Eingabevektoren x polynomial in der Bitlänge $bin(x)$ von x ist und die in der Maximum–Norm kürzeste Relation bis auf einen Faktor $2^{\log^{0.5-\zeta} bin(x)}$ approximiert. Dabei ist ζ eine beliebig kleine positive Konstante.

Wir übertragen dieses Resultat auf das Problem, zu gegebenen rationalen Zahlen $x_1, \ldots, x_{n-1}$ und einem rationalen $\epsilon > 0$ gute simultane diophantische Approximationen zu finden, das heißt rationale Zahlen $\frac{p_1}{q}, \ldots, \frac{p_{n-1}}{q}$ mit möglichst kleinem Hauptnenner q zu konstruieren, so daß $\max_{1 \leq i \leq n-1} |q\, x_i - p_i| \leq \epsilon$. Wir zeigen unter obiger Annahme, daß kein Algorithmus existiert, der für gegebene rationale Zahlen $x_1, \ldots, x_{n-1}$ und natürlicher Zahl N polynomial–Zeit in der Bitlänge $bin(x)$ von x ist und simultane diophantische Approximationen berechnet, so daß $\max_{1 \leq i \leq n-1} |q\, x_i - p_i|$ für $q \in [1, N]$ bis auf den Faktor $2^{\log^{0.5-\zeta} bin(x)}$ minimal ist. Hierbei ist ζ wieder eine beliebig kleine positive Konstante.

Die vorliegende Arbeit umfaßt 5 Kapitel. Kapitel 1 stellt einige Grundbegriffe sowie Notationen bereit und gibt eine Einführung in die Gittertheorie. In Kapitel 2 analysieren wir den Relationenalgorithmus von Just und berichten über die Implementierung. In Kapitel 3 untersuchen wir die Anwendung des Relationenalgorithmus auf das Problem der Diophantischen Approximation. In Kapitel 4 beweisen wir die untere Schranke für die Approximierbarkeit kürzester Relationen in der Maximum–Norm. In Kapitel 5 zeigen wir die untere Schranke für die Approximierbarkeit minimaler guter simultaner diophantischer Approximationen.

Bedanken möchte ich mich insbesondere bei meinem akademischen Lehrer, Prof. Dr. Claus Peter Schnorr, für die umfassende Ausbildung und viele fruchtbare sowie anregende Diskussionen bei der Durchführung dieser Arbeit. Für äußerst hilfreiche Denkanstöße richte ich meinen Dank außerdem an Prof. Dr. Johann Baumeister. Bei Dr. Ralph Werchner möchte ich mich für viele nützliche Verbesserungsvorschläge zu dieser Arbeit bedanken. Mein Dank gilt auch Jean–Pierre Seifert für die Zusammenarbeit bei einigen Papers und Prof. Dr. Jeff Lagarias (AT & T) für mehrere wertvolle Tips.

Kapitel 1

Einführung

In der nachfolgenden Einführung sind im 1. Paragraphen grundlegende Definitionen aufgeführt. Die Paragraphen 2 und 3 sind für das Verständnis der vorliegenden Arbeit nicht erforderlich und dienen nur als Referenz für die in Kapitel 2 und 3 zitierten Sätze und Algorithmen.

Im 2. Paragraphen stellen wir Begriffe der Gittertheorie bereit. Im 3. Paragraphen behandeln wir die Größenreduktion von Gitterbasen und den L^3–Algorithmus zur Gitterbasenreduktion. Im 4. Paragraphen führen wir Relationen ein. Im 5. Paragraphen geben wir einen polynomial–Zeit Algorithmus zum Finden von Relationen an.

Kapitel 4 und 5 benutzen die Resultate aus Paragraph 2 bis 5 nicht.

1.1 Notationen

Es bezeichnen $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{Z}$, $\mathbb{N}$ respektive die Menge der reellen, rationalen, ganzen und natürlichen Zahlen. Für die Menge aller positiven reellen bzw. rationalen Zahlen steht $\mathbb{R}_+$ bzw. $\mathbb{Q}_+$. Es ist $\mathbb{R}^n$ bzw. $\mathbb{Q}^n$ der mit dem Standardskalarprodukt $< .,. >$ und der (euklidischen) Länge bzw. Norm $\|y\| :=< y, y >^{1/2}$ ausgestattete n–dimensionale Vektorraum über $\mathbb{R}$ bzw. $\mathbb{Q}$. Vektoren v des $\mathbb{R}^n$ oder $\mathbb{Q}^n$ sind stets Spaltenvektoren $v = (v_1, \ldots, v_n)^\top$. Wir betrachten für einen n–dimensionalen Vektor $y := (y_1, \ldots, y_n)^\top$ ebenso die *Maximum*– bzw. ∞–*Norm* $\|y\|_\infty := \max_{1 \le i \le n} |y_i|$. Es ist $\mathbb{Z}^n$ die Menge aller ganzzahliger Vektoren im $\mathbb{R}^n$. $\mathbb{Z}^n$ und $\mathbb{Q}^n$ sind additive Untergruppen des $\mathbb{R}^n$.

Die *Dimension* $\dim(U)$ eines Unter(vektor–)raumes $U \subseteq \mathbb{R}^n$ ist die maximale Anzahl linear unabhängiger Vektoren, die U aufspannen. Wir schreiben $vol_m(K)$ für das bezüglich des Jordan–Maßes definierte Volumen einer m–dimensionalen kompakten Teilmenge K des $\mathbb{R}^n$. Für einen Unterraum $U \subseteq \mathbb{R}^n$ steht $U^\perp$ für dessen orthogonales Komplement. Es bezeichnet $span(y_1, \ldots, y_m) \subseteq \mathbb{R}^n$ den von den Vektoren $y_1, \ldots, y_m \in \mathbb{R}^n$ aufgespannten reellen Unterraum. Wir verstehen unter $\{y_1, \ldots, y_m\}$ das *(geordnete) System* der Vektoren $y_1, \ldots, y_m \in \mathbb{R}^n$. Es stehen e_i , $i = 1, \ldots, n$ für die Einheitsvektoren des $\mathbb{R}^n$.

Es stellt $M(n,m,\mathbb{R})$ die Menge aller $n \times m$–Matrizen mit reellen Einträgen, $M(n,m,\mathbb{Q})$ solche mit rationalen Einträgen und $M(n,m,\mathbb{Z})$ diejenigen mit ganzzahligen Einträgen dar. Es steht $M^\top$ für die *Transponierte* einer Matrix M. Für eine quadratische Matrix M bezeichnet $\det(M)$ ihre *Determinante*. $GL_n(\mathbb{R})$, $GL_n(\mathbb{Q})$ sowie $GL_n(\mathbb{Z})$ ist die Gruppe aller in $M(n,n,\mathbb{R})$, $M(n,n,\mathbb{Q})$ und $M(n,n,\mathbb{Z})$, respektive, invertierbaren $n \times n$–Matrizen. Die Elemente aus $GL_n(\mathbb{Z})$ heißen auch *unimodulare* Matrizen. Eine ganzzahlige Matrix M ist unimodular genau dann, wenn $|\det(M)| = 1$ ist.

Weiterhin bezeichnet $[y_1,\ldots,y_m] \in M(n,m,\mathbb{R})$ die $n \times m$–Matrix mit Spaltenvektoren $y_1,\ldots,y_m \in \mathbb{R}^n$.

Wir definieren ferner die Funktion $\lfloor . \rceil$ der nächsten ganzen Zahl durch $\lfloor x \rceil := \lceil x - 0.5 \rceil$ für $x \in \mathbb{R}$ und das *Kronecker–Symbol* $\delta_{i,j}$ durch $\delta_{i,j} = 1$, falls $i = j$, und $\delta_{i,j} = 0$ sonst.

$\log(.)$ bezeichnet stets die Logarithmus–Funktion zur Basis 2. Für zwei ganze Zahlen a, b ist $ggT(a,b)$ der positive *größte gemeinsame Teiler* von a und b.

Rechenmodelle. Für die Analyse von Algorithmen verwenden wir in dieser Arbeit zwei unterschiedliche Rechenmodelle:

Einheitskostenmodell: Im Einheitskostenmodell werden folgende arithmetische Operationen auf exakten reellen Zahlen erlaubt: $+$ (Addition), $-$ (Subtraktion), $*$ (Multiplikation), $/$ (Division), $<$ (Vergleich) sowie die Funktion $\lfloor . \rceil$ (nächste ganze Zahl). Jede arithmetische Operation wird als 1 Rechenschritt gezählt.

Arithmetik auf ganzen Zahlen: Für die Arithmetik auf ganzen Zahlen sind als Eingaben nur ganze und rationale Zahlen erlaubt. Als Komplexitätsmaß für Algorithmen verwenden wir die *Bitkomplexität.* In diesem Berechnungsmodell wird jede Bitoperation $\vee$ (Disjunktion), $\wedge$ (Konjunktion), $\oplus$ (Addition modulo 2) sowie $\neg$ (Negation) als 1 Rechenschritt gezählt.

Die Bitlänge $bin(.)$ einer ganzen Zahl z ist dabei definiert durch $bin(z) := 1 + \lceil \log(|z|+1) \rceil$, die Bitlänge $bin(.)$ einer rationalen Zahl p/q mit teilerfremden p,q durch $bin(p/q) := 1 + \lceil \log(|p|+1) \rceil + \lceil \log(|q|+1) \rceil$. Die Bitlänge eines rationalen Vektors $v := (v_1,\ldots,v_n)^\top \in \mathbb{Q}^n$ ist gegeben durch $\sum_{i=1}^n bin(v_i)$.

Die Anzahl der Bitoperationen für die nach der Schulmethode auf ganzen Zahlen der Bitlänge n_1 und n_2 ausgeführten arithmetischen Operationen $+$ (Addition), $-$ (Subtraktion), $<$ (Vergleich) ist durch $n_1 + n_2$ beschränkt. Die Operationen $*$ (Multiplikation), *div* (ganzzahlige Division mit Rest), angewendet auf zwei ganzen Zahlen der Bitlänge n_1 und n_2, kostet hingegen nach der Schulmethode $2\,n_1\,n_2$ Bitoperationen. Damit ist die Anzahl der Bitoperationen für die nach der Schulmethode auf rationalen Zahlen der Bitlänge n_1 und n_2 ausgeführten arithmetischen Operationen $+$ (Addition), $-$ (Subtraktion), $*$ (Multiplikation), $/$ (Division), $<$ (Vergleich) durch $O(n_1\,n_2)$ beschränkt. Diese Schranke gilt auch für die Anzahl der Bitoperationen der Funktion $\lfloor . \rceil$ (nächste ganze Zahl), angewendet auf eine rationale Zahl mit Zähler der Bitlänge n_1 und Nenner der Bitlänge n_2.

Ein Algorithmus hat *polynomiale Bitkomplexität,* wenn die Anzahl der von ihm durchgeführten Bitoperationen polynomial in der Bitlänge seiner Eingabe beschränkt ist.

Die Algorithmen der vorliegenden Arbeit verwenden auf rationalen und ganzen Zahlen die arithmetischen Operationen $+$ (Addition), $-$ (Subtraktion), $*$ (Multiplikation), $/$ (Division), $<$ (Vergleich) sowie die Funktion $\lfloor . \rceil$ (nächste ganze Zahl). Für die Bitkomplexität der Algorithmen genügt es, statt der Anzahl der Bitoperationen die Anzahl der arithmetischen Operationen $+\,,\ -\,,\ *\,,\ /\,,\ <\,,\ \lfloor . \rceil$ und die *Bitlänge* der im Algorithmus auftretenden ganzen und rationalen Zahlen anzugeben.

NP–Begriffe. *Alphabete* sind endliche Mengen von Symbolen. Das *binäre Alphabet* $\{0,1\}$ besteht nur aus den Symbolen 0 und 1. Ein *String* x ist eine endliche Folge von Elementen eines Alphabets mit Länge $|x|$. Für ein Alphabet Γ bezeichnet Γ^* die Menge aller endlichen Strings einschließlich des *leeren Strings* ε. Eine Teilmenge $L \subseteq \Gamma^*$ heißt *Entscheidungsproblem* oder auch *Sprache* (über Γ). Ohne Beschränkung der Allgemeinheit werden wir nur Sprachen über $\{0,1\}$ betrachten.

Die *Klasse* $\boldsymbol{P}$ *der polynomial–Zeit Sprachen* ist die Klasse der Sprachen L, deren charakteristische Funktion in polynomial–Zeit berechenbar ist. Eine Sprache L ist in der *Klasse* $\boldsymbol{NP}$ *der nicht–deterministischen polynomial–Zeit Sprachen,* falls ein $L' \in \mathbf{P} \cap (\Sigma^* \times \Sigma^*)$ und ein $k \in \mathbb{N}$ existieren, so daß

$$x \in L \Leftrightarrow \exists y \in \Sigma^* \ :\ (x,y) \in L' \ \text{ und } \ |y| \le |x|^k .$$

y bezeichnen wir hierbei als *Zeugen* oder *polynomial–Zeit Beweis* für $x \in L$.

Definition 1.1 *Eine Sprache $L_1 \subseteq \Sigma^*$ ist* (Karp–)reduzierbar *auf eine Sprache $L_2 \subseteq \Sigma^*$, in Zeichen $L_1 \le_{pol.} L_2$, falls eine polynomial–Zeit berechenbare Funktion $f : \Sigma^* \to \Sigma^*$ existiert, so daß*

$$\forall x \in \Sigma^* \ :\ x \in L_1 \Leftrightarrow f(x) \in L_2 \ .$$

Eine Sprache L heißt NP–hart *, falls für alle $L' \in \boldsymbol{NP}$ gilt: $L' \le_{pol.} L$. Eine Sprache L heißt* NP–vollständig *, falls sie $\boldsymbol{NP}$–hart ist und zudem in der Klasse $\boldsymbol{NP}$ enthalten ist.*

Definition 1.2 *Laufzeiten, die mit festem $k \in \mathbb{N}$ durch $2^{(\log|x|)^k}$ beschränkt sind, heißen* quasi–polynomial. *Die* Klasse **QP** der quasipolynomial–Zeit Sprachen *ist die Klasse der in quasipolynomialer Zeit entscheidbaren Sprachen.*

Wir übernehmen folgende Sprechweise aus [FGLSS91, ABSS93]:

Definition 1.3 *Eine Sprache L heißt* fast–**NP**–hart, *falls ein polynomial–Zeit Algorithmus, der L entscheidet, jedes Problem $L' \in \boldsymbol{NP}$ in quasipolynomialer Zeit entscheidet.*

1.2 Gitter

Definition 1.4 *Seien $b_1, \ldots, b_m \in \mathbb{R}^n$ linear unabhängige Vektoren. Dann heißt die Menge*

$$L := L(b_1, \ldots, b_m) := \left\{ \sum_{i=1}^{m} r_i\, b_i \ \middle|\ r_i \in \mathbb{Z} \right\}$$

Gitter. *Das System* $\{b_1, \ldots, b_m\}$ *ist* Basis *des Gitters* L, $dim(L) := m$ *wird als* Dimension *bzw.* Rang *von* L *bezeichnet.*

Satz 1.5 *Eine additive Untergruppe* U *des* $\mathbb{R}^n$ *ist genau dann ein Gitter, wenn* U diskret *ist, das heißt keinen Häufungspunkt im* $\mathbb{R}^n$ *besitzt.*

Beweis. [GrL87], Seite 18, Theorem 1. □

Korollar 1.6 *Die Menge* $\mathbb{Z}^n$ *aller* n*–dimensionalen ganzzahligen Vektoren und das Bild jeder durch unimodulare Matrizen definierten linearen Abbildung des* $\mathbb{Z}^n$ *ist ein Gitter.*

Definition 1.7 *Sei* $L \subseteq \mathbb{R}^n$ *ein Gitter. Wir nennen ein System* $\{b_1, \ldots, b_k\} \subseteq \mathbb{R}^n$ *von linear unabhängigen Gittervektoren* primitiv bezüglich L, *falls* $L(b_1, \ldots, b_k) = span(b_1, \ldots, b_k) \cap L$. *Ein nicht trivialer Gitterpunkt* $b \in L$ *heißt primitiv, falls im Innern der Strecke zwischen dem Nullpunkt* 0 *und dem Punkt* b *kein Gitterpunkt liegt.*

Aus obiger Definition folgt unmittelbar:

Korollar 1.8 *Die primitiven Punkte im Gitter* $\mathbb{Z}^n$ *sind alle Vektoren* $v := (v_1, \ldots, v_n)^\top \neq 0$ *mit* $ggT(v_1, \ldots, v_n) = 1$.

Satz 1.9 *Sei* $L \subseteq \mathbb{R}^n$ *Gitter und* $\{b_1, \ldots, b_k\} \subseteq \mathbb{R}^n$ *ein System von linear unabhängigen Gittervektoren. Das System* $\{b_1, \ldots, b_k\} \subseteq \mathbb{R}^n$ *ist genau dann primitiv bezüglich* L, *wenn es zu einer Gitterbasis von* L *ergänzbar ist.*

Beweis. [GrL87], Seite 21, Theorem 5. □

Definition 1.10 *Die* Determinante $d(L)$ *eines Gitters* $L \subseteq \mathbb{R}^n$ *mit Gitterbasis* $\{b_1, \ldots, b_m\}$ *ist definiert durch*

$$d(L) := (det(\,(< b_i, b_j >)_{1 \leq i,j \leq m})\,)^{1/2} = det(B^\top B)^{1/2} \ ,$$

wobei $B := [b_1, \ldots, b_m]$.

Satz 1.11 *Sei* $L \subseteq \mathbb{R}^n$ *ein Gitter mit Gitterbasis* $\{b_1, \ldots, b_m\}$. *Das System* $\{\overline{b}_1, \ldots, \overline{b}_m\}$ *ist genau dann (Gitter–)Basis von* L, *wenn eine unimodulare Matrix* $T \in GL_m(\mathbb{Z})$ *existiert, so daß*

$$[\overline{b}_1, \ldots, \overline{b}_m] = [b_1, \ldots, b_m]\, T \ .$$

Beweis. [GrL87] Seite 22, Theorem 7. □

Bemerkungen. *1. Die Determinante* $d(L)$ *eines* m*–dimensionalen Gitters* $L :=$

$L(b_1,\ldots,b_m)$ ist geometrisch interpretierbar als das m–dimensionale Volumen des von den Vektoren $b_1,\ldots,b_m \in \mathbb{R}^n$ im $\mathbb{R}^m$ erzeugten Parallelepipeds:

$$d(L) = vol_m \left\{ \sum_{i=1}^{m} x_i\, b_i \mid x_i \in [0,1] \cap \mathbb{R}\ ,\ \ i = 1,\ldots,m \right\} .$$

2. Nach Satz 1.11 ist die Gitterdeterminante basisunabhängig. Wir verwenden später diese Eigenschaft unter dem Begriff Invarianz der Gitterdeterminante (bei unimodularen Transformationen).

Definition 1.12 *Die* sukzessiven Minima $\lambda_1(L),\ldots,\lambda_m(L)$ *eines Gitters $L \subseteq \mathbb{R}^n$ bezüglich der euklidischen Norm sind erklärt durch*

$$\begin{aligned} \lambda_i(L) \ &:= \ \min\{\, r \in \mathbb{R}_+ \ :\ \exists\ i \text{ linear unabhängige Vektoren } c_j \in L \text{ mit} \\ &\qquad \|c_j\| \ \le\ r\ ,\ \ j = 1,\ldots,i\,\}\ . \end{aligned}$$

$\lambda(L) := \lambda_1(L)$ gibt dabei die Länge des kürzesten Gittervektors an.

Bemerkung. *Wie die Gitterdeterminante sind auch die sukzessiven Minima geometrische Invarianten des Gitters, das heißt, sie ändern sich unter isometrischen Abbildungen nicht.*

Definition 1.13 *Für $n \in \mathbb{N}$ ist die* Hermite–Konstante γ_n *definiert durch*

$$\gamma_n \ :=\ \max_{\substack{L \text{ Gitter} \\ dim(L)=n}} \frac{\lambda(L)^2}{d(L)^{2/n}}\ .$$

Die Hermite–Konstante ist für die Dimensionen 2 bis 8 bekannt und jeweils kleiner als 2 . Sei $\Gamma : \mathbb{R} \to \mathbb{R}$ die *Gamma–Funktion.* Für allgemeine Dimensionen n hat Minkowski [Mi05] folgende obere Schranke für γ_n gezeigt:

Satz 1.14 $\quad \gamma_n \ \le\ \frac{4}{\pi}\,\Gamma(1+n/2)^{2/n}$.

Dabei gilt für die Werte der Γ–Funktion $\Gamma(x+1) = x\,\Gamma(x)$ für $x \in \mathbb{R}$ und $\Gamma(\frac{1}{2}) = \sqrt{\pi}$. Insbesondere folgt γ_9 , $\gamma_{10} \le 2.5$. Nach der Stirling'schen Formel ist

$$\Gamma(1+n/2)\ \le\ \left(\frac{n+2}{2\,e}\right)^{(n+2)/2} \sqrt{(n+2)\,\pi}\, e^{\frac{1}{6\,(n+2)}}\ .$$

Für $n \ge 10$ gilt $e^{\frac{1}{6\,(n+2)}} \ \le\ 1.014$. Wir erhalten folgendes

Korollar 1.15 $\quad \gamma_n \ \le\ \max\{2.5, 1.708\,\frac{4}{\pi}\,\frac{n+1}{2\,e}\}\ \le\ \max\{2.5, 0.408\,(n+1)\}$.

Definition 1.16 *Sei $L \subseteq \mathbb{R}^n$ Gitter mit Basis $\{b_1, \dots, b_m\}$. Dann ist das* duale Gitter L^* *von L erklärt durch* $L^* := \{y \in span(b_1, \dots, b_m) \mid \forall x \in L \, : < y, x > \in \mathbb{Z}\}$.

Satz 1.17 *Sei $L \subseteq \mathbb{R}^n$ Gitter mit Basis $\{b_1, \dots, b_m\}$. Dann existiert eine Basis $\{a_1, \dots, a_m\} \subseteq \mathbb{R}^n$ des dualen Gitters L^*, die durch $< b_i, a_j > = \delta_{i,j}$, $1 \le i, j \le m$ eindeutig bestimmt ist. Insbesondere gilt:*

1. $dim(L^*) = dim(L) = m$.
2. $[a_1, \dots, a_m]^\top [b_1, \dots, b_m] = I_m$, *wobei I_m die Einheismatrix des $\mathbb{R}^m$ ist.*
3. $d(L^*) = d(L)^{-1}$.

Beweis. [Cas71], Seite 24, Lemma 5. □

Korollar 1.18 *Das Gitter $\mathbb{Z}^n$ ist* selbstdual*, das heißt $(\mathbb{Z}^n)^* = \mathbb{Z}^n$.*

1.3 Reduktion von Gitterbasen

Gitterbasenreduktion auf linear unabhängigen Systemen. Ziel der Reduktionstheorie von Gitterbasen ist es, solche Basen eines gegebenen Gitters zu konstruieren, deren Basisvektoren kurz und nahezu orthogonal zueinander sind. Verfahren zur Gitterbasenreduktion wurden für die Dimensionen 2 und 3 von Lagrange [La1773], Gauß [Ga1801], Dirichlet [Di1850] und Vallée [Va86] studiert. Für beliebige Dimensionen prägten Hermite [He1845], Korkine und Zolotarev [KZ1873] sowie Minkowski [Mi05] Reduktionsbegriffe für Gitterbasen in der Sprache der positiv quadratischen Formen.

Lenstra, Lenstra und Lovasz [LLL82] stellten als erste ein polynomial–Zeit Verfahren vor, welches eine Gitterbasis für beliebige Dimension m in eine nahezu orthogonale Basis überführt, in denen die kürzesten Gitterbasisvektoren am Anfang der Basis stehen. Insbesondere ist der erste Basisvektor für sogenannte L^3–reduzierte Basen bis auf einen Faktor $2^{m/2}$ der kürzeste Gittervektor.

Wir stellen nun die zur Einführung der Reduktionsbegriffe nötigen Notationen bereit: Sei $L \subseteq \mathbb{R}^n$ Gitter mit (geordneter) Basis $\{b_1, \dots, b_m\}$. Wir definieren für $j = 1, \dots, m$ die orthogonalen Projektionen

$$\pi_j \; : \; \mathbb{R}^n \longrightarrow span(b_1, , \dots, , b_{j-1})^\perp$$

und setzen $\widehat{b}_j := \pi_j(b_j)$. Die Menge $\{\widehat{b}_1, \dots, \widehat{b}_m\} \subseteq \mathbb{R}^n$ bildet eine *Orthogonalbasis*, welche rekursiv durch die *Gram–Schmidt Orthogonalisierung*, abgekürzt GSO, gegeben ist:

$$\begin{aligned}
\widehat{b}_1 &:= b_1 \\
\widehat{b}_i &:= b_i - \sum_{j=1}^{i-1} \mu_{i,j} \, \widehat{b}_j \, , \text{ wobei} \\
\mu_{i,j} &:= < b_i, \widehat{b}_j > / < \widehat{b}_j, \widehat{b}_j > , \; 1 \le i, j \le n.
\end{aligned}$$

Die Größen $\mu_{i,j}$, $1 \leq i, j \leq m$ heißen *Gram–Schmidt Koeffizienten,* die orthogonalen Vektoren $\widehat{b}_i$, $i = 1, \ldots, m$ werden wir im folgenden als *Höhen* oder *Höhenvektoren* bezeichnen. Die Größen $\mu_{i,j}$, $1 \leq i, j \leq m$ zusammen mit den Quadratlängen der Höhen $\|\widehat{b}_i\|^2$, $1 \leq i \leq m$ werden wir *Gram–Schmidt Größen* nennen.

Für einen Index $1 \leq i \leq m$ gibt der Vektor $(\mu_{i,1}, \ldots, \mu_{i,m})^\top$ die Koordinatendarstellung des Basisvektors b_i bezüglich der Orthogonalbasis $\{\widehat{b}_1, \ldots, \widehat{b}_m\}$ an. Es gilt insbesondere $\mu_{i,i} = 1$ sowie $\mu_{i,j} = 0$, $1 \leq i < j \leq m$, und daher

$$\pi_k(b_i) \;=\; \sum_{j=k}^{i} \mu_{i,j}\, \widehat{b}_j$$

sowie wegen der Orthogonalität der $\widehat{b}_j$

$$\|\pi_k(b_i)\|^2 \;=\; \sum_{j=k}^{i} \mu_{i,j}^2\, \|\widehat{b}_j\|^2 \ .$$

Für die Basis $\{b_1, \ldots, b_m\} \subseteq \mathbb{R}^n$ erhält man die Darstellung

$$[b_1, \ldots, b_m] \;=\; [\widehat{b}_1, \ldots, \widehat{b}_m]\, (\mu_{i,j})^\top_{1 \leq i,j \leq m} \ .$$

Da die Matrix $(\mu_{i,j})_{1 \leq i,j \leq m}$ unimodular ist, erhalten wir mit Defintion 1.10:

Korollar 1.19 *Sei $L \subseteq \mathbb{R}^n$ mit Gitterbasis $\{b_1, \ldots, b_m\}$. Dann gilt für die Determinante $d(L)$ von L :*

$$d(L) \;=\; \det((< b_i, b_j >)_{1 \leq i,j \leq m})^{1/2} \;=\; \prod_{i=1}^{m} \|\widehat{b}_i\| \ .$$

Wir werden später die *normalisierten Gram–Schmidt Koeffizienten* $\tau_{i,j} := < b_i, \widehat{b}_j / \|\widehat{b}_j\| >$ verwenden und entsprechend auch die Darstellung von $\{b_1, \ldots, b_m\}$ durch die *Orthonormalbasis* $\{\widehat{b}_1/\|\widehat{b}_1\|, \ldots, \widehat{b}_m/\|\widehat{b}_m\|\}$, in Matrizenschreibweise

$$[b_1, \ldots, b_m] \;=\; [\widehat{b}_1/\|\widehat{b}_1\|, \ldots, \widehat{b}_m/\|\widehat{b}_m\|]\, (\tau_{i,j})^\top_{1 \leq i,j \leq m} \ .$$

Definition 1.20 *Eine (geordnete) Basis $\{b_1, \ldots, b_m\} \subseteq \mathbb{R}^n$ heißt* größenreduziert, *falls $|\mu_{i,j}| \leq \frac{1}{2}$ für $1 \leq j < i \leq m$. Für eine (geordnete) Basis $\{b_1, \ldots, b_m\} \subseteq \mathbb{R}^n$ heißt insbesondere ein Vektor b_k* größenreduziert, *falls $|\mu_{k,j}| \leq \frac{1}{2}$ für $1 \leq j < k$.*

In Abbildung 1.1 ist eine Routine zur Größenreduktion einer (geordneten) Basis $\{b_1, \ldots, b_m\} \subseteq \mathbb{R}^n$ dargestellt. Ein *Größenreduktionsschritt* $b_k := b_k - \lceil \mu_{k,j} \rfloor\, b_j$ reduziert den Vektor b_k bezüglich b_j. Damit bleiben die Gram–Schmidt Koeffizienten $\mu_{k,i}$, $i = j+1, \ldots, k$, unverändert, und die Gram–Schmidt Koeffizienten $\mu_{k,i}$, $i = 1, \ldots, j$, ändern sich gemäß $\mu_{k,i} := \mu_{k,i} - \lceil \mu_{k,j} \rfloor\, \mu_{j,i}$. Insbesondere wird dadurch $|\mu_{k,j}| \leq \frac{1}{2}$.

Definition 1.21 *Eine (geordnete) Basis $\{b_1, \ldots, b_m\} \subseteq \mathbb{R}^n$ heißt L^3*–reduziert, *falls sie größenreduziert ist und für $2 \leq k \leq m$:*

$$\tfrac{3}{4}\, \|\pi_{k-1}(b_{k-1})\|^2 \;\leq\; \|\pi_{k-1}(b_k)\|^2 \;=\; \|\widehat{b}_k\|^2 + \mu_{k,k-1}^2\, \|\widehat{b}_{k-1}\|^2 \ . \qquad (1.1)$$

Größenreduktions–Routine

EINGABE Basis $\{b_1, \ldots, b_m\} \subseteq \mathbb{R}^n$ eines Gitters $L := L(b_1, \ldots, b_m)$.

1. *Initialisierung.*

Berechne die Gram–Schmidt Koeffizienten $\mu_{i,j}$, $1 \leq i, j \leq m$, mit dem GSO–Verfahren.

2. FOR $k = 2, \ldots, m$ DO

Größenreduktion des Vektors b_k :

FOR $j = k-1, \ldots, 1$ DO

$b_k := b_k - \lceil \mu_{k,j} \rfloor \, b_j$;

Neuberechnung der Gram–Schmidt Koeffizienten $\mu_{k,j}$, $1 \leq i \leq j$:

FOR $i = 1$ TO $j-1$ DO $\quad \mu_{k,i} := \mu_{k,i} - \lceil \mu_{k,j} \rfloor \, \mu_{j,i}$;

$\mu_{k,j} := \mu_{k,j} - \lceil \mu_{k,j} \rfloor$;

AUSGABE größenreduzierte Basis $\{b_1, \ldots, b_m\}$ von L.

Abbildung 1.1: Routine zur Größenreduktion einer Gitterbasis

Bemerkung. *Die Bedingung (1.1) in Definition 1.21 ist äquivalent zu*

$$\tfrac{3}{4}\, \tau_{k-1,k-1}^2 \quad \leq \quad \tau_{k,k}^2 + \tau_{k,k-1}^2 \; .$$

Folgender Satz faßt die Eigenschaften L^3–reduzierter Gitterbasen zusammen; wir werden im folgenden nur die erste Ungleichung benötigen, welche unmittelbar aus der Defintion 1.21 folgt.

Satz 1.22 *Sei L ein Gitter und $\{b_1, \ldots, b_m\} \subseteq \mathbb{R}^n$ eine L^3–reduzierte Basis von L. Dann gilt*

1. $\|\widehat{b}_i\|^2 \;\leq\; 2^{j-i}\, \|\widehat{b}_j\|^2$, $1 \leq i \leq j \leq m$;
2. $\|b_i\|^2 \;\leq\; 2^{i-1}\, \|\widehat{b}_i\|^2$, $1 \leq i \leq m$;
3. $2^{1-i} \;\leq\; \|\widehat{b}_i\|^2 / \lambda_i(L)^2 \;\leq\; 2^{m-1}$, $1 \leq i \leq m$;
4. $\|b_1\|^2 \;\leq\; 2^{(n-1)/2}\, d(L)^{2/m}$.

Beweis. [LLL82], Proposition 1.6 und Bemerkung vor (1.14), Seite 518 (unten). □

Die Konstante $\frac{3}{4}$ ist beliebig gewählt und kann durch jede feste reelle Zahl $\frac{1}{4} < \delta < 1$ ersetzt werden. In Satz 1.22 ist dann der Faktor 2 durch $(\delta - \frac{1}{4})^{-1}$ zu ersetzen. Lenstra, Lenstra und Lovász [LLL82] gaben einen Algorithmus zur Konstruktion L^3–reduzierter Basen an, der für festes $\delta \in (\frac{1}{4}, 1)$ und ganzzahlige Eingabebasen $\{b_1, \ldots, b_m\} \subseteq \mathbb{Z}^n$ polynomial in m , n und der Bitlänge der Eingabevektoren ist. Das Verfahren ist in Abbildung 1.2 angegeben. Es gilt folgender

Gitterbasenreduktions–Algorithmus

nach Lenstra, Lenstra und Lovász [LLL82]

EINGABE Basis $\{b_1, \ldots, b_m\} \subseteq \mathbb{R}^n$ eines Gitters $L := L(b_1, \ldots, b_m)$.

1. *Initialisierung.*

$k := 2$, berechne die Gram–Schmidt Größen $\mu_{i,j}$, $1 \leq i, j \leq m$, und $\|\widehat{b}_i\|^2$, $i = 1, \ldots, m$, mit dem GSO–Verfahren.

2. WHILE $(k \leq m)$ DO

   a. *Größenreduktion des Vektors b_k.*
   FOR $j = k-1, \ldots, 1$ DO $b_k := b_k - \lceil \mu_{k,j} \rfloor\, b_j$;
   berechne die Gram–Schmidt Größen $\mu_{k,j}$, $j = 1, \ldots, k-1$, neu;

   b. *Test auf Austausch.*
   IF $\frac{3}{4}\,\|\pi_{k-1}(b_{k-1})\|^2 \;>\; \|\widehat{b}_k\|^2 + \mu_{k,k-1}^2\,\|\widehat{b}_{k-1}\|^2$

      THEN [vertausche b_{k-1}, b_k; $k := \max\{2,\ k-1\}$; berechne die Gram–Schmidt Größen $\mu_{i,j}$, $1 \leq j < i \leq m$, und $\|\widehat{b}_{k-1}\|^2$, $\|\widehat{b}_k\|^2$ neu]
      ELSE [$k := k+1$]

AUSGABE L^3–reduzierte Basis $\{b_1, \ldots, b_m\}$ von L.

Abbildung 1.2: L^3–Algorithmus

Satz 1.23 *Sei $\{b_1, \ldots, b_m\} \subseteq \mathbb{R}^n$ reelle Eingabebasis des L^3–Algorithmus und $B := \max_{1 \leq i \leq n} \|b_i\|$. Dann berechnet der L^3–Algorithmus eine L^3–reduzierte Gitterbasis und führt höchstens $O(m^3\, n \log(B/\lambda))$ arithmetische Operationen auf reellen Zahlen aus. Dabei ist $\lambda := \lambda(L)$ die Länge des kürzesten Gittervektors in dem durch $b_1, \ldots, b_m$ definierten Gitter $L := L(b_1, \ldots, b_m)$.*

Für ganzzahlige Eingabebasen $\{b_1, \ldots, b_m\} \subseteq \mathbb{Z}^n$ führt der L^3–Algorithmus höchstens $O(m^3\, n \log B)$ arithmetische Operationen auf ganzen Zahlen der Bitlänge $O(m \log B)$ aus.

Beweis. [LLL82], Proposition 1.26. □

Reduktion auf linear abhängigen Systemen. Wir verallgemeinern den L^3–Reduktionsbegriff auf ein (geordnetes) System $D := \{b_0, b_1, \ldots, b_m\} \subseteq \mathbb{R}^n$ von linear abhängigen Vektoren x, $b_1, \ldots, b_m$ mit festem Vektor x. Wir nennen $D := \{x, b_1, \ldots, b_m\} \subseteq \mathbb{R}^n$ ein *Basissystem*, falls $\{b_1, \ldots, b_m\} \subseteq \mathbb{R}^n$ eine Basis des Gitters $\mathbb{Z}^m$ darstellt. (Diese Situation tritt auf, wenn der Vektor x durch Basisvektoren $b_1, \ldots, b_n$ des Gitters $\mathbb{Z}^n$ simultan approximiert wird. Bei der Approximation der Geraden $x\,\mathbb{R}$ durch Gitterbasen $b_1, \ldots, b_n$ des $\mathbb{Z}^n$ werden Austausche und Größenreduktionsschritte auf den linear abhängigen zu x orthogonalen Projektionen $\pi_x(b_1), \ldots, \pi_x(b_n)$, jedoch nicht mit dem Vektor x ausgeführt.)

Für $j = 1, \ldots, m$ definieren wir die orthogonalen Projektionen

$$\pi_{j,x} \,:\, \mathbb{R}^n \longrightarrow span(x\,,\; b_1, , \ldots, , b_{j-1})^\perp$$

und setzen $\widehat{b}_{j,x} := \pi_{j,x}(b_j)\,,\, j = 1, \ldots, m\,.$

Die Menge $\{x\,,\, \widehat{b}_{1,x}, \ldots, \widehat{b}_{m,x}\} \subseteq \mathbb{R}^n$ bildet ein Orthogonalsystem mit mindestens $m+1-n$ verschwindenden Vektoren. Die Gram–Schmidt Koeffizienten $\mu_{i,j}$ werden bezüglich der Höhen $x\,,\, \widehat{b}_{1,x}, \ldots, \widehat{b}_{m,x}$ definiert, wobei $\mu_{i,0} :=< b_i, x > / < x, x >$ für $j = 0$ und $\mu_{i,j} := 0$ für $\widehat{b}_{j,x} = 0$. Für die normalisierten Gram–Schmidt Koeffizienten $\tau_{i,j}$ setzen wir analog $\tau_{i,0} :=< b_i, x/\|x\| >$ für $j = 0$ und $\tau_{i,j} := 0$ für $\widehat{b}_{j,x} = 0$.

Im HJLS–Algorithmus in Paragraph 1.5 und in den Algorithmen in Kapitel 2 und 3 werden auf einem Basissystem $\{b_0\,,\, b_1, \ldots, b_m\}$ stets vor Austauschen $b_{k-1}\leftrightarrow b_k$, das heißt im Falle $\frac{3}{4}\,\|\pi_{k-1,x}(b_{k-1})\|^2 \;>\; \|\pi_{k-1,x}(b_k)\|^2$, Größenreduktionsschritte der Form $b_k := b_k - \lceil \mu_{k,k-1} \rfloor\, b_{k-1}$ ausgeführt. Folgendes Lemma, welches wir später implizit benutzen werden, zeigt, daß für solche Austauschschritte das Längenquadrat $\|\widehat{b}_{k-1,x}\|^2$ der Höhe $\widehat{b}_{k-1,x}$ mindestens um den Faktor $\frac{3}{4}$ erniedrigt, $\max_{1\leq i\leq m} \|\widehat{b}_{i,x}\|^2$ nicht erhöht wird und $\min_{1\leq i\leq m} \|\widehat{b}_{i,x}\|^2$ im Falle $\widehat{b}_{k,x} \neq 0$ nicht erniedrigt wird.

Lemma 1.24 *Sei $\overline{D} := \{\overline{b}_0\,,\, \overline{b}_1, \ldots, \overline{b}_m\}$ ein Basissystem und es gelte $\frac{3}{4}\,\|\pi_{k-1,x}(\overline{b}_{k-1})\|^2 \;>\; \|\pi_{k-1,x}(\overline{b}_k)\|^2$. Sei $D := \{b_0, \ldots, b_{k-1}\,,\; b_k, \ldots, b_m\}$ das Basissystem, das aus $\overline{D}$ durch Größenreduktion von $\overline{b}_k$ bezüglich $\overline{b}_{k-1}$ und anschließendem Austausch $\overline{b}_{k-1}\leftrightarrow\overline{b}_k$ entsteht. Dann gilt*

1. $\|\widehat{b}_{k-1,x}\|\,\|\widehat{b}_{k,x}\| \;=\; \|\widehat{\overline{b}}_{k-1,x}\|\,\|\widehat{\overline{b}}_{k,x}\|\,.$
2. $\|\widehat{b}_{k-1,x}\|^2 \;\leq\; \frac{3}{4}\,\|\widehat{\overline{b}}_{k-1,x}\|^2\,.$
3. $\max\{\|\widehat{b}_{k-1,x}\|\,,\, \|\widehat{b}_{k,x}\|\} \;\leq\; \|\widehat{\overline{b}}_{k-1,x}\|\,.$
4. $\min\{\|\widehat{b}_{k-1,x}\|\,,\, \|\widehat{b}_{k,x}\|\} \;\geq\; \|\widehat{\overline{b}}_{k,x}\|\,;$
5. $\widehat{b}_{i,x} \;=\; \widehat{\overline{b}}_{i,x}$ *für* $i \neq k-1\,,\; k\,.$

Beweis. 1. Wegen der Invarianz der Gitterdeterminante bei unimodularen Transformationen gilt

$$\begin{aligned}\|\widehat{b}_{k-1,x}\|\,\|\widehat{b}_{k,x}\| \;&=\; d(L(\pi_{k-1,x}(b_{k-1})\,,\; \pi_{k-1,x}(b_k)))\\ &=\; d(L(\pi_{k-1,x}(\overline{b}_{k-1})\,,\; \pi_{k-1,x}(\overline{b}_k))) \;=\; \|\widehat{\overline{b}}_{k-1,x}\|\,\|\widehat{\overline{b}}_{k,x}\|\;.\end{aligned}$$

Dies gilt insbesondere für die Fälle $\widehat{b}_{k-1,x} = \widehat{\overline{b}}_{k,x} = 0$ und $\widehat{b}_{k,x} = \widehat{\overline{b}}_{k,x} = 0$.

2. Es gilt

$$\begin{aligned}\widehat{b}_{k-1,x} \;=\; \pi_{k-1,x}(b_{k-1}) \;&=\; \pi_{k-1,x}(\overline{b}_k - \lceil \overline{\mu}_{k,k-1} \rfloor\, \overline{b}_{k-1})\\ &=\; \widehat{\overline{b}}_{k,x} + (\overline{\mu}_{k,k-1} - \lceil \overline{\mu}_{k,k-1} \rfloor)\, \widehat{\overline{b}}_{k-1,x}\;. \qquad (1.2)\end{aligned}$$

Im Falle $\widehat{\overline{b}}_{k,x} = 0$ ist somit $\|\widehat{b}_{k-1,x}\|^2 \leq \frac{1}{4} \|\widehat{\overline{b}}_{k-1,x}\|^2$. Für den Fall $\widehat{\overline{b}}_{k,x} \neq 0$ erhalten wir nach Voraussetzung

$$\|\widehat{b}_{k-1,x}\|^2 = \|\widehat{\overline{b}}_{k,x}\|^2 + (\overline{\mu}_{k,k-1} - \lceil \overline{\mu}_{k,k-1} \rfloor)^2 \|\widehat{\overline{b}}_{k-1,x}\|^2 < \frac{3}{4} \|\widehat{\overline{b}}_{k-1,x}\|^2 .$$

3. und 4. Zunächst gilt

$$\begin{aligned} \|\widehat{b}_{k,x}\|^2 \leq \|\widehat{b}_{k,x}\|^2 + \mu_{k,k-1}^2 \|\widehat{b}_{k-1,x}\|^2 &= \|\pi_{k-1,x}(b_k)\|^2 \\ &= \|\pi_{k-1,x}(\overline{b}_{k-1})\|^2 = \|\widehat{\overline{b}}_{k-1,x}\|^2 . \end{aligned} \qquad (1.3)$$

Für den Fall $\widehat{b}_{k-1,x} = 0$ zeigt (1.3) insbesondere $\widehat{b}_{k,x} = \widehat{\overline{b}}_{k-1,x}$ und damit 3. Andernfalls folgt 3 aus (1.3) und 2.

Im Falle $\widehat{\overline{b}}_{k,x} = 0$ ist Aussage 4 trivial.
Wegen 1, 2 gilt andernfalls $\|\widehat{b}_{k,x}\|^2 \geq \frac{4}{3} \|\widehat{\overline{b}}_{k,x}\|^2 > \|\widehat{\overline{b}}_{k,x}\|^2$. Aus (1.2) folgt nunmehr $\|\widehat{b}_{k-1,x}\| \geq \|\widehat{\overline{b}}_{k,x}\|$ und somit auch 4.
5 ergibt sich wegen $span(b_0, \ldots, b_{i-1}) = span(\overline{b}_0, \ldots, \overline{b}_{i-1})$ für alle $0 \leq i \leq m$, $i \neq k-1$, k, das heißt, die orthogonalen Projektionen bleiben für alle $0 \leq i \leq m$, $i \neq k-1$, k invariant. □

1.4 Relationen

Definition 1.25 *Für einen Vektor $x \in \mathbb{R}^n$ heißt $m \in \mathbb{Z}^n - \{0\}$* (ganzzahlige) Relation, *falls $< m, x >= 0$. Es bezeichnet $L_x := \{ m \in \mathbb{Z}^n : < m, x >= 0 \}$ das* Relationengitter *zu x.*

L_x ist die Menge aller Relationen für x zuzüglich des Nullvektors 0 und bildet ein Gitter. Für einen beliebigen reellen Vektor x ist $0 \leq dim(L_x) \leq n-1$, wobei offensichtlich $dim(L_x) = n-1$ für rationale Vektoren x gilt. Wir definieren in kanonischer Weise die sukzessiven Minima $\lambda(x) := \lambda_1(x), \ldots \lambda_{dimL_x}(x)$ von L_x :

Definition 1.26

$$\begin{aligned} \lambda_i(x) := \min\{ r \in \mathbb{R}_+ : \exists\, i \textit{ linear unabhängige Vektoren } c_j \in L_x \textit{ mit} \\ \|c_j\| \leq r \,,\ j = 1, \ldots, i \} . \end{aligned}$$

$\lambda(x)$ gibt dabei die Länge der kürzesten Relation für x an, wobei $\lambda(x) := \infty$, falls keine Relation zu x existiert.

Babai, Just, Meyer auf der Heide [BJM88] haben gezeigt, daß in einem sehr allgemeinen Berechnungsmodell[1] für beliebiges $x \in \mathbb{R}^n$ es nicht entscheidbar ist, ob $dim(L_x) \geq 1$ bzw. eine Relation für x existiert.

[1] In dem von [BJM88] betrachteten Berechnungsmodell werden auf reellen Zahlen neben den arithmetischen Operationen $+, -, *, /, <$ und der Rundungsfunktion $\lceil . \rfloor$ auch analytische Funktionen zugelassen (wie $\sqrt{.}$, $\log(.)$, $\sin(.)$ und dergleichen) und werden als 1 Rechenschritt betrachtet. In diesem Berechnungsmodell lassen sich Algorithmen durch sogenannte *analytische Berechnungsbäume* darstellen.

Das Problem, für gegebenes $x \in \mathbb{Q}^n$ und $\epsilon \in \mathbb{Q}_+$ zu entscheiden, ob eine Relation mit Maximum–Norm kleiner als $1/\epsilon$ existiert, ist **NP**–vollständig [vEB81]. Es ist offen, ob dieses Problem **NP**–hart ist, wenn man für die Maximum–Norm die euklidische Norm einsetzt. Es ist daher nicht zu erwarten, daß Algorithmen existieren, die für festes $k \in \mathbb{N}$ und beliebige Eingaben $x \in \mathbb{Q}^n$, in polynomialer Zeit in n bis auf den Faktor n^k für ein $k \in \mathbb{N}$ in der euklidischen Norm kürzeste Relationen finden. Weiterhin ist nicht zu erwarten, daß Algorithmen existieren, die für festes $k \in \mathbb{N}$ und beliebige Eingaben $x \in \mathbb{Q}^n$, $\epsilon \in \mathbb{Q}_+$ in polynomialer Zeit in n und $\lceil |\log \epsilon| \rceil$ die Existenz von Relationen mit euklidischer Länge kleiner als n^k/ϵ für ein $k \in \mathbb{N}$ entscheiden.

Für rationale Vektoren $x \in \mathbb{Q}^n$ ist folgende obere Schranke für $\lambda(x)$ implizit aus [HJLS89], Theorem 6.2 bekannt.

Satz 1.27 *[HJLS89] Für* $x = (p_1, \ldots, p_n)/q \in \mathbb{Q}^n$ *mit* $p_1, \ldots, p_n \in \mathbb{Z}$, $q \in \mathbb{N}$ *und* $ggT(p_1, \ldots, p_n, q) = 1$ *ist*

$$\lambda(x) \;\le\; \max\{\sqrt{2.5}, \sqrt{0.408\, n}\} \left(\sum_{i=1}^n p_i^2\right)^{\frac{1}{2\,(n-1)}} \;\le\; \max\{1.6, 0.64\sqrt{n}\} \left(\sum_{i=1}^n p_i^2\right)^{\frac{1}{2\,(n-1)}} .$$

Beweis. Da x rational ist, hat das Relationengitter L_x den Rang $n-1$. Sei $a_2, \ldots, a_n$ eine Basis von L_x. Jede ganzzahlige Relation zu x ist ganzzahlige Relation zu $q\,x$ und umgekehrt. Somit gilt $L_x = L(a_2, \ldots, a_n) = span(a_2, \ldots, a_n) \cap \mathbb{Z}^n$. Nach Satz 1.9 ist damit das System $\{a_2, \ldots, a_n\}$ zu einer Basis $\{a_1, a_2, \ldots, a_n\}$ von $\mathbb{Z}^n$ ergänzbar. Umgekehrt ist nach Voraussetzung und Korollar 1.8 der Vektor $q\,x$ primitiv bezüglich $\mathbb{Z}^n$. Es gilt daher $|< a_1, q\,x >| = \min\{|< m, q\,x >| \,:\, m \in \mathbb{Z}^n\} = ggT(p_1, \ldots, p_n) = 1$. Da die zu $span(a_2, \ldots, a_n)$ orthogonale Komponente von a_1 genau $< a_1, x/\|x\| > x/\|x\|$ ist, folgt

$$1 \;=\; d(\mathbb{Z}^n) \;=\; d(L_x)\,\frac{|< a_1, q\,x >|}{\|q\,x\|} \;=\; d(L_x)/(q\,\|x\|)$$

und somit

$$d(L_x) \;=\; \left(\sum_{i=1}^n p_i^2\right)^{1/2} .$$

Die Behauptung ergibt sich nun nach Definition 1.13 und Korollar 1.15:

$$\lambda(x) \;\le\; \sqrt{\gamma_{n-1}}\, d(L_x)^{\frac{1}{n-1}} \;\le\; \max\{\sqrt{2.5}, \sqrt{0.408\, n}\} \left(\sum_{i=1}^n p_i^2\right)^{\frac{1}{2\,(n-1)}} . \qquad \square$$

1.5 Der HJLS–Algorithmus

Gegeben ein reeller Vektor $x \in \mathbb{R}^n$ und eine positive reelle Zahl $\epsilon > 0$ findet der HJLS–Algorithmus entweder eine Relation für x der Länge $2^{n/2-1} \min\{\epsilon^{-1}, \lambda(x)\}$ oder beweist, daß jede Relation für x euklidische Länge größer oder gleich ϵ^{-1} haben muß. Der HJLS–Algorithmus terminiert nach $O(n^3\,(n + \lceil |\log \epsilon| \rceil))$ arithmetischen Operationen auf reellen

Relationen–Algorithmus

nach Hastad, Just, Lagarias, Schnorr [HJLS89]

EINGABE $x \in \mathbb{R}^n - \{0\}$, $\epsilon > 0$.

1. Initialisierung. $[b_0, b_1, \ldots, b_n] := [x, e_1, \ldots, e_n]$, $s := 1$, berechne die Größen $\|\widehat{b}_{i,x}\|^2$, $i = 0, \ldots, n$, und $\mu_{i,j}$, $0 \leq i, j \leq n$, mit dem GSO–Verfahren.

2. Test auf Abbruch. Im Falle $\|\widehat{b}_{n,x}\| \neq 0$ ist a_n Relation für x. Berechne $[a_1, \ldots, a_n]^\top := [b_1, \ldots, b_n]^{-1}$, gebe a_n aus und stoppe.
Falls $s = n$, dann gilt $\|\widehat{b}_{i,x}\| \leq \epsilon$ für $i = 1, \ldots, n$ und es existiert keine Relation für x mit Länge kleiner als ϵ^{-1}. Gebe in diesem Falle '$\lambda(x) \geq 1/\epsilon$' aus und stoppe.

3. Austausche. Wähle den Index $1 \leq k \leq n$, der $2^k \|\widehat{b}_{k,x}\|^2$ maximiert. Setze $b_{k+1} := b_{k+1} - \lceil \mu_{k+1,k} \rfloor b_k$ und vertausche b_k und b_{k+1}. Aktualisiere $s := \sharp\{1 \leq i \leq n : \|\widehat{b}_{i,x}\|^2 \leq \epsilon^2\}$ und die Gram–Schmidt Größen $\|\widehat{b}_{i,x}\|^2$, $i = 1, \ldots, n$, sowie $\mu_{i,j}$, $0 \leq i, j \leq n$. Gehe nach 2.

Abbildung 1.3: HJLS–Algorithmus

Zahlen. Der in n exponentielle Faktor, um den die Länge der im HJLS–Algorithmus gefundenen Relation von der kürzesten Relation, das heißt einer Relation mit Länge $\lambda(x)$, abweicht, erscheint erforderlich, damit der HJLS–Algorithmus polynomial–Zeit ist. Bisher sind keine Algorithmen bekannt, die in polynomialer Zeit in n und $\lceil |\log \epsilon| \rceil$ eine Relation für x mit euklidischer Länge kleiner als ϵ^{-1} finden oder eine solche Relation ausschließen.

Der HJLS–Algorithmus beruht auf folgendem Lemma (Proposition 3.1 in [HJLS89]), welches in schächerer Form schon in [FF79] und [Bre81] erschienen ist:

Lemma 1.28 *[HJLS89] Für jede Basis $b_1, \ldots, b_n$ des Gitters $\mathbb{Z}^n$ gilt*

$$\lambda(x) \quad \geq \quad 1/\max_{i=1,\ldots,n} \|\widehat{b}_{i,x}\| \,. \tag{1.4}$$

Methode. Der HJLS–Algorithmus approximiert die Gerade $x\,\mathbb{R}$ durch eine Folge von Gitterbasen des $\mathbb{Z}^n$. Hierzu führt der HJLS–Algorithmus auf dem Basissystem $\{x, b_1, \ldots, b_n\} \subseteq \mathbb{R}^n$ Austausche und Größenreduktionsschritte —also unimodulare Transformationen— durch, wobei $b_1, \ldots, b_n$ anfangs die Einheitsvektoren des $\mathbb{R}^n$ sind. Der Vektor x bleibt unverändert und die Vektoren $b_1, \ldots, b_n$ bilden stets eine Basis des $\mathbb{Z}^n$. Ziel des HJLS–Algorithmus ist es, $\max_{1 \leq i \leq n-1} \|\widehat{b}_{i,x}\|$ zu minimieren. Falls $\max_{1 \leq i \leq n-1} \|\widehat{b}_{i,x}\| \leq \epsilon$, terminiert der HJLS–Algorithmus und beweist in diesem Falle mit Lemma 1.28, daß $\lambda(x) \geq 1/\epsilon$. Die Approximation der Geraden $x\,\mathbb{R}$ mit $\max_{1 \leq i \leq n-1} \|\widehat{b}_{i,x}\| \leq \epsilon$ mißlingt, falls ein Austausch $b_{n-1} \leftrightarrow b_n$ zu einer nicht verschwindenden Höhe $\widehat{b}_{n,x} \neq 0$ führt; in diesem Falle bricht der HJLS–Algorithmus mit $\widehat{b}_{n-1,x} = 0$ bzw. $x \in span(b_1, \ldots, b_{n-1})$ ab. Dann bildet der letzte duale Basisvektor a_n eine Relation zu x.

Aktualisierung der Gram–Schmidt Größen. Wir geben an dieser Stelle die Formeln für die Neuberechnung der Gram–Schmidt Größen $\|\widehat{b}_{i,x}\|^2$, $i = 1, \ldots, n$ sowie $\mu_{i,j}$, $0 \le i, j \le n$, nach einem Austausch $b_{k-1} \leftrightarrow b_k$ und nach Größenreduktionsschritten $b_k := b_k - \lceil \mu_{k,j} \rfloor b_j$ mit $1 \le j < k$ an.

Lemma 1.29 *Seien $b_1^{(alt)}, \ldots, b_n^{(alt)}$ bzw. $b_1^{(neu)}, \ldots, b_{k-1}^{(neu)}$ die Basisvektoren des Basissystems $D := \{x\ ,\ b_1, \ldots, b_n\}$ im HJLS–Algorithmus vor einem Austausch $b_{k-1} \leftrightarrow b_k$. Dann gilt*

$$\begin{aligned} \mu_{k,j}^{(neu)} &= \mu_{k-1,j}^{(alt)}\ ,\ j = 0, \ldots, k-2\ ; && (1.5)\\ \mu_{k-1,j}^{(neu)} &= \mu_{k,j}^{(alt)}\ ,\ j = 0, \ldots, k-2\ ; && (1.6)\\ \|\widehat{b}_{k-1,x}^{(neu)}\|^2 &= \|\widehat{b}_{k,x}^{(alt)}\|^2 + {\mu_{k,k-1}^{(alt)}}^2\ \|\widehat{b}_{k-1,x}^{(alt)}\|^2\ ; && (1.7) \end{aligned}$$

$$\|\widehat{b}_{k,x}^{(neu)}\| = \begin{cases} \|\widehat{b}_{k,x}^{(alt)}\|\ \dfrac{\|\widehat{b}_{k-1,x}^{(alt)}\|}{\|\widehat{b}_{k-1,x}^{(neu)}\|} & \textit{falls}\ \ \widehat{b}_{k-1,x}^{(neu)} \neq 0 \\ \|\widehat{b}_{k,x}^{(alt)}\| & \textit{sonst} \end{cases}\ ; \qquad (1.8)$$

$$\mu_{k,k-1}^{(neu)} = \begin{cases} \mu_{k,k-1}^{(alt)}\ \dfrac{\|\widehat{b}_{k-1,x}^{(alt)}\|^2}{\|\widehat{b}_{k-1,x}^{(neu)}\|^2} & \textit{falls}\ \ \widehat{b}_{k-1,x}^{(neu)} \neq 0 \\ 0 & \textit{sonst} \end{cases}\ ; \qquad (1.9)$$

$$\mu_{i,k-1}^{(neu)} = \begin{cases} \mu_{i,k}^{(alt)} + \mu_{k,k-1}^{(neu)}\ (\mu_{i,k-1}^{(alt)} - \mu_{k,k-1}^{(alt)}\ \mu_{i,k}^{(alt)}) & \textit{falls}\ \ \widehat{b}_{k-1,x}^{(neu)} \neq 0 \\ 0 & \textit{sonst} \end{cases}\ , \qquad (1.10)$$

$$\mu_{i,k}^{(neu)} = \begin{cases} (\mu_{i,k-1}^{(alt)} - \mu_{k,k-1}^{(alt)}\ \mu_{i,k}^{(alt)}) & \textit{falls}\ \ \widehat{b}_{k-1,x}^{(neu)} \neq 0 \\ \mu_{i,k-1}^{(alt)} & \textit{sonst} \end{cases}\ ,\ k < i \le n\ . \qquad (1.11)$$

Die Neuberechnung der Gram–Schmidt Größen $\|\widehat{b}_{i,x}\|^2$, $i = 1, \ldots, n$, sowie $\mu_{i,j}$, $0 \le i, j \le n$, geht in $O(n)$ arithmetischen Operationen.

Beweis. Die Höhen $\widehat{b}_{j,x}$, $j \neq k-1$, k bleiben bei einem Austausch $b_{k-1} \leftrightarrow b_k$ gemäß Lemma 1.24 (5) unverändert. Somit gilt für $j = 0, \ldots, k-2$

$$\mu_{k,j}^{(neu)} = \frac{< b_k^{(neu)}, \widehat{b}_{j,x}^{(neu)} >}{\|\widehat{b}_{j,x}^{(neu)}\|^2} = \frac{< b_{k-1}^{(alt)}, \widehat{b}_{j,x}^{(alt)} >}{\|\widehat{b}_{j,x}^{(alt)}\|^2} = \mu_{k-1,j}^{(alt)}$$

und analog $\mu_{k-1,j}^{(neu)} = \mu_{k,j}^{(alt)}$, $j = 0, \ldots, k-2$.

Es gilt

$$\widehat{b}_{k-1,x}^{(neu)} = \pi_{k-1,x}(b_{k-1}^{(neu)}) = \pi_{k-1,x}(b_k^{(alt)}) = \widehat{b}_{k,x}^{(alt)} + \mu_{k,k-1}^{(alt)}\ \widehat{b}_{k-1,x}^{(alt)}$$

und somit

$$\|\widehat{b}_{k-1,x}^{(neu)}\|^2 \;=\; \|\widehat{b}_{k,x}^{(alt)}\|^2 + \mu_{k,k-1}^{(alt)\,2}\,\|\widehat{b}_{k-1,x}^{(alt)}\|^2 \;.$$

Umgekehrt ist

$$\widehat{b}_{k-1,x}^{(alt)} \;=\; \pi_{k-1,x}(b_{k-1}^{(alt)}) \;=\; \pi_{k-1,x}(b_k^{(neu)}) \;=\; \widehat{b}_{k,x}^{(neu)} + \mu_{k,k-1}^{(neu)}\,\widehat{b}_{k-1,x}^{(neu)} \;.$$

Im Falle $\widehat{b}_{k-1,x}^{(neu)} = 0$ folgt somit $\|\widehat{b}_{k,x}^{(neu)}\|^2 = \|\widehat{b}_{k-1,x}^{(alt)}\|^2$ sowie nach Definition $\mu_{k,k-1}^{(neu)} = 0$. Andernfalls folgt aus der Invarianz der Gitterdeterminante $d(L(\pi_{k-1,x}(b_{k-1})\,,\;\pi_{k-1,x}(b_k))) = \|\widehat{b}_{k-1,x}\|\,\|\widehat{b}_{k,x}\|$ bei Austauschen $b_{k-1}\leftrightarrow b_k$

$$\|\widehat{b}_{k,x}^{(neu)}\|^2 \;=\; \|\widehat{b}_{k,x}^{(alt)}\|^2\,\frac{\|\widehat{b}_{k-1,x}^{(alt)}\|^2}{\|\widehat{b}_{k-1,x}^{(alt)}\|^2}$$

und außerdem

$$\mu_{k,k-1}^{(neu)} \;=\; \frac{< b_k^{(neu)}, \widehat{b}_{k-1,x}^{(neu)} >}{\|\widehat{b}_{k-1,x}^{(neu)}\|^2} \;=\; \frac{< b_{k-1}^{(alt)}, \widehat{b}_{k,x}^{(alt)} + \mu_{k,k-1}^{(alt)}\,\widehat{b}_{k-1,x}^{(alt)} >}{\|\widehat{b}_{k-1,x}^{(neu)}\|^2} \;=\; \mu_{k,k-1}^{(alt)}\,\frac{\|\widehat{b}_{k-1,x}^{(alt)}\|^2}{\|\widehat{b}_{k-1,x}^{(neu)}\|^2}\;.$$

Für die Gleichungen (1.10) und (1.11) im Falle $\widehat{b}_{k-1,x}^{(neu)} \neq 0$ verweisen wir auf die in [LLL82], Figur 1, Seite 521 angegebenen Formeln. Im Falle $\widehat{b}_{k-1,x}^{(neu)} = 0$ ist nach Definition $\mu_{i,k-1}^{(neu)} = 0$, $i = k+1,\ldots,n$, und wegen $\widehat{b}_{k,x}^{(neu)} = \widehat{b}_{k-1,x}^{(alt)}$ folgt dann $\mu_{i,k-1}^{(neu)} = \mu_{i,k}^{(alt)}$, $i = k+1,\ldots,n$.

Man sieht unmittelbar, daß $O(k) + O(n-k) = O(n)$ Rechenschritte erforderlich sind, um die durch den Austausch $b_{k-1}\leftrightarrow b_k$ veränderten Gram–Schmidt Größen aus den alten Gram–Schmidt Größen neu zu berechnen. □

Größenreduktionsschritte $b_k := b_k - \lceil \mu_{k,j} \rfloor\, b_j$ mit $1 \leq j < k$ lassen die Höhen $\widehat{b}_{j,x}$ unverändert. Für einen Größenreduktionsschritt $b_k^{(neu)} = b_k^{(alt)} - \lceil \mu_{k,j}^{(alt)} \rfloor\, b_j^{(alt)}$ ändern sich lediglich die Gram–Schmidt Koeffizienten $\mu_{k,i}$, $1 \leq i \leq j$, gemäß

$$\begin{aligned}\mu_{k,i}^{(neu)} \;&=\; \frac{< b_k^{(neu)}, \widehat{b}_{i,x}^{(neu)} >}{\|\widehat{b}_{i,x}^{(neu)}\|^2} \;=\; \frac{< b_k^{(alt)} - \lceil \mu_{k,j}^{(alt)} \rfloor\, b_j^{(alt)}, \widehat{b}_{i,x}^{(alt)} >}{\|\widehat{b}_{i,x}^{(alt)}\|^2} \\ &=\; \mu_{k,i}^{(alt)} - \lceil \mu_{k,j}^{(alt)} \rfloor\, \mu_{j,i}^{(alt)}\;. \end{aligned} \tag{1.12}$$

Nach einem Größenreduktionsschritt von b_k bezüglich b_j, das heißt nach Setzen von $b_k^{(neu)} := b_k^{(alt)} - \lceil \mu_{k,j}^{(alt)} \rfloor\, b_j^{(alt)}$, gilt damit insbesondere $|\mu_{k,j}^{(neu)}| \;\leq\; \frac{1}{2}$.

Jeder Größenreduktionsschritt $b_k := b_k - \lceil \mu_{k,j} \rfloor\, b_j$ mit $1 \leq j < k$ einschließlich der in (1.12) angegebenen Aktualisierung der Gram–Schmidt Größen kostet $O(n+j) = O(n)$ arithmetische Operationen auf reellen Zahlen.

Folgender Satz faßt Korrektheit und Laufzeit des HJLS–Algorithmus zusammen:

Satz 1.30 *[HJLS89], Theorem 3.2*
Für Eingaben $x \in \mathbb{R}^n$ und $\epsilon > 0$ findet der HJLS–Algorithmus entweder eine Relation a_n für x mit $\|a_n\| \leq 2^{n/2-1}\,\min\{\epsilon^{-1}\,,\;\lambda(x)\}$ oder beweist $\lambda(x) \geq \epsilon^{-1}$. Der HJLS–Algorithmus terminiert nach $O(n^3\,(n + \lceil|\log\epsilon|\rceil))$ arithmetischen Operationen auf reellen Zahlen.

Beweis. Wir folgen dem Beweis von [HJLS89].

Größe der Relation a_n : Seien $b_1, \dots, b_n$ die Endbasisvektoren von $\mathbb{Z}^n$ im Relationen–Algorithmus. Falls der Algorithmus mit $\|\widehat{b}_{i,x}\| \le \epsilon$, $i = 1, \dots, n-1$ abbricht, beweist Lemma 1.28, daß $\lambda(x) \ge 1/\epsilon$. Findet der Relationen–Algorithmus eine Relation a_n, so gilt nach Satz 1.17, daß $< a_n, b_n >= 1$ und $a_n \in span(b_1, \dots, b_{n-1})^\perp$. Nach der Abbruchbedingung in diesem Falle ist weiterhin $\widehat{b}_{n-1,x} = 0$, insbesondere also $x \in span(b_1, \dots, b_{n-1})$. Hieraus folgt $\widehat{b}_{n,x} = \widehat{b}_n$ sowie $span(\widehat{b}_{n,x}) = span(a_n)$ und somit

$$1 \;=\; < a_n, b_n > \;=\; < a_n, \widehat{b}_{n,x} > \;=\; \|a_n\| \, \|\widehat{b}_{n,x}\| \,,$$

$$\Rightarrow \; a_n \;=\; \widehat{b}_{n,x}/\|\widehat{b}_{n,x}\|^2 \;=\; \widehat{b}_n/\|\widehat{b}_n\|^2 \,, \;\; \|a_n\| \;=\; \|\widehat{b}_{n,x}\|^{-1} \,.$$

Seien $\overline{b}_1, \dots, \overline{b}_n$ die Basisvektoren vor dem letzten Austausch $b_{n-1} \leftrightarrow b_n$. Falls kein Austausch $b_{n-1} \leftrightarrow b_n$ existiert ist $a_n = e_n$ und die Behauptung über die Länge von a_n gezeigt. Wegen $2^{n-1} \|\widehat{\overline{b}}_{n-1,x}\|^2 = \max_{1 \le i \le n} 2^i \|\widehat{\overline{b}}_{i,x}\|^2$ und $\widehat{b}_{n,x} = \widehat{\overline{b}}_{n-1,x}$ gilt $\|\widehat{\overline{b}}_{i,x}\|^2 \le 2^{n-1-i} \|\widehat{\overline{b}}_{n-1,x}\|^2 = 2^{n-1-i} \|\widehat{b}_{n,x}\|^2$, $i = 1, \dots, n$. Da der Algorithmus nicht zuvor abgebrochen war, existiert ein Index j mit $\|\widehat{\overline{b}}_{j,x}\| \ge \epsilon$. Somit folgt

$$\|a_n\|^2 \;=\; \|\widehat{\overline{b}}_{n-1,x}\|^{-2} \;\le\; 2^{n-2}/\max_{1 \le i \le n} \|\widehat{\overline{b}}_{i,x}\|^2 \;\stackrel{Lemma\ 1.28}{\le}\; 2^{n-2} \min\{\epsilon^{-1}, \; \lambda(x)\} \,.$$

Laufzeit: Wir zeigen zunächst, daß jeder Austausch $b_{k-1} \leftrightarrow b_k$ mit vorangegangener Größenreduktion von b_k bezüglich b_{k-1} ($b_k := b_k - \lceil \mu_{k,k-1} \rfloor \, b_{k-1}$) die Größe $D := \prod_{i=1}^{n-1} \beta(\widehat{b}_{i,x})^{n-i}$ mit $\beta(\widehat{b}_{i,x}) := \max\{\|\widehat{b}_{i,x}\|^2 \, 2^n \,, \; \epsilon^2\}$ um den Faktor $\frac{3}{4}$ erniedrigt. Seien $b_j^{(alt)}$, $1 \le j \le n$ die Vektoren b_j vor, $b_j^{(neu)}$, $1 \le j \le n$ die Vektoren b_j nach einem Austausch $b_{k-1} \leftrightarrow b_k$. Vor dem Austausch $b_{k-1} \leftrightarrow b_k$ ist $i := k-1$ so gewählt, daß $2^{k-1} \|\widehat{b}_{k-1,x}^{(alt)}\|^2$ maximal ist. Damit gilt insbesondere $\|\widehat{b}_{k-1,x}^{(alt)}\|^2 \ge 2 \, \|\widehat{b}_{k,x}^{(alt)}\|^2$. Weiterhin ist $b_k^{(alt)}$ bezüglich $b_{k-1}^{(alt)}$ größenreduziert und daher ist $|\mu_{k,k-1}| \le \frac{1}{2}$ vor jedem Austausch $b_{k-1} \leftrightarrow b_k$. Somit folgt

$$\|\widehat{b}_{k-1,x}^{(neu)}\|^2 \;=\; \|\pi_{k-1,x}(b_k^{(alt)})\|^2 \;=\; \|\widehat{b}_{k,x}^{(alt)}\|^2 + \mu_{k,k-1}^2 \, \|\widehat{b}_{k-1,x}^{(alt)}\|^2 \;\le\; \tfrac{3}{4} \, \|\widehat{b}_{k-1,x}^{(alt)}\|^2 \tag{1.13}$$

Da der Relationen–Algorithmus in Schritt 2 nicht terminiert ist, existiert mindestens ein Index $1 \le j \le n-1$, so daß $\|\widehat{b}_{j,x}^{(alt)}\| \ge \epsilon$. Somit ist

$$2^n \, \|\widehat{b}_{k-1,x}^{(alt)}\|^2 \;\ge\; 2^{k-1} \, \|\widehat{b}_{k-1,x}^{(alt)}\|^2 \;\ge\; 2^j \, \|\widehat{b}_{j,x}^{(alt)}\|^2 \;\ge\; 2\,\epsilon^2 \,. \tag{1.14}$$

Hieraus und aus $\|\widehat{b}_{k,x}^{(neu)}\|^2 \le \|\widehat{b}_{k-1,x}^{(alt)}\|^2$ folgt

$$\beta(\widehat{b}_{k,x}^{(neu)}) \;\le\; \beta(\widehat{b}_{k-1,x}^{(alt)}) \,. \tag{1.15}$$

Wir beweisen nun folgende Ungleichung

$$\frac{\beta(\widehat{b}_{k-1,x}^{(neu)}) \, \beta(\widehat{b}_{k,x}^{(neu)})}{\beta(\widehat{b}_{k-1,x}^{(alt)}) \, \beta(\widehat{b}_{k,x}^{(alt)})} \;\le\; 1 \,. \tag{1.16}$$

Beweis. <u>Fall (i):</u> $\beta(\widehat{b}_{k-1,x}^{(neu)}) = \epsilon^2$. Dann folgt (1.16) aus $\beta(\widehat{b}_{k,x}^{(alt)}) \ge \epsilon^2$ und (1.15).

Fall (ii): $\beta(\widehat{b}_{k,x}^{(neu)}) = \epsilon^2$. (1.13) impliziert insbesondere $\beta(\widehat{b}_{k-1,x}^{(neu)}) \leq \beta(\widehat{b}_{k-1,x}^{(alt)})$. Mit $\beta(\widehat{b}_{k,x}^{(alt)}) \geq \epsilon^2$ folgt daher (1.16).

Fall (iii): $\beta(\widehat{b}_{k-1,x}^{(neu)}) = 2^n \, \|\widehat{b}_{k-1,x}^{(neu)}\|^2 \geq \epsilon^2$, $\beta(\widehat{b}_{k,x}^{(neu)}) = 2^n \, \|\widehat{b}_{k,x}^{(neu)}\|^2 \geq \epsilon^2$. In diesem Fall ergibt sich (1.16) aus der Invarianz der Gitterdeterminante

$$\begin{aligned}\|\widehat{b}_{k-1,x}^{(alt)}\| \, \|\widehat{b}_{k,x}^{(alt)}\| &= d(L(\pi_{k-1,x}(b_{k-1}^{(alt)}), \ \pi_{k-1,x}(b_k^{(alt)}))) \\ &= d(L(\pi_{k-1,x}(b_{k-1}^{(neu)}), \ \pi_{k-1,x}(b_k^{(neu)}))) \ = \ \|\widehat{b}_{k-1,x}^{(neu)}\| \, \|\widehat{b}_{k,x}^{(neu)}\| \, . \qquad \Diamond\end{aligned}$$

Wir zeigen weiterhin

$$\beta(\widehat{b}_{k-1,x}^{(neu)}) / \beta(\widehat{b}_{k-1,x}^{(alt)}) \ \leq \ \tfrac{3}{4} \, . \tag{1.17}$$

Beweis. Fall (i): $\beta(\widehat{b}_{k-1,x}^{(neu)}) = \epsilon^2$. Dann folgt (1.17) direkt aus (1.14).

Fall (ii): $\beta(\widehat{b}_{k-1,x}^{(neu)}) = 2^n \, \|\widehat{b}_{k-1,x}^{(neu)}\|^2 \geq \epsilon^2$. Wegen (1.13) muß dann $\beta(\widehat{b}_{k-1,x}^{(alt)}) = 2^n \, \|\widehat{b}_{k-1,x}^{(alt)}\|^2$ gelten; wiederum mit (1.13) folgt dann (1.17). $\Diamond$

Bezeichne $D^{(alt)}$, $D^{(neu)}$ die Größe D vor bzw. nach einem Austausch $b_{k-1} \leftrightarrow b_k$, so gilt nunmehr

$$\frac{D^{(neu)}}{D^{(alt)}} \ = \ \frac{\beta(\widehat{b}_{k-1,x}^{(neu)})^{n-k+1} \, \beta(\widehat{b}_{k,x}^{(neu)})^{n-k}}{\beta(\widehat{b}_{k-1,x}^{(alt)})^{n-k+1} \, \beta(\widehat{b}_{k,x}^{(alt)})^{n-k}} \ \stackrel{(1.16)}{\leq} \ \frac{\beta(\widehat{b}_{k-1,x}^{(neu)})}{\beta(\widehat{b}_{k-1,x}^{(alt)})} \ \stackrel{(1.17)}{\leq} \ \frac{3}{4} \, ,$$

das heißt jeder Austausch $b_{k-1} \leftrightarrow b_k$ mit vorangegangener Größenreduktion $b_k := b_k - \lceil \mu_{k,k-1} \rfloor \, b_{k-1}$ verringert D um den Faktor $\frac{3}{4}$. Zu Beginn ist $b_i = e_i$, $1 \leq i \leq n$, und somit $\|\widehat{b}_{i,x}\| \leq 1$, $1 \leq i \leq n$, also $D \leq 2^{n \binom{n}{2}}$. Wenn der Algorithmus abbricht, ist $D \geq \epsilon^{2 \binom{n}{2}}$. Somit führt der Relationen–Algorithmus nach [HJLS89] höchstens $\binom{n}{2}(\lceil \log_{\frac{4}{3}} 2 \rceil \, n + 2 \lceil \log_2 1/\epsilon \rceil) \ \leq \ \binom{n}{2}(3\,n + 2\lceil |\log_2 \epsilon| \rceil)$ Austausche $b_{k-1} \leftrightarrow b_k$ aus. Nach Lemma 1.29 kostet jeder Austausch $b_{k-1} \leftrightarrow b_k$ und die vorausgehende Reduktion jeweils höchstens $O(n)$ arithmetische Operationen. Die Berechnung der Gram–Schmidt Größen $\|\widehat{b}_{i,x}\|^2$, $i = 1, \ldots, n$, sowie $\mu_{i,j}$, $0 \leq i, j \leq n$, zu Beginn des HJLS–Algorithmus kostet $O(n^3)$ arithmetische Operationen. Die Invertierung der Matrix $[b_1, \ldots, b_n]$ der Endbasisvektoren zur Berechnung von a_n erfordert nach der Schulmethode ebenfalls $O(n^3)$ arithmetische Operationen. Damit ist die Gesamtzahl der arithmetischen Operationen durch $O(n \binom{n}{2}(3\,n + 2\lceil |\log_2 \epsilon| \rceil)) + O(n^3) \ = O(n^3 \, (n + \lceil |\log \epsilon| \rceil))$ beschränkt. $\Box$

Bemerkungen. *1. Statt bei Abbruch des Relationen–Algorithmus die dualen Basisvektoren a_i, $i = 1, \ldots, n$ über die inverse Matrix $B^* := [b_1, \ldots, b_n]^{-1}$ gemäß $a_i = B^{*\top} e_i$ zu berechnen, kann man die dualen Basisvektoren auch nach jeder unimodularen Transformation von $b_1, \ldots, b_n$ wie folgt aktualisieren: Zu Beginn des Algorithmus ist $a_i = e_i$, $i = 1, \ldots, n$. Für jeden Austausch $b_{k-1} \leftrightarrow b_k$ werden a_{k-1}, a_k miteinander vertauscht. Für jeden Größenreduktionsschritt $b_k := b_k - \lceil \mu_{k,k-1} \rfloor \, b_{k-1}$ wird $a_{k-1} := a_{k-1} + \lceil \mu_{k,k-1} \rfloor \, a_k$ gesetzt. Letzeres folgt daraus, daß die zur unimodularen Matrix*

$\begin{pmatrix} 1 & 0 \\ -\lceil \mu_{k,k-1} \rfloor & 1 \end{pmatrix}$ *transponierte Inverse die Form* $\begin{pmatrix} 1 & 0 \\ \lceil \mu_{k,k-1} \rfloor & 1 \end{pmatrix}$ *hat.*

2. Der HJLS–Algorithmus verwendet die Bergman*–Austauschregel, das heißt, zwei Basisvektoren* b_{k-1} , b_k *werden ausgetauscht, falls* $i := k-1$ *die Größe* $2^i \, \|\widehat{b}_{i,x}\|^2$ *maximiert. Die* L^3*–Austauschregel hingegen vertauscht die Vektoren* b_{k-1} , b_k*, mit minimalem Index* $i := k-1$*, so daß* $2 \, \|\widehat{b}_{k-1,x}\|^2 > \|\widehat{b}_{k,x}\|^2$ *gilt. Die Bergman–Austauschregel garantiert polynomiale Laufzeit des HJLS–Algorithmus. Würde man in dem HJLS–Algorithmus statt der Bergman–Austauschregel die* L^3*–Austauschregel einsetzen, so ist eine in* n *und* $\lceil |\log \epsilon| \rceil$ *polynomiale Rechenzeitschranke für den so konstruierten Algorithmus bei beliebigen reellen Eingaben nicht mehr gegeben. Falls keine Relation für die Eingabe* x *existiert, gerät ein solcher Algorithmus in die Situation, daß die ersten beiden Vektoren unendlich oft miteinander vertauscht werden.*

In Kapitel 2 werden wir einen Algorithmus zum Finden ganzzahliger Relationen angeben, der für Eingaben $x \in \mathbb{R}^n$ *und* $\epsilon > 0$ *höchstens* $O(n^3 \, (n + \lceil |\log \epsilon| \rceil))$ *arithmetische Operationen auf reellen Zahlen ausführt, gleichermaßen, ob die* L^3*–Austauschregel oder die Bergman–Austauschregel verwendet wird.*

Kapitel 2

Relationenalgorithmen und Stabilität

In diesem Kapitel präsentieren wir einen Relationenalgorithmus, der —wie der HJLS–Algorithmus— für Eingaben $x \in \mathbb{R}^n$ und $\epsilon > 0$ in polynomialer Zeit in n und $\lceil |\log \epsilon| \rceil$ entweder eine kurze Relation für x findet oder $\lambda(x) \geq 1/\epsilon$ beweist. Zusätzlich berechnet der Algorithmus Folgen $(\{b_1^{(k)}, \ldots, b_n^{(k)}\})_{k \in \mathbb{N}}$, $(\{a_1^{(k)}, \ldots, a_n^{(k)}\})_{k \in \mathbb{N}}$ von dualen Gitterbasen des $\mathbb{Z}^n$, deren Basisvektoren während des gesamten Algorithmus klein bleiben. Wir zeigen, daß der letzte Basisvektor der dualen Basis $\{a_1^{(k)}, \ldots, a_n^{(k)}\}_{k \in \mathbb{N}}$ während des gesamten Algorithmus in der Länge durch $2^{n/2} \min\{\epsilon^{-1}, \lambda(x)\}$ beschränkt ist und verbessern damit die Analyse von [HJLS89]. Hastad, Just, Lagarias und Schnorr bewiesen diese obere Schranke bei Terminierung ihres Relationenalgorithmus. Mit der oberen Schranke für den letzten dualen Basisvektor gelingt es uns, die Länge aller dualen Basisvektoren $a_1^{(k)}, \ldots, a_n^{(k)}$ durch $1.5^n \, (\max_{1 \leq i \leq n-1} \|\widehat{b}_{i,x}\|^{-1} + 2^{n/2} \min\{\epsilon^{-1}, \lambda(x)\})$ zu beschränken. Mit der oberen Schranke für die Länge des letzten dualen Basisvektors beweisen wir die bisher beste obere Schranke für die Länge der primären Basisvektoren. In Kapitel 3 verwenden wir dieses Resultat dazu, gute diophantische Approximationen $(p_1, \ldots, p_{n-1}, q)$ an reelle Eingaben $x_1, \ldots, x_{n-1}$ zu konstuieren (Satz 3.3, Kapitel 3).

Unser Algorithmus konstruiert im Falle der Ausgabe $\lambda(x) \geq 1/\epsilon$ einen *Nahebeipunkt* x' zu x, für den der letzte duale Basisvektor a_n eine Relation darstellt. a_n ist eine bis auf den Faktor $2^{n/2+1}$ kürzeste Relation für x'. Wir zeigen, daß für Punkte $\overline{x}$, die in einer offenen Umgebung um x mit Radius $\|x - x'\|/2$ liegen, keine Relation der Länge kleiner als $(2\,\epsilon)^{-1}$ existiert. Unser Algorithmus liefert in diesem Sinne eine stetige untere Schranke für die Länge der kürzesten Relation für Punkte $\overline{x}$, deren Abstand von x kleiner als $\|x - x'\|/2$ ist.

Bei der Approximation der Geraden $x\,\mathbb{R}$ durch eine Folge $(\{b_1^{(k)}, \ldots, b_n^{(k)}\})_{k \in \mathbb{N}}$ von Gitterbasen des $\mathbb{Z}^n$ liefert die Orthogonalisierung des Systems $\{x, b_1^{(k)}, \ldots, b_n^{(k)}\}$ kleine orthogonale Vektoren, deren Längen keine feste untere Schranke erfüllen. Insbesondere ist das in [HJLS89] vorgeschlagene Verfahren zur Orthogonalisierung von $\{x, b_1^{(k)}, \ldots, b_n^{(k)}\}$ —einschließlich des Aktualisierungsverfahrens gemäß Lemma 1.29—, instabil. Unser Relationenalgorithmus ist numerisch stabil, wenn wir für die Berechnung der Gram–Schmidt Größen das Verfahren der Givens Rotation benutzen. Wir bezeichnen daher unseren Algorithmus im folgenden auch als Stabilen Relationenalgorithmus SIRA (Stable Integer Relation Algorithm)[1].

Wir stellen unseren Algorithmus im 1. Paragraphen vor. Die Analyse des Algorithmus erfolgt in Paragraph 2. Im 3. Paragraphen geben wir eine Variante des Algorithmus an, welche für rationale Eingaben polynomiale Bitkomplexität besitzt. Im 4. Paragraphen stellen wir als Orthogonalisierungsverfahren die Givens Rotation vor und analysieren damit die numerische Stabilität unseres Algorithmus. Details über die Implementierung des Algorithmus und experimentelle Resultate stehen im 5. Paragraphen.

2.1 Der Relationenalgorithmus

SIRA findet auf Eingaben $x \in \mathbb{R}^n$, $\epsilon > 0$ entweder ein ganzzahlige Relation für x der Länge $2^{n/2-1} \min\{\epsilon^{-1}, \lambda(x)\}$ oder beweist '$\lambda(x) \geq 1/\epsilon$'. Im zweiten Fall konstruiert SIRA außerdem einen Nahebeipunkt $x' \in \mathbb{R}^n$ zu x und eine Relation $m \in \mathbb{Z}^n - \{0\}$ für x', welche $\|m\| \leq 4 \cdot 2^{n/2-1}\lambda(x')$ erfüllt.

Methode. Wie der HJLS–Algorithmus approximiert SIRA die Gerade $x\,\mathbb{R}$ durch eine Folge von Gitterbasisvektoren $b_1, \ldots, b_n$ des $\mathbb{Z}^n$. Hierzu werden auf den zu x orthogonalen Projektionen $\pi_x(b_1), \ldots, \pi_x(b_n)$ Austausche und Größenreduktionsschritte durchgeführt mit dem Ziel, $\max_{1 \leq i \leq n-1} \|\widehat{b}_{i,x}\|$ zu minimieren. Falls ein Austausch $b_{n-1} \leftrightarrow b_n$ zu einer nicht verschwindenden Höhe $\widehat{b}_{n,x} \neq 0$ führt, bricht der Algorithmus mit $x \in span(b_1, \ldots, b_{n-1})$ ab und gibt als Relation den letzten dualen Basisvektor a_n aus.

Solange $\max_{1 \leq i \leq n} \|\widehat{b}_{i,x}\| > \epsilon$ und $\widehat{b}_{n,x} = 0$ führt SIRA folgende Schritte aus:

1. L^3–Reduktion der projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_{n-1})$:

Es werden Vektoren b_{k-1}, b_k, $2 \leq k \leq n-1$ vertauscht, welche die L^3–Bedingung $\frac{3}{4}\|\pi_{k-1,x}(b_{k-1})\|^2 \leq \|\pi_{k-1,x}(b_k)\|^2$ nicht erfüllen; die Wahl des jeweiligen Stufenindex k kann hierbei sowohl nach der L^3–Austauschregel als auch nach der Bergman–Austauschregel erfolgen. Entscheidend ist, daß die projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_{n-1})$ ein Gitter aufspannen, das L^3–Reduktionsverfahren somit für beide Austauschregeln terminiert. Im Unterschied zum HJLS–Algorithmus sind in Analogie zu dem in Abschnitt 1.3 angegebenen L^3–Algorithmus alle projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_n)$ vor einem Austausch $b_{n-1} \leftrightarrow b_n$ an letzter Stelle *vollständig* reduziert, das heißt jeder projizierte Vektor $\pi_x(b_k)$, $2 \leq k \leq n$ ist bezüglich *aller* vorangehenden projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_{k-1})$ reduziert. Dies erfolgt durch die Größenreduktion aller projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_n)$ am Ende der L^3–Reduktion.

[1] Diese Bezeichnung mag auf den ersten Blick irreführend sein, da die Stärke des Relationenalgorithmus eindeutig in seiner Anwendung auf Konstruktion guter diophantischer Approximationen liegt (siehe Kapitel 3). Die Namensgebung ist schlichtweg dadurch bedingt, daß der Ausgangspunkt der Problemstellung der vorliegenden Arbeit die Eingabestabilität und numerische Stabilität von Relationenalgorithmus war.

Nach der L^3–Reduktion wird die Anzahl s der Basisvektoren b_i mit $\|\widehat{b}_{i,x}\| \leq \epsilon$ gezählt.

Der Algorithmus stoppt, falls $s = n$ bzw. $\|\widehat{b}_{i,x}\| \leq \epsilon$ für $i = 1, \ldots, n-1$. In diesem Falle ist der letzte duale Basisvektor $a_n = \widehat{b}_n / \|\widehat{b}_n\|^2$ eine Relation für $x' := x - < x, \widehat{b}_n / \|\widehat{b}_n\| > \widehat{b}_n / \|\widehat{b}_n\|$ und es gilt '$\lambda(x) \geq 1/\epsilon$'.

2. Austausch an letzter Stelle:

Nach L^3–Reduktion der projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_{n-1})$ und Größenreduktion aller projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_n)$ werden die Vektoren b_{n-1}, b_n vertauscht. Diese Modifikation stammt von [Ju92]. Falls nach dem Austausch $b_{n-1} \leftrightarrow b_n$ die Höhe $\widehat{b}_{n,x} \neq 0$ wird, bricht das Verfahren ab; dann ist $x \in span(b_1, \ldots, b_{n-1})$ und der letzte duale Basisvektor a_n eine Relation für x.

3. Für die Berechnung der L^3–Bedingung und die Bestimmung der Reduktionskoeffizienten $\lceil \mu_{i,j} \rfloor$ berechnen wir die Orthonormalisierung $(\tau_{i,j})_{\substack{0 \leq i \leq n \\ 1 \leq j \leq n}}$ mit $\tau_{i,j} := < b_i, \widehat{b}_{j,x} > / \|\widehat{b}_{j,x}\|$, $0 \leq i \leq n$, $1 \leq j \leq n$ von $[x, b_1, \ldots, b_n]$. Es gilt dann $\tau_{i,i}^2 = \|\widehat{b}_{i,x}\|^2$, $i = 0, \ldots, n$ sowie $\mu_{i,j} = \tau_{i,j} / \tau_{j,j}$, $0 \leq i, j \leq n$. Für die Orthonormalisierung von $[x, b_1, \ldots, b_n]$ werden wir entweder das GSO–Verfahren oder das Verfahren der Givens Rotation verwenden, welches in Abschnitt 2.4 vorgestellt wird.

4. Während des Algorithmus gilt stets $[a_1, \ldots, a_n]^\top := [b_1, \ldots, b_n]^{-1}$. Statt bei Abbruch von SIRA den Vektor a_n über die inverse Matrix $B^* := [b_1, \ldots, b_n]^{-1}$ gemäß $a_n = B^{*\top} e_n$ zu berechnen, kann man die dualen Basisvektoren $a_1, \ldots, a_n$ analog zum HJLS–Algorithmus wie folgt aktualisieren: Zu Beginn des Algorithmus ist $a_i = e_i$, $i = 1, \ldots, n$. Für jeden Austausch $b_{k-1} \leftrightarrow b_k$ werden a_{k-1}, a_k vertauscht. Für jeden Größenreduktionsschritt $b_k := b_k - \lceil \tau_{k,j} / \tau_{j,j} \rfloor b_j$, $1 \leq j < k$, ist $a_j = a_j + \lceil \tau_{k,j} / \tau_{j,j} \rfloor a_k$ zu setzen, damit $[a_1, \ldots, a_n]^\top := [b_1, \ldots, b_n]^{-1}$ erfüllt bleibt.

Stabiler Relationenalgorithmus (SIRA)

EINGABE $x \in \mathbb{R}^n - \{0\}$, $\epsilon > 0$.

1. Initialisierung. $[b_0, b_1, \ldots, b_n] := [x, e_1, \ldots, e_n]$, $s := 1$,

berechne die Orthonormalisierung $(\tau_{i,j})_{\substack{0 \le i \le n \\ 1 \le j \le n}}$ von $[x, b_1, \ldots, b_n]$.

Falls $\tau_{n,n} > 0$, dann ist e_n eine Relation für x. Gebe den Punkt $x' := x$ und die Relation $a_n := e_n$ für x aus, und stoppe.

2. *L^3-Reduktion von $\pi_x(b_1), \ldots, \pi_x(b_{n-1})$.*

WHILE $(\exists 1 < k < n : \frac{3}{4}\tau_{k-1,k-1}^2 > \tau_{k,k}^2 + \tau_{k,k-1}^2)$ DO

Setze $b_k := b_k - \lceil \tau_{k,k-1}/\tau_{k-1,k-1} \rfloor b_{k-1}$;

vertausche b_{k-1} und b_k und berechne die Orthonormalisierung $(\tau_{i,j})_{\substack{0 \le i \le n \\ 1 \le j \le n}}$ neu.

Größenreduziere alle projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_n)$.

WHILE $|\tau_{s,s}| \le \epsilon$ DO $s := s + 1$.

3. *Austausch $b_{n-1} \leftrightarrow b_n$.*

Vertausche b_{n-1} und b_n und berechne die Orthonormalisierung $(\tau_{i,j})_{\substack{0 \le i \le n \\ 1 \le j \le n}}$ neu.

Falls $\tau_{n,n} = 0$ und $s < n$ dann gehe nach 2.

4. *Abbruch.* Berechne $[a_1, \ldots, a_n]^\top := [b_1, \ldots, b_n]^{-1}$.

Im Falle $\tau_{n,n} > 0$ ist a_n Relation für x. Gebe den Punkt $x' := x$ und a_n aus, und stoppe. Falls $s = n$, dann gilt $\tau_{i,i} \le \epsilon$ für $i = 1, \ldots, n$, und es existiert keine Relation für x mit Länge kleiner als ϵ^{-1}. Berechne in diesem Fall $\pi_n(x) =< x, \widehat{b}_n/\|\widehat{b}_n\| > \widehat{b}_n/\|\widehat{b}_n\| \in span(b_1, \ldots, b_{n-1})^\perp$, und gebe den Punkt $x' := x - \pi_n(x)$, die Relation a_n für x' sowie '$\lambda(x) \ge 1/\epsilon$' aus.

Korrektheit von SIRA. Die Korrektheit von SIRA beweist folgendes

Lemma 2.1 *1. Nach Schritt 2 gilt stets $\tau_{i,i} \le \epsilon$ for $i = 1, \ldots, s-1$ und die projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_{n-1})$ sind L^3-reduziert.*

2. Vor jedem Austausch $b_{n-1} \leftrightarrow b_n$ gilt stets $s < n$.

3. Der Ausgabevektor a_n ist ganzzahlige Relation zum ausgegebenen Punkt x'.

4. Bricht SIRA mit $x' \ne x$ ab, so gilt $\lambda(x) \ge 1/\epsilon$.

Beweis. 1 und 2 folgen unmittelbar aus den Anweisungsvorschriften der Schritte 2 und 3.

3. Seien $b_1, \ldots, b_n$ die Endbasisvektoren für SIRA. Sowohl im Falle $x' = x$ (vergleiche Satz 1.30) als auch im Falle $x' = x- < x, \widehat{b}_n/\|\widehat{b}_n\| > \widehat{b}_n/\|\widehat{b}_n\| \ne x$ ist $x' \in span(b_1, \ldots, b_{n-1})$. Der letzte duale Basisvektor $a_n = \widehat{b}_n/\|\widehat{b}_n\|^2 \in span(b_1, \ldots, b_{n-1})^\perp$ ist damit in beiden Fällen der Terminierung von SIRA ganzzahlige Relation zu x'.

4 wurde schon in Satz 1.30 bewiesen. □

2.2 Analyse in exakter reeller Arithmetik

Wir zeigen zuerst, daß SIRA für Eingaben $x \in \mathbb{R}^n$, $\epsilon > 0$ bei Ausgabe $x' \neq x$ die Aussage $\lambda(\overline{x}) \geq 1/(2\,\epsilon)$ für Punkte $\overline{x}$ in einer offenen Kugel um x mit Radius $\|x - x'\|/2$ beweist. Hierzu benötigen wir folgendes

Lemma 2.2 *Seien x , $\overline{x} \in \mathbb{R}^n$ und π_x , $\pi_{\overline{x}}$ die orthogonalen Projektionen auf span$(x)^\perp$, span$(\overline{x})^\perp$, respektive.*

1. Dann gilt für alle $b \in \mathbb{R}^n$

$$\|\pi_x(b) - \pi_{\overline{x}}(b)\| \;\leq\; \frac{2\,\|b\|\,\|x - \overline{x}\|}{\max\{\|x\|\,,\,\|\overline{x}\|\}}\,. \tag{2.1}$$

2. Für jede Basis $b_1, \ldots, b_n$ des $\mathbb{Z}^n$ gilt

$$\|\widehat{b}_{i,x} - \widehat{b}_{i,\overline{x}}\| \;\leq\; 2\,\|\widehat{b}_{i,x}\|\,\frac{\|x - \overline{x}\|}{\|\pi_n(x)\|}\,,\; i = 1, \ldots, n-1\,.$$

3. Für die Ausgabebasis $b_1, \ldots, b_n \in \mathbb{Z}^n$ und den ausgegebenen Punkt $x' = x - \pi_n(x)$ von SIRA gilt

$$\|\widehat{b}_{i,x} - \widehat{b}_{i,x'}\| \;\leq\; \|\widehat{b}_{i,x}\|\,,\; i = 1, \ldots, n-1\,.$$

Beweis. 1. Der Beweis verwendet folgende Ungleichung, die von Clarkson, [Cl92] Lemma 3.2 bewiesen wurde:

$$\left|\frac{< b, x >}{\|x\|^2} - \frac{< b, \overline{x} >}{\|\overline{x}\|^2}\right| \;\leq\; \frac{\|b\|\,\|x - \overline{x}\|}{\|x\|\,\|\overline{x}\|}\,.$$

Hieraus und aus der Cauchy–Schwarz Ungleichung folgt

$$\begin{aligned}
\|\pi_x(b) - \pi_{\overline{x}}(b)\| &\leq \left\| b - \frac{< b, x >}{\|x\|^2} x - (b - \frac{< b, \overline{x} >}{\|\overline{x}\|^2}\overline{x})\right\| \\
&= \left\|\frac{< b, \overline{x} >}{\|\overline{x}\|^2}\overline{x} - \frac{< b, x >}{\|x\|^2}\overline{x} + \frac{< b, x >}{\|x\|^2}\overline{x} - \frac{< b, x >}{\|x\|^2}x\right\| \\
&\leq \|\overline{x}\|\left|\frac{< b, \overline{x} >}{\|\overline{x}\|^2} - \frac{< b, x >}{\|x\|^2}\right| + \frac{|< b, x >|}{\|x\|^2}\|\overline{x} - x\| \\
&\leq \frac{\|b\|\,\|\overline{x} - x\|}{\|x\|} + \frac{\|b\|}{\|x\|}\|\overline{x} - x\| \;=\; 2\,\frac{\|b\|\,\|x - \overline{x}\|}{\|x\|}\,.
\end{aligned}$$

Vertauschen der Rollen von x und $\overline{x}$ im obigen Beweis ergibt die gewünschte Behauptung.
2. Wir wenden Ungleichung (2.1) auf $b = \widehat{b}_i$, $x = \pi_i(x)$, $\overline{x} = \pi_i(\overline{x})$ an. Wegen $\widehat{b}_i$, $\pi_i(x)$, $\pi_i(\overline{x}) \in span(b_1, \ldots, b_{i-1})^\perp$ ist $\pi_{\pi_i(x)}(\widehat{b}_i) = \widehat{b}_{i,x}$ und $\pi_{\pi_i(\overline{x})}(\widehat{b}_i) = \widehat{b}_{i,\overline{x}}$. Wir erhalten somit

$$\|\widehat{b}_{i,x} - \widehat{b}_{i,\overline{x}}\| \;\leq\; \|\widehat{b}_i\|\,\frac{2\,\|\pi_i(x) - \pi_i(\overline{x})\|}{\max\{\|\pi_i(x)\|\,,\,\|\pi_i(\overline{x})\|\}} \;=\; \|\widehat{b}_{i,x}\|\,\frac{2\,\|\pi_i(x) - \pi_i(\overline{x})\|}{\max\{\|\pi_{i+1}(x)\|\,,\,\|\pi_{i+1}(\overline{x})\|\}}\,.$$

Die letzte Gleichung folgt aus der Invarianz der Gitterdeterminante $\|\pi_i(x)\|\,\|\widehat{b}_{i,x}\| = |\det(\pi_i(x)\,,\,\pi_i(b_i))| = \|\widehat{b}_i\|\,\|\pi_{i+1}(x)\|$. Aus $\|\pi_n(x)\| \leq \|\pi_{i+1}(x)\|$, $\|\pi_i(x) - \pi_i(\overline{x})\| = \|\pi_i(x - \overline{x})\| \leq \|x - \overline{x}\|$ für $i = 1, \ldots, n-1$ folgt daher

$$\|\widehat{b}_{i,x} - \widehat{b}_{i,\overline{x}}\| \;\leq\; 2\,\|\widehat{b}_{i,x}\|\,\frac{\|x - \overline{x}\|}{\|\pi_n(x)\|}\,.$$

3. Wegen $\widehat{b}_i$, $\pi_i(x)$, $\pi_i(x') \in span(b_1,\ldots,b_{i-1})^{\perp}$ ist $<\widehat{b}_i,\pi_i(x)> = <\widehat{b}_i,\pi_i(x')>$, $i=1,\ldots,n-1$ und $\widehat{b}_{i,x} = \widehat{b}_i - \frac{<\widehat{b}_i,\pi_i(x)>}{\|\pi_i(x)\|^2}\,\pi_i(x)$. Wir erhalten

$$\|\widehat{b}_{i,x}-\widehat{b}_{i,x'}\| \;=\; |<\widehat{b}_i,\pi_i(x')>|\;\|\frac{\pi_i(x)}{\|\pi_i(x)\|^2}-\frac{\pi_i(x')}{\|\pi_i(x')\|^2}\|\,.$$

Mit der Gleichung $\|\frac{b}{\|b\|^2}-\frac{c}{\|c\|^2}\| = \frac{\|b-c\|}{\|b\|\,\|c\|}$ und der Cauchy–Schwarz Ungleichung folgt für $i=1,\ldots,n-1$:

$$\|\widehat{b}_{i,x}-\widehat{b}_{i,x'}\| \;\le\; \frac{\|\widehat{b}_i\|\,\|\pi_i(x')\|\,\|\pi_i(x)-\pi_i(x')\|}{\|\pi_i(x)\|\,\|\pi_i(x')\|} \;=\; \frac{\|\widehat{b}_i\|\,\|\pi_n(x)\|}{\|\pi_i(x)\|} \;=\; \frac{\|\widehat{b}_{i,x}\|\,\|\pi_n(x)\|}{\|\pi_{i+1}(x)\|} \;\le\; \|\widehat{b}_{i,x}\|\,. \qquad \square$$

Satz 2.3 *Seien $x\in\mathbb{R}^n$, $\epsilon>0$ die Eingaben für SIRA. Dann gilt für die Ausgaben $x'\in\mathbb{R}^n$ und $m\in\mathbb{Z}^n-\{0\}$ von SIRA:*

1. Für alle $\overline{x}\in\mathbb{R}^n$ mit $\|x-\overline{x}\| < \|x-x'\|/2$ ist $\lambda(\overline{x}) \ge 1/(2\,\epsilon)$.

2. Der letzte duale Basisvektor a_n erfüllt während des gesamten Algorithmus stets $\|a_n\| \le 2^{n/2}\min\{\epsilon^{-1},\ \lambda(x)\}$.

3. $\|m\| \le 2^{n/2+1}\,\lambda(x')$.

Beweis. 1. Für jeden Punkt $\overline{x}\in\mathbb{R}^n$ mit $\|\overline{x}-x\| < \|\pi_n(x)\|/2$ gilt nach Lemma 2.2(2), daß $\|\widehat{b}_{i,x}-\widehat{b}_{i,\overline{x}}\| < \|\widehat{b}_{i,x}\|$, $i=1,\ldots,n-1$ und somit

$$\|\widehat{b}_{i,\overline{x}}\| \;>\; 0 \quad\text{sowie}\quad \|\widehat{b}_{i,\overline{x}}\| \;<\; 2\,\epsilon\,,\; i=1,\ldots,n-1\,.$$

Mit dem in [HJLS89] bewiesenen Lemma 1.28 folgt damit $\lambda(\overline{x}) \ge 1/(2\,\epsilon)$ für alle $\overline{x}\in\mathbb{R}^n$ mit $\|x-\overline{x}\| < \|x-x'\|/2$.

2. Seien $\overline{b}_1,\ldots,\overline{b}_n$, $\overline{a}_1,\ldots,\overline{a}_n$ die dualen Basen vor, $b_1,\ldots,b_n$, $a_1,\ldots,a_n$ die dualen Basen nach einem beliebigen Austausch $b_{n-1}\leftrightarrow b_n$ von SIRA. Seien $\overline{\mu}_{i,j}$ und $\widehat{\overline{b}}_{i,x}$ die entsprechenden Gram–Schmidt Koeffizienten bzw. Höhen von $x,\overline{b}_1,\ldots,\overline{b}_n$. Es gilt $\widehat{b}_{n-1,x} = \overline{\mu}_{n,n-1}\,\widehat{\overline{b}}_{n-1,x}$ und $|\overline{\mu}_{n,n-1}| \le \frac{1}{2}$.

Aus $<a_{n-1},b_i> = \delta_{n-1,i}$, $i=1,\ldots,n$ ergibt sich folgende Darstellung von a_{n-1} während des gesamten Algorithmus

$$a_{n-1} \;=\; \frac{\widehat{b}_{n-1,x}}{\|\widehat{b}_{n-1,x}\|^2} - \frac{<b_n,\widehat{b}_{n-1,x}>}{\|\widehat{b}_{n-1,x}\|^2}\,a_n\;.$$

Angewendet auf die Vektoren $\overline{b}_1,\ldots,\overline{b}_n$, $\overline{a}_1,\ldots,\overline{a}_n$, erhalten wir mit $|\overline{\mu}_{n,n-1}| \le \frac{1}{2}$ folgende Rekursionsformel für Austausche $b_{n-1}\leftrightarrow b_n$:

$$\begin{aligned}\|a_n\| \;=\; \|\overline{a}_{n-1}\| \;&\le\; \|\widehat{\overline{b}}_{n-1,x}\|^{-1} + |\overline{\mu}_{n,n-1}|\,\|\overline{a}_n\|\\ &\le\; \|\widehat{\overline{b}}_{n-1,x}\|^{-1} + \tfrac{1}{2}\,\|\overline{a}_n\|\,.\end{aligned}$$

Wegen der L^3–Reduziertheit der projizierten Vektoren $\pi_x(\overline{b}_1),\ldots,\pi_x(\overline{b}_{n-1})$ folgt mit Satz 1.22 (1) und Lemma (1.28)

$$2^{(n-1)/2}\,\|\widehat{\overline{b}}_{n-1,x}\| \;\overset{Satz\ 1.22(1)}{\ge}\; \max_{1\le i\le n} 2^{i/2}\,\|\widehat{\overline{b}}_{i,x}\| \;\overset{Lemma\ 1.28}{\ge}\; 2^{1/2}\,\lambda(x)^{-1}\;.$$

Wegen der Korrektheitsbedingung (1) von Lemma 2.1 gilt außerdem $\|\widehat{\overline{b}}_{n-1,x}\|^{-1} \leq 2^{n/2-1}\,\epsilon^{-1}$ und somit insgesamt $\|\widehat{\overline{b}}_{n-1,x}\|^{-1} \leq 2^{n/2-1} \min\{\epsilon^{-1}\,,\ \lambda(x)\}$. Für obige Rekursionsformel erhalten wir daher

$$\|a_n\| \quad\leq\quad 2^{n/2-1} \min\{\epsilon^{-1}\,,\ \lambda(x)\} + \tfrac{1}{2}\,\|\overline{a}_n\|\,. \tag{2.2}$$

Diese Ungleichung gilt für jeden Austausch $b_{n-1} \leftrightarrow b_n$ von SIRA. Angenommen, SIRA führt genau t solche Austausche durch. Da a_n zu Beginn von SIRA der n–te Einheitsvektor e_n und damit anfangs $\|a_n\| = 1$ ist, ergibt sich für die Auflösung der Rekursionsformel (2.2):

$$\|a_n\| \quad\leq\quad 2^{n/2-1} \min\{\epsilon^{-1}\,,\ \lambda(x)\} \sum_{j=0}^{t-1} 2^{-j} + 2^{-t} \quad\leq\quad 2 \cdot 2^{n/2-1} \min\{\epsilon^{-1}\,,\ \lambda(x)\}\,. \tag{2.3}$$

Der Vektor a_n wird zwischen einem Austausch $b_{n-1} \leftrightarrow b_n$ nicht verändert. Somit ist Ungleichung (2.3) während des gesamten Algorithmus erfüllt.

3. Wir betrachten den Fall der Ausgabe $m = a_n$ und $x' \neq x$. Nach Lemma 2.2 (3) gelten für die Endbasisvektoren $b_1, \ldots, b_n$ von SIRA für $x' = x - \pi_n(x)$ die Ungleichungen

$$\|\widehat{b}_{i,x} - \widehat{b}_{i,x'}\| \quad\leq\quad \|\widehat{b}_{i,x}\|\,,\ i = 1, \ldots, n-1\,,$$

und damit insbesondere

$$\|\widehat{b}_{i,x'}\| \quad\leq\quad 2\,\|\widehat{b}_{i,x}\|\,,\ i = 1, \ldots, n-1\,. \tag{2.4}$$

Aus $\lambda(x') \geq 1/\max_{1\leq i\leq n} \|\widehat{b}_{i,x'}\|$ (Lemma 1.28), $\|a_n\| = \|\widehat{b}_{n,x'}\|^{-1}$ und $\|\widehat{b}_{n-1,x'}\| = 0$ folgt daher

$$\frac{\|a_n\|}{\lambda(x')} \quad\leq\quad \max_{1\leq i\leq n} \frac{\|\widehat{b}_{i,x'}\|}{\|\widehat{b}_{n,x'}\|} \quad=\quad \max_{1\leq i\leq n-2} \{\,1,\ \|\widehat{b}_{i,x'}\|\,\|a_n\|\,\}\,.$$

Falls SIRA mit $x' \neq x$ terminiert, gilt nach Lemma 2.1 (1) $\|\widehat{b}_{i,x}\| \leq \epsilon$. Hieraus und aus (2.4) schließen wir

$$\frac{\|a_n\|}{\lambda(x')} \quad\leq\quad \max\{1,\ 2\,\epsilon\,\|a_n\|\} \quad\overset{(2.3)}{\leq}\quad \max\{1,\ 2\,\epsilon\, 2 \cdot 2^{n/2-1}\,\epsilon^{-1}\} \quad=\quad 2^{n/2+1}\,,$$

was die behauptete Ungleichung ist. □

Bemerkung. *Der Beweis von Satz 2.3 (2) bleibt korrekt, wenn man statt ϵ die Länge $\|\widehat{\overline{b}}_{1,x}\|$ der Höhen $\widehat{\overline{b}}_{1,x}$ vor dem Austausch $b_{n-1} \leftrightarrow b_n$ einsetzt. Statt der Ungleichung $\|\widehat{\overline{b}}_{n-1,x}\|^{-1} \leq 2^{n/2-1}\,\epsilon^{-1}$ gilt dann $\|\widehat{\overline{b}}_{n-1,x}\|^{-1} \leq 2^{n/2-1}\,\|\widehat{\overline{b}}_{1,x}\|^{-1}$. Die Rekursionsformel (2.2) bleibt gültig, da $\|\widehat{b}_{1,x}\|$ zwischen Austauschen $b_{n-1} \leftrightarrow b_n$ nicht wächst: Austausche $b_1 \leftrightarrow b_2$ vermindern $\|\widehat{b}_{1,x}\|$ nach Lemma 1.24 (3) um den Faktor $\sqrt{\frac{3}{4}} < 1$.*

Wir folgern aus Satz 2.3, daß der von SIRA im Falle $x' \neq x$ ausgegebene Vektor a_n eine bis auf einen Faktor $2^{n/2+1}$ kürzeste Fast–Relation für x ist.

Korollar 2.4 *Falls SIRA mit der Ausgabe $x' \neq x$ stoppt, dann ist a_n, bis auf einen Faktor* $2^{n/2+1}$, *eine* kürzeste Fast–Relation *für x im folgenden Sinn:*

Gilt für einen Vektor $m \in \mathbb{Z}^n - \{0\}$ die Ungleichung $| < x, m/\|m\| > | < | < x, a_n/\|a_n\| > |/2$ dann ist $\|m\| \geq \|a_n\| \, 2^{-n/2-1}$.

Beweis. Setze $\overline{x} := x - < x, m/\|m\| > m/\|m\|$. Da $x' := x - < x, a_n/\|a_n\| > a_n/\|a_n\|$, ist $\|x - \overline{x}\| < \|x - x'\|/2$. Aus Satz 2.3(1), (2) folgt $\lambda(\overline{x}) \geq 1/(2\,\epsilon)$ und $\|a_n\| \leq 2^{n/2}\,\epsilon^{-1}$, respektive. Somit gilt $\|m\| \geq \lambda(\overline{x}) \geq \|a_n\| \, 2^{-n/2-1}$. □

Größe der dualen Basisvektoren.

Satz 2.5 *Seien $x \in \mathbb{R}^n$, $\epsilon > 0$ die Eingaben von SIRA. Nach der Größenreduktion der projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_n)$ in Schritt 2 von SIRA gilt für die dualen Basen $b_1, \ldots, b_n$ und $a_1, \ldots, a_n$*

1. $\|a_i\| \leq 1.5^{n-i} \left(\max_{i \leq j < n} \|\widehat{b}_{j,x}\|^{-1} + 2^{n/2} \min\{\epsilon^{-1} , \lambda(x)\}\right)$, $i = 1, \ldots, n-1$,
2. $\|b_i\| \leq 2^{n/2} \min\{\epsilon^{-1} , \lambda(x)\} \sum_{j=1}^{\min\{i,n-1\}} \prod_{\substack{k=1 \\ k \neq j}}^{n-1} \|\widehat{b}_{k,x}\|^{-1} + \sum_{j=1}^{i} \|\widehat{b}_{j,x}\|$, $i = 1, \ldots, n$.

Beweis. 1. Da SIRA nicht zuvor abgebrochen ist, gilt $\widehat{b}_{n,x} = \tau_{n,n} = 0$ und damit $\widehat{b}_{j,x} \neq 0$ für $j = 1, \ldots, n-1$. Mit den Gram–Schmidt Koeffizienten $\mu_{i,j}$ von $x, b_0, \ldots, b_n$ seien die Größen $\nu_{i,j}$ durch $(\nu_{i,j})_{1 \leq i,j \leq n} := (\mu_{i,j})^{-1}_{1 \leq i,j \leq n}$ definiert. Wir erhalten damit für $i = 1, \ldots, n$ folgende Darstellung des i–ten dualen Basisvektors:

$$a_i = \sum_{j=i}^{n-1} \nu_{j,i} \frac{\widehat{b}_{j,x}}{\|\widehat{b}_{j,x}\|^2} + \nu_{n,i}\, a_n \tag{2.5}$$

Beweis. Setzt man für a_i Formel (2.5) in das Skalarprodukt $< a_i, b_k >$ ein, so gilt in der Tat

$$< a_i, b_k > = < \sum_{j=i}^{n-1} \nu_{j,i} \frac{\widehat{b}_{j,x}}{\|\widehat{b}_{j,x}\|^2} + \nu_{n,i}\, a_n \,, \sum_{j=0}^{k} \mu_{k,j}\, \widehat{b}_{j,x} >$$

$$= \sum_{j=1}^{n-1} \nu_{j,i}\, \mu_{k,j} + \nu_{n,i} < a_n, b_k > = \delta_{i,k} - \nu_{n,i}\, \delta_{k,n} + \nu_{n,i}\, \delta_{n,k} = \delta_{i,k} \,. \qquad \diamond$$

Nach der Größenreduktion der projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_n)$ in Schritt 2 von SIRA ist nun $|\mu_{i,j}| \leq \frac{1}{2}$, $1 \leq j < i \leq n$. Wir benutzen folgendes

Fakt. Sei $(m_{i,j})_{1 \leq i,j \leq n}$ eine untere Dreiecksmatrix, so daß $m_{i,i} = 1$, $1 \leq i \leq n$, und $|m_{i,j}| \leq M$, $1 \leq j < i \leq n$. Dann ist die inverse Matrix $(v_{i,j})_{1 \leq i,j \leq n} := (m_{i,j})^{-1}_{1 \leq i,j \leq n}$ eine untere Dreiecksmatrix mit $v_{i,i} = 1$, $1 \leq i \leq n$, und $|v_{i,j}| \leq (1+M)^{i-j}$, $1 \leq j < i \leq n$.

Angewendet auf die Matrix $(\nu_{i,j})_{1 \leq i,j \leq n} := (\mu_{i,j})^{-1}_{1 \leq i,j \leq n}$, erhalten wir

$$|\nu_{i,j}| \leq 1.5^{i-j} \,, \; 1 \leq j \leq i \leq n \,.$$

Gleichung (2.5) liefert somit für $i = 1, \ldots, n$

$$\|a_i\|^2 \;\; \leq \;\; 1.5^{2(n-i)} \max_{i \leq j < n} \|\widehat{b}_{j,x}\|^{-2} \; + \; 1.5^{2(n-i)} \, \|a_n\|^2 \, .$$

Mit der in Satz 2.3 (2) bewiesenen Ungleichung $\|a_n\| \leq 2^{n/2} \min\{\epsilon^{-1}, \; \lambda(x)\}$ folgt damit die erste Behauptung:

$$\|a_i\| \;\; \leq \;\; 1.5^{n-i} \, (\max_{i \leq j < n} \|\widehat{b}_{j,x}\|^{-1} + 2^{n/2} \min\{\epsilon^{-1}, \; \lambda(x)\}) \, .$$

2. Wir schreiben Gleichung (2.5) in Matrizenform als

$$[a_1, \ldots, a_n] \;\; = \;\; \left[\frac{\widehat{b}_{1,x}}{\|\widehat{b}_{1,x}\|^2}, \ldots, \frac{\widehat{b}_{n-1,x}}{\|\widehat{b}_{n-1,x}\|^2} \, , \, a_n \right] \, (\nu_{i,j})_{1 \leq i,j \leq n} \; .$$

Die Vektoren $x, \widehat{b}_{1,x}, \ldots, \widehat{b}_{n-1,x}$ sind paarweise orthogonal. Daher existiert eine orthogonale Matrix U, das heißt $U^{-1} = U^{\top}$, so daß

$$\left[\frac{\widehat{b}_{1,x}}{\|\widehat{b}_{1,x}\|^2}, \ldots, \frac{\widehat{b}_{n-1,x}}{\|\widehat{b}_{n-1,x}\|^2} \, , \, a_n \right] \;\; =$$

$$U \begin{bmatrix} 1 & & 0 & a'_{n,1} \\ & \ddots & & \vdots \\ 0 & & 1 & a'_{n,n-1} \\ 0 & \ldots & 0 & a'_{n,n} \end{bmatrix} \begin{bmatrix} \|\widehat{b}_{1,x}\|^{-1} & & 0 & \vdots \\ & \ddots & & 0 \\ 0 & & \|\widehat{b}_{n-1,x}\|^{-1} & \vdots \\ \ldots & 0 & \ldots & 1 \end{bmatrix} ,$$

mit $a'_n := (a'_{n,1}, \ldots, a'_{n,n})^{\top} := U \, a_n$ und $a'_{n,n} = \|\widehat{b}_{1,x}\| \cdot \ldots \cdot \|\widehat{b}_{n-1,x}\|$.

(U bildet die orthonormierten Vektoren $\widehat{b}_{1,x}/\|\widehat{b}_{1,x}\|, \ldots, \widehat{b}_{n-1,x}/\|\widehat{b}_{n-1,x}\|, x/\|x\|$, in die Einheitsvektoren $e_1, \ldots, e_n$ ab und erhält die Skalarprodukte $a'_{n,i} = < a'_n, e_i > = < U \, a_n, U \, \widehat{b}_{i,x}/\|\widehat{b}_{i,x}\| > = \; < a_n, \widehat{b}_{i,x}/\|\widehat{b}_{i,x}\| >$, $i = 1, \ldots, n-1$ sowie $a'_{n,n} = < a'_n, e_n > = < U \, a_n, U \, x/\|x\| > = < a_n, x/\|x\| > = \prod_{j=1}^{n-1} \|\widehat{b}_{j,x}\|$ (siehe Gleichung (2.13).)

Wegen $[b_1, \ldots, b_n]^{\top} = [a_1, \ldots, a_n]^{-1}$ folgt

$$[b_1, \ldots, b_n] \;\; =$$

$$(U^{-1})^{\top} \left(\begin{bmatrix} 1 & & 0 & a'_{n,1} \\ & \ddots & & \vdots \\ 0 & & 1 & a'_{n,n-1} \\ 0 & \ldots & 0 & a'_{n,n} \end{bmatrix}^{-1} \right)^{\top} \begin{bmatrix} \|\widehat{b}_{1,x}\|^{-1} & & 0 & \vdots \\ & \ddots & & 0 \\ 0 & & \|\widehat{b}_{n-1,x}\|^{-1} & \vdots \\ \ldots & 0 & \ldots & 1 \end{bmatrix}^{-1} ((\nu_{i,j})^{-1}_{1 \leq i,j \leq n})^{\top}$$

und daher

$$[b_1, \ldots, b_n] \;\; =$$

$$U \begin{bmatrix} 1 & & 0 & \vdots \\ & \ddots & & 0 \\ 0 & & 1 & \vdots \\ \overline{a}_{n,1} & \dots & \overline{a}_{n,n-1} & \overline{a}_{n,n} \end{bmatrix} \begin{bmatrix} \|\widehat{b}_{1,x}\| & & 0 & \vdots \\ & \ddots & & 0 \\ 0 & & \|\widehat{b}_{n-1,x}\| & \vdots \\ \dots & 0 & \dots & 1 \end{bmatrix} (\mu_{i,j})^{\top}_{1\le i,j\le n} \;, \tag{2.6}$$

mit $\overline{a}_{n,n} := a'^{-1}_{n,n}$ und $\overline{a}_{n,i} := -a'_{n,i}/a'_{n,n}$ für $i < n$. Wegen der Orthogonalität von U ist $\|U^{-1}\, b_i\| = \|b_i\|$ und daher $\|b_i\|$ genau die Länge des Spaltenvektors der Matrix, die als Matrixprodukt den Kofaktor von U in obiger Gleichung (2.6) darstellt. Aus $\overline{a}_{n,n} = \|\widehat{b}_{1,x}\|^{-1} \cdot \ldots \cdot \|\widehat{b}_{n-1,x}\|^{-1}$ und $\|a'_n\| = \|U\, a_n\| \;=\; \|a_n\|$ folgt für $i = 1, \ldots, n-1$

$$\begin{aligned} \|b_i\|^2 &= a'^{-2}_{n,n} \left(\sum_{j=1}^{i} a'_{n,j}\, \mu_{i,j}\, \|\widehat{b}_{j,x}\| \right)^2 + \sum_{j=1}^{i} \mu^2_{i,j}\, \|\widehat{b}_{j,x}\|^2 \\ &\le \|a_n\|^2 \sum_{j=1}^{i} \mu^2_{i,j} \prod_{\substack{k=1\\k\neq j}}^{n-1} \|\widehat{b}_{k,x}\|^{-2} + \sum_{j=1}^{i} \mu^2_{i,j}\, \|\widehat{b}_{j,x}\|^2 \;, \end{aligned} \tag{2.7}$$

und für $i = n$

$$\begin{aligned} \|b_n\|^2 &= a'^{-2}_{n,n} \left(\sum_{j=1}^{n-1} a'_{n,j}\, \mu_{n,j}\, \|\widehat{b}_{j,x}\| + 1 \right)^2 \\ &\le \|a_n\|^2 \sum_{j=1}^{n-1} \mu^2_{n,j} \prod_{\substack{k=1\\k\neq j}}^{n-1} \|\widehat{b}_{k,x}\|^{-2} \;. \end{aligned} \tag{2.8}$$

Wegen $|\mu_{i,j}| \;\le\; 1\,,\; 1 \le j \le i \le n$ erhalten wir

$$\|b_i\|^2 \;\le\; \|a_n\|^2 \sum_{j=1}^{i} \prod_{\substack{k=1\\k\neq j}}^{\min\{i,n-1\}} \|\widehat{b}_{k,x}\|^{-2} + (1-\delta_{i,n}) \sum_{j=1}^{i} \|\widehat{b}_{j,x}\|^2 \;,\;\; i = 1, \ldots, n \;. \tag{2.9}$$

Die obere Schranke von $\|b_i\|$ ergibt sich nunmehr mit der Ungleichung $\|a_n\| \le 2^{n/2} \min\{\epsilon^{-1}\,,\; \lambda(x)\}$ von Satz 2.3 (2). □

Die im Beweis von Satz 2.5 bewiesenen Ungleichungen (2.7) und (2.8) gelten während des gesamten Algorithmus. Von Satz 2.3 (2) wissen wir, daß die Ungleichung $\|a_n\| \le 2^{n/2} \min\{\epsilon^{-1}\,,\; \lambda(x)\}$ ebenso für alle Berechungsschritte von SIRA gültig ist. Somit erhalten wir aus obigem Beweis ebenfalls:

Proposition 2.6 *Während der Berechnungsschritte von SIRA erfüllen die Basisvektoren $b_1, \ldots, b_n$ für $i = 1, \ldots, n$ stets*

$$\|b_i\| \;\le\; 2^{n/2} \min\{\epsilon^{-1}\,,\; \lambda(x)\} \sum_{j=1}^{\min\{i,n-1\}} |\mu_{i,j}| \prod_{\substack{k=1\\k\neq j}}^{n-1} \|\widehat{b}_{k,x}\|^{-1} + (1-\delta_{i,n}) \left(\sum_{j=1}^{i} \mu^2_{i,j}\, \|\widehat{b}_{j,x}\|^2 \right)^{1/2} .$$

Approximation von x durch den Nahebeipunkt. Wir geben eine obere und untere Schranke für den euklidischen Abstand der Punkte x und x' an.

Proposition 2.7 *SIRA stoppe auf Eingaben $x \in \mathbb{R}^n$, $\epsilon > 0$, mit Ausgaben x' und einer Relation a_n für x'. Dann gilt*

$$\|x - x'\| \quad \leq \quad \|x\| \, \epsilon^{n-1} \, / \, \|a_n\| \, . \tag{2.10}$$

Beweis. Seien $b_1, \ldots, b_n$ und $a_1, \ldots, a_n$ die dualen Endbasisvektoren und $m := a_n$. Im Falle $x = x'$ ist obige Behauptung trivial.

Sei also $x \neq x'$ und SIRA mit $\widehat{b}_{n,x} = 0$ abgebrochen. Dann sind die Vektoren $x, b_1, ..., b_{n-1}$ linear unabhängig und bilden eine Basis des Gitters $L = L(x, b_1, ..., b_{n-1})$. Die Determinante $d(L)$ ist das Volumen des durch die Basisvektoren von L erzeugten Parallelepipeds. Damit können wir $d(L)$ als das Produkt der Längen der jeweiligen Höhen der Gram–Schmidt Orthogonalisierung darstellen. Angewendet auf die geordneten Basen $x, b_1, ..., b_{n-1}$ und $b_1, ..., b_{n-1}, x$, erhalten wir

$$d(L(x, b_1, \ldots, b_{n-1})) \quad = \quad \|x\| \prod_{j=1}^{n-1} \|\widehat{b}_{j,x}\| \quad = \quad \left(\prod_{j=1}^{n-1} \|\widehat{b}_j\| \right) \|\pi_n(x)\| \, . \tag{2.11}$$

Während des Algorithmus SIRA bilden die Vektoren $b_1, ..., b_n$ stets eine Basis des Gitters $\mathbb{Z}^n$. Daher gilt

$$\det(L(b_1, \ldots, b_n)) \quad = \quad \prod_{j=1}^{n} \|\widehat{b}_j\| \quad = \quad 1$$

und somit

$$\|\widehat{b}_n\|^{-1} \quad = \quad \prod_{j=1}^{n-1} \|\widehat{b}_j\| \quad = \quad \prod_{j=1}^{n-1} \|\widehat{b}_{j,x}\| \, (\|x\| \, / \, \|\pi_n(x)\|) \, . \tag{2.12}$$

Wegen $\|\widehat{b}_{j,x}\| \leq \epsilon$, $j = 1, \ldots, n-1$ und $\|a_n\| = \|\widehat{b}_n\|^{-1}$ folgt daher

$$\|x - x'\| \quad = \quad \|\pi_n(x)\| \quad = \quad \frac{\|x\|}{\|a_n\|} \prod_{j=1}^{n-1} \|\widehat{b}_{j,x}\| \quad \leq \quad \|x\| \, \epsilon^{n-1} \, \|a_n\|^{-1} \, . \qquad \square$$

Proposition 2.8 *Für Eingaben $x := (p_1, \ldots, p_n)/q \in \mathbb{Q}^n$ mit $p_1, \ldots, p_n \in \mathbb{Z}$, $q \in \mathbb{N}$ und $\epsilon \in \mathbb{Q}_+$ führt SIRA höchstens $\lfloor \log_2 |p_n| \rfloor$ Austausche $b_{n-1} \leftrightarrow b_n$, aus. Außerdem gilt im Falle der Ausgabe $x' \neq x$*

$$\|x - x'\| \quad \geq \quad 2^{-n/2} \, \epsilon \, q^{-1} \, .$$

Beweis. Wie im Beweis von Proposition 2.7 sieht man, daß die Vektoren $x, b_1, \ldots, b_{n-1}$ vor einem Austausch $b_{n-1} \leftrightarrow b_n$ linear unabhängig sind und ein Parallelepiped mit nicht verschwindendem Volumen erzeugen. Es ist:

$$\|x\| \prod_{j=1}^{n-1} \|\widehat{b}_{j,x}\| \quad = \quad \prod_{j=1}^{n-1} \|\widehat{b}_j\| \, \|\pi_n(x)\| \quad = \quad \|\widehat{b}_n\|^{-1} \, \|\pi_n(x)\| \quad = \quad \|a_n\| \, \|\pi_n(x)\| \, .$$

und wegen $a_n = \widehat{b}_n / \|\widehat{b}_n\|^2$ somit

$$\prod_{j=1}^{n-1} \|\widehat{b}_{j,x}\| \quad = \quad | < x, a_n > | \, \|x\|^{-1} \, . \tag{2.13}$$

Diese Gleichung gilt für die dualen Basen $\overline{a}_1, \ldots, \overline{a}_n$, $\overline{b}_1, \ldots, \overline{b}_n$ vor, und für die dualen Basen $a_1, \ldots, a_n$, $b_1, \ldots, b_n$ nach dem Austausch. Wir erhalten

$$\frac{\|\widehat{b}_{n-1,x}\|}{\|\widehat{\overline{b}}_{n-1,x}\|} \;=\; \frac{\prod_{j=1}^{n-1} \|\widehat{b}_{j,x}\|}{\prod_{j=1}^{n-1} \|\widehat{\overline{b}}_{j,x}\|} \;=\; \frac{| < x, a_n > |}{| < x, \overline{a}_n > |} \, .$$

Aus $\|\widehat{b}_{n-1,x}\| \, = |\overline{\mu}_{n,n-1}| \, \|\widehat{\overline{b}}_{n-1,x}\| \, \leq \frac{1}{2} \, \|\widehat{\overline{b}}_{n-1,x}\|$ folgt

$$| < x, a_n > | \quad \leq \quad \tfrac{1}{2} \, | < x, \overline{a}_n > | \, .$$

Sei t die Anzahl der Austausche $b_{n-1} \leftrightarrow b_n$. Zu Beginn von SIRA war $< x, a_n >= p_n/q$ und bei Abbruch von SIRA ist

$$q^{-1} \quad \leq \quad | < x, a_n > | \; \leq \; 2^{-t} \, |p_n| \, q^{-1} \, . \tag{2.14}$$

Somit führt SIRA $t \leq \lfloor \log_2 |p_n| \rfloor$ Austausche $b_{n-1} \leftrightarrow b_n$ aus. Aus (2.14) und Lemma 2.3 (2) folgt weiterhin

$$\|x - x'\| \;=\; \frac{| < x, a_n > |}{\|a_n\|} \;\geq\; \frac{1}{q \, \|a_n\|} \;\geq\; 2^{-n/2} \, \epsilon \, q^{-1} \, . \qquad \Box \tag{2.15}$$

Proposition 2.7 und Ungleichung (2.15) zeigen, daß für Eingaben $x = (p_1, \ldots, p_n)/q \in \mathbb{Q}^n$, $p_1, \ldots, p_n \in \mathbb{Z}$, $q \in \mathbb{N}$ und $\epsilon \in \mathbb{Q}_+$ im Falle der Ausgabe $x' \neq x$ von SIRA der euklidische Abstand von x' und x im Intervall $\|a_n\| \, [\frac{1}{q}, \|x\| \, \epsilon^{n-1}]$ liegen muß. Wir erhalten somit:

Korollar 2.9 *Für Eingaben $x = (p_1, \ldots, p_n)/q \in \mathbb{Q}^n$, $p_1, \ldots, p_n \in \mathbb{Z}$, $q \in \mathbb{N}$ und $\epsilon \in \mathbb{Q}_+$ mit $\|x\| \, < \, q \, \epsilon^{1-n}$ findet SIRA stets eine Relation a_n zum rationalen Eingabevektor x.*

Laufzeit. Wir können die Analyse des HJLS–Algorithmus zur Abschätzung der Austausche $b_{k-1} \leftrightarrow b_k$ auf SIRA analog übertragen (Beweis von Satz 1.30). SIRA führt höchstens $\binom{n}{2}(\lceil \log_{4/3} 2 \rceil \, n + 2 \, \lceil | \log_2 \epsilon | \rceil) \, \leq \binom{n}{2}(3 \, n + 2 \, \lceil | \log_2 \epsilon | \rceil)$ Austausche $b_{k-1} \leftrightarrow b_k$ aus. Nach Lemma 1.29 kostet jeder Austausch $b_{k-1} \leftrightarrow b_k$ und die vorausgehende Reduktion jeweils höchstens $O(n)$ arithmetische Operationen. Für die Berechnung der Gram–Schmidt Größen $\tau_{i,i}^2 \, = \|\widehat{b}_{i,x}\|^2$, $i = 1, \ldots, n$ sowie $\mu_{i,j} \, = \tau_{i,j}/\tau_{j,j}$, $0 \leq i, j \leq n$ verwenden wir das Verfahren der Gram–Schmidt Orthogonalisierung. Dann kostet die Berechnung der Gram–Schmidt Größen zu Beginn des HJLS–Algorithmus $O(n^3)$ arithmetische Operationen und deren Aktualisierung nach einem Austausch $b_{k-1} \leftrightarrow b_k$ mit vorausgehender Reduktion gemäß Lemma 1.29 jeweils höchstens $O(n)$ arithmetische Operationen. Die Größenreduktion aller projizierter Vektoren $\pi_x(b_1), \ldots, \pi_x(b_n)$ kostet $O(n^3)$ arithmetische Operationen inklusive der Neuberechnung aller veränderten Gram–Schmidt Koeffizienten $\mu_{i,j}$, $1 \leq j < i \leq n$. Die Invertierung der Matrix $[b_1, \ldots, b_n]$ der Endbasisvektoren zur Berechnung von a_n erfordert nach der Schulmethode $O(n^3)$ arithmetische Operationen. Wir beschränken die Anzahl der Austausche $b_{n-1} \leftrightarrow b_n$ (sehr grob) durch die Anzahl aller Austausche. Dann ist die Gesamtanzahl der arithmetischen Operationen durch $O(n^3 \binom{n}{2}(3\,n + 2\,\lceil|\log\epsilon|\rceil)) + O(n^3) = O(n^5\,(n + \lceil|\log\epsilon|\rceil))$ beschränkt. Wir erhalten daher insgesamt mit Satz 2.3:

Korollar 2.10 *Auf Eingaben $x \in \mathbb{R}^n$ und $\epsilon > 0$ berechnet SIRA in $O(n^5\,(n + \lceil|\log\epsilon|\rceil))$ arithmetischen Operationen auf reellen Zahlen einen Punkt x' und eine Relation m zu x', so daß folgendes gilt:*

1. Für alle $\overline{x} \in \mathbb{R}^n$ mit $\|x - \overline{x}\| < \|x - x'\|/2$ ist $\lambda(\overline{x}) \geq 1/(2\,\epsilon)$.

2. $\|m\| \leq 2^{n/2} \min\{\epsilon^{-1}\,,\ \lambda\,,\ 2\,\lambda(x')\}$.

Bemerkung. *Die Laufzeit von SIRA bleibt $O(n^5\,(n + \lceil|\log\epsilon|\rceil))$, wenn wir im Test der L^3–Bedingung in Schritt 2 den Faktor $\frac{3}{4}$ durch eine beliebige Konstante $\delta \in (\frac{1}{2}, 1)$ ersetzen. Die Länge des letzten dualen Basisvektors ist dann während des gesamten Algorithmus durch $2\,(\delta - \frac{1}{4})^{-(n/2-1)} \min\{\epsilon^{-1}\,,\ \lambda(x)\,,\ 2\,\lambda(x')\}$ beschränkt. Mit dieser oberen Schranke erhalten wir in den Sätzen 2.3 und 2.5, in den Korollaren 2.4, 2.9 und 2.10, in Propositionen 2.6 und 2.8 Schranken, in denen der Faktor $2^{n/2-1}$ jeweils durch $(\delta - \frac{1}{4})^{-(n/2-1)}$ zu ersetzen ist.*

2.3 Analyse des Relationenalgorithmus in rationaler Arithmetik

In diesem Abschnitt geben wir eine Variante von SIRA an, die für rationale Eingaben $x \in \mathbb{Q}^n$, $\epsilon \in \mathbb{Q}_+$ polynomial–Zeit in der binären Länge der Eingabe ist. Für rationale Eingaben $x \in \mathbb{Q}^n$, $\epsilon \in \mathbb{Q}_+$ können wir für SIRA selbst keine polynomiale Bitkomplexität beweisen. Dies liegt daran, daß die Zähler und Nenner der Gram–Schmidt Koeffizienten $\mu_{i,j}$ während der L^3–Reduktion der projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_{n-1})$ beliebig groß werden können. Damit erhalten wir auch keine obere Schranke für die Länge der primären Basisvektoren $b_1, \ldots, b_n$ nach den Größenreduktionsschritten während der L^3–Reduktion der projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_{n-1})$.

Methode. Wir führen in der L^3–Reduktion im 2. Schritt von SIRA vor jedem Austausch $b_{k-1} \leftrightarrow b_k$ die Größenreduktion des Vektors b_k nicht nur bezüglich b_{k-1}, sondern vollständig, das heißt bezüglich aller vorangehenden Basisvektoren $b_{k-1}, \ldots, b_1$, durch. Zusätzlich reduzieren wir alle projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_n)$ im 1. Schritt von SIRA vollständig. Damit bleibt die binäre Länge der Zähler und Nenner der Gram–Schmidt Koeffizienten während der L^3–Reduktion in Schritt 2 polynomial in der Bitlänge der Eingaben beschränkt. Mit Proposition 2.6 erhalten wir für die binäre Länge der Einträge der primären Basisvektoren ebenfalls eine obere Schranke, die polynomial in der Bitlänge der Eingaben ist (2.15). Die Invertierung der Matrix $[b_1, \dots, b_n]$ am Ende des Algorithmus wird modulo einer genügend großen Primzahlpotenz ausgeführt. Dadurch bleibt die binäre Länge der in Schritt 4 von SIRA auftretenden ganzen Zahlen beschränkt.

Für die Orthonormalisierung von $x, b_1, \dots, b_n$ verwenden wir GSO und können damit die Analyse der Bitkomplexität von [LLL82] anwenden.

Die L^3–Reduktion der projizierten Vektoren $\pi_x(b_1), \dots, \pi_x(b_{n-1})$ kann sowohl mit der L^3–Austauschregel als auch mit der Bergman–Austauschregel erfolgen. In [HJLS89], Kapitel 6 wurde ein ähnlicher Relationenalgorithmus angegeben. [HJLS89] bewiesen die polynomiale Bitkomplexität dieses Algorithmus im Falle der Verwendung der L^3–Austauschregel. In Analogie zu SIRA nennen wir unseren Relationenalgorithmus für rationale Eingaben rat–SIRA. Für die L^3–Reduktion der projizierten Vektoren $\pi_x(b_1), \dots, \pi_x(b_{n-1})$ lassen wir anstatt $\frac{3}{4}$ jede beliebige Konstante $\delta \in (\frac{1}{2}, 1)$ zu. Man überzeugt sich leicht, daß die in Abschnitt 2.1 und 2.2 für den Algorithmus SIRA bewiesenen Resultate für den Algorithmus rat–SIRA entsprechend gelten.

Rationaler Stabiler Relationenalgorithmus (rat–SIRA)

EINGABE $x := (p_1, \dots, p_n)/q \in \mathbf{Q}^n - \{0\}$ mit $p_1, \dots, p_n$, $q \in \mathbb{Z}$, $\epsilon \in \mathbf{Q}_+$, $\delta \in (\frac{1}{2}, 1)$.

1. Initialisierung. $[b_0, b_1, \dots, b_n] :=[x, e_1, \dots, e_n]$, $s := 1$,

berechne die Orthonormalisierung $(\tau_{i,j})_{\substack{0 \le i \le n \\ 1 \le j \le n}}$ von $[x, b_1, \dots, b_n]$.

Falls $\tau_{n,n} > 0$ dann ist e_n eine Relation für x. Gebe den Punkt $x' := x$ und die Relation $a_n := e_n$ für x aus, und stoppe.

Größenreduziere alle projizierten Vektoren $\pi_x(b_1), \dots, \pi_x(b_n)$.

2. *L^3–Reduktion von* $\pi_x(b_1), \dots, \pi_x(b_{n-1})$.

WHILE $(\exists 1 < k < n \ : \delta\, \tau_{k-1,k-1}^2 \ > \tau_{k,k}^2 + \tau_{k,k-1}^2 \ )$ DO

FOR $j = k-1, \dots, 1$ DO $b_k := b_k - \lceil \tau_{k,j}/\tau_{j,j} \rfloor\, b_j$;

vertausche b_{k-1} , b_k und berechne die Orthonormalisierung $(\tau_{i,j})_{\substack{0 \le i \le n \\ 1 \le j \le n}}$ neu.

Größenreduziere alle projizierten Vektoren $\pi_x(b_1), \dots, \pi_x(b_n)$.

WHILE $|\tau_{s,s}| \le \epsilon$ DO $s := s + 1$.

3. *Austausch* $b_{n-1} \leftrightarrow b_n$.

Vertausche b_{n-1} und b_n und berechne die Orthonormalisierung $(\tau_{i,j})_{\substack{0 \le i \le n \\ 1 \le j \le n}}$ neu.

Falls $\tau_{n,n} = 0$ und $s < n$ dann gehe nach 2.

4. *Abbruch.* Berechne $[a_1, \dots, a_n]^\top :=[b_1, \dots, b_n]^{-1}$.

Falls $\tau_{n,n} > 0$ dann ist eine Relation für x gefunden. Gebe den Punkt $x' := x$ und die Relation a_n für x aus, und stoppe.

Falls $s = n$, dann gilt $\tau_{i,i} \le \epsilon$ für $i = 1, \dots, n$, und es existiert keine Relation für x mit Länge kleiner als ϵ^{-1} . Berechne in diesem Fall $\pi_n(x) =< x, a_n/\|a_n\| > a_n/\|a_n\| \in span(b_1, \dots, b_{n-1})^\perp$, und gebe den Punkt $x' := x - \pi_n(x)$, die Relation a_n für x' sowie '$\lambda(x) \ge 1/\epsilon$' aus.

In der Analyse der Bitkomplexität von rat–SIRA zeigen wir obere Schranken für die im Algorithmus auftretenden ganzen Zahlen. Diese sind

- die Einträge der Vektoren $x = (p_1, \ldots, p_n)q, b_1, \ldots, b_n$ des Basisystems und die Einträge der Vektoren $a_1, \ldots, a_n$ der dualen Basis,
- die Zähler und Nenner der Höhenquadratlängen $\|x\|^2, \|\widehat{b}_{1,x}\|^2, \ldots, \|\widehat{b}_{n,x}\|^2$,
- die Zähler und Nenner der Gram–Schmidt Koeffizienten $\mu_{i,j}$, $0 \le j \le i \le n$.

Hierzu definieren wir für $j = 0, \ldots, n$ die Größen

$$d_j := \prod_{\substack{i=0 \\ \widehat{b}_{i,x} \neq 0}}^{j} \|\widehat{b}_{i,x}\|^2 \quad .$$

Nach Korollar 1.19 ist d_j die Determinante der Matrix $(< b_i, b_l >)_{\substack{0 \le i,l \le j \\ \widehat{b}_{l,x} \neq 0}}$ und daher rational mit Nenner q. Aus [LLL82], Formeln (1.28) und (1.29) wissen wir, daß $\|\widehat{b}_{j,x}\|^2 \in d_{j-1}^{-1}\,\mathbb{Z}$ bzw. $\mu_{i,j} \in d_j^{-1}\,\mathbb{Z}$. Alle in rat–SIRA auftretenden Zahlen sind daher rational mit maximalem Nenner $\max_{0 \le j \le n} d_j$. Nach Lemma 1.22 (3) wächst $\max_{1 \le i \le n} \|\widehat{b}_{i,x}\|$ während rat–SIRA nicht. Zu Beginn ist $\|\widehat{b}_{i,x}\| \le 1$ für alle $i = 1, \ldots, n$. Wegen $b_0 = x$ gilt daher

$$d_j \quad \le \quad \|x\|^2 \quad \le \quad \left(\sum_{i=1}^{n} p_i^2 \right) \quad , \; j = 0, \ldots, n \tag{2.16}$$

während des gesamten Algorithmus. Die Bitlänge der Zähler und Nenner der Höhenquadratlängen $\|x\|^2$, $\|\widehat{b}_{i,x}\|^2$, $1 \le i \le n$, ist damit während des gesamten Algorithmus durch $O(\sum_{j=1}^{n} \lceil \log |p_i| \rceil + \lceil \log |q| \rceil)$ beschränkt.

Wir benötigen im folgenden noch eine untere Schranke für nicht verschwindende Höhen $\widehat{b}_{i,x}$. Mit Gleichung (2.13) folgt

$$d_j \quad \ge \quad < x, a_n >^2 \; \ge \; q^{-2} \; , j = 0, \ldots, n-1 \; .$$

Nach Definition von d_j erhalten wir für alle nicht verschwindenden Höhen $\widehat{b}_{j,x}$ während des gesamten Algorithmus

$$\|\widehat{b}_{j,x}\| \quad = \quad d_j / d_{j-1} \;\; \ge \;\; (q\,\|x\|)^{-1} \quad = \quad \left(\sum_{i=1}^{n} p_i^2 \right)^{\frac{1}{2}} \quad , \; j = 1, \ldots, n-1 \; . \tag{2.17}$$

Folgendes Lemma beweist obere Schranken für die Größe der Gram–Schmidt Koeffizienten und die Länge der Basisvektoren vor und nach der Größenreduktion eines Vektors während der L^3–Reduktion der projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_{n-1})$ von rat–SIRA:

Lemma 2.11 1. *Bei Eintritt in die L^3–Reduktion von Schritt 2 ist* $|\mu_{i,j}| \le \frac{1}{2}$, $1 \le j < i \le n$.

2. *Sei k der Stufenindex für einen Austausch $b_{k-1} \leftrightarrow b_k$ während der L^3–Reduktion in Schritt 2. Dann gilt vor und nach der Größenreduktion von b_k :*

$$\|b_k\| \;\leq\; 1.28\,(\delta - \tfrac{1}{4})^{-(n/2-1)}\, n^{3/2} \left(\sum_{i=1}^{n} p_i^2\right)^{1+\frac{1}{2\,(n-1)}} .$$

Beweis. 1. Schritt 2 wird entweder von Schritt 1 aus oder von Schritt 3 aus erreicht. In beiden Fällen sind jedoch die projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_n)$ größenreduziert.

2. Bei Eintritt in die L^3–Reduktion von Schritt 2 gilt nach Teil 1 $|\mu_{i,j}| \;\leq\; \frac{1}{2}$, $1 \leq j < i \leq n$, sowie $\widehat{b}_{n,x} = 0$, da andernfalls rat–SIRA schon vorher abgebrochen wäre. Mit Proposition 2.6 und Gleichung (2.13) erhalten wir somit für $i = 1, \ldots, n$:

$$\begin{aligned}
\|b_i\| &\leq 2\,(\delta - \tfrac{1}{4})^{-(n/2-1)} \min\{\epsilon^{-1},\ \lambda(x)\} \sum_{j=1}^{\min\{i,n-1\}} |\mu_{i,j}| \prod_{\substack{k=1\\k\neq j}}^{n-1} \|\widehat{b}_{k,x}\|^{-1} + \left(\sum_{j=1}^{i} \mu_{i,j}^2\, \|\widehat{b}_{j,x}\|^2\right)^{1/2} \\
&\leq 2\,(\delta - \tfrac{1}{4})^{-(n/2-1)} \min\{\epsilon^{-1},\ \lambda(x)\}\, i\, \frac{\|x\|}{|< x, a_n >|} + \frac{\sqrt{i+1}}{2} .
\end{aligned}$$

Aus $< x, a_n > \neq 0$ folgt $|< x, a_n >| \geq q^{-1}$ und daher für $i = 1, \ldots, n$

$$\|b_i\| \;\leq\; 2\,(\delta - \tfrac{1}{4})^{-(n/2-1)} \min\{\epsilon^{-1},\ \lambda(x)\}\, n\, q\, \|x\| .$$

Diese Schranke gilt ebenso nach jeder Größenreduktion eines Vektors b_k, da alle Basisvektoren b_i, $i \neq k$ unverändert bleiben und $|\mu_{k,j}| \;\leq\; \frac{1}{2}$, $1 \leq j < k$. Somit gilt die gezeigte Schranke auch vor jeder Größenreduktion in der L^3–Reduktion von Schritt 2, denn Austausche $b_{k-1} \leftrightarrow b_k$ verändern die Länge der Basisvektoren nicht.

Mit Satz 1.27 erhalten wir für $i = 1, \ldots, n$:

$$\|b_i\| \;\leq\; 1.28\,(\delta - \tfrac{1}{4})^{-(n/2-1)}\, n^{3/2} \left(\sum_{i=1}^{n} p_i^2\right)^{\frac{1}{2}+\frac{1}{2\,(n-1)}} . \qquad \square$$

Setzt man im obigen Beweis für $\min\{\epsilon^{-1}, \lambda(x)\}$ die Größe ϵ^{-1} ein, so erhält man entsprechend vor und nach der Größenreduktion von b_k :

$$\|b_i\| \;\leq\; 2\,(\delta - \tfrac{1}{4})^{-(n/2-1)}\, \epsilon^{-1}\, n^{3/2} \left(\sum_{i=1}^{n} p_i^2\right)^{\frac{1}{2}} ,\ i = 1, \ldots, n . \tag{2.18}$$

Wir geben nun obere Schranken für die in den Größenreduktionsschritten von rat–SIRA auftretenden ganzen Zahlen an. Die Größenreduktionsschritte treten sowohl in der L^3–Reduktion als auch in der Größenreduktion der Schritte 1 und 3 auf.

Satz 2.12 *Seien $x := (p_1, \ldots, p_n)/q \in \mathbb{Q}^n$ $p_i \in \mathbb{Z}$, $i = 1, \ldots, n$, $q \in \mathbb{N}$, $\delta \in (\frac{1}{2}, 1)$ und $\epsilon \in \mathbb{Q}_+$ die Eingaben von rat–SIRA. Dann sind die Absolutbeträge der Gram–Schmidt Koeffizienten $\mu_{i,j}$, $0 \leq j < i \leq n$ und die Längen der Basisvektoren $b_1, \ldots, b_n$ während der Schritte 1–3 von rat–SIRA durch $(\delta - \frac{1}{4})^{-n^2/2}\, n^{3\,n/2}\, (\sum_{i=1}^{n} p_i^2)^{n+3}$ beschränkt.*

Beweis. Nach Lemma 2.11 gilt die Behauptung jeweils vor und nach der Größenreduktion eines Vektors.

Wir analysieren nun den Effekt der Größenreduktion auf diese Größen. Seien $b_k^{(l)}$, $\mu_{k,i}^{(l)}$ der Vektor b_k und die Gram–Schmidt Koeffizienten $\mu_{k,i}$, respektive, nach Ausführung von $b_k := b_k - \lceil \mu_{k,j} \rfloor b_j$ für $j = k-1, \ldots, k-l$. $b_k^{(0)}$ bezeichnet den Vektor b_k vor der Größenreduktion von b_k, $b_k^{(k-1)}$ den Vektor b_k nach Beendigung der Größenreduktion.

Lemma 2.13 *Sei* $|\mu_{i,j}| \leq M$ *für* $1 \leq j < i < k$ *bei Eintritt in die Größenreduktion von* b_k. *Dann gilt für* $i = k-l, \ldots, 1$

$$|\mu_{k,i}^{(l)}| \;\leq\; |\mu_{k,i}^{(0)}| + [(M+1)^l - 1](\tfrac{1}{2} + \max_{j=k-1,\ldots,k-l} |\mu_{k,j}^{(0)}|) \;.$$

Beweis. Wir beweisen durch Induktion nach l die Ungleichung

$$|\mu_{k,i}^{(l)}| \;\leq\; |\mu_{k,i}^{(0)}| + M \sum_{j=0}^{l-1} (M+1)^j (\tfrac{1}{2} + |\mu_{k,k-l+j}^{(0)}|) \;,\; l = 1, \ldots, k-1 \;.$$

Da $|\mu_{k-l,j}| \;\leq\; M$ für $1 \leq j < k-l$, ist

$$\begin{aligned} |\mu_{k,i}^{(l)}| &= |\mu_{k,i}^{(l-1)} - \lceil \mu_{k,k-l}^{(l-1)} \rfloor \mu_{k-l,i}| \\ &\leq |\mu_{k,i}^{(l-1)}| + M\,(\tfrac{1}{2} + |\mu_{k,k-1}^{(l-1)}|) \;, \end{aligned}$$

was die Behauptung im Falle $l = 1$ zeigt. Die Induktionsvoraussetzung für $l = 1$ auf die letzte Ungleichung angewendet, liefert

$$\begin{aligned} |\mu_{k,i}^{(l)}| &\leq |\mu_{k,i}^{(0)}| + M \sum_{j=0}^{l-2} (M+1)^j\,(\tfrac{1}{2} + |\mu_{k,k-l+j+1}^{(0)}|) \\ &+ M\,(\tfrac{1}{2} + |\mu_{k,k-l}^{(0)}| + M \sum_{j=0}^{l-2} (M+1)^j\,(\tfrac{1}{2} + |\mu_{k,k-l+j+1}^{(0)}|)) \\ &\leq |\mu_{k,i}^{(0)}| + M \sum_{j=1}^{l-1} (M+1)^j\,(\tfrac{1}{2} + |\mu_{k,k-l+j}^{(0)}|) + M\,(\tfrac{1}{2} + |\mu_{k,k-l}^{(0)}|) \\ &\leq |\mu_{k,i}^{(0)}| + M \sum_{j=0}^{l-1} (M+1)^j (\tfrac{1}{2} + |\mu_{k,k-l+j}^{(0)}|) \end{aligned}$$

Mit der geometrischen Summenformel ergibt sich die behauptete Ungleichung. □

Korollar 2.14 *Sei* $|\mu_{i,j}| \;\leq\; M$ *für* $1 \leq j < i < k$ *und* $M \;\geq\; \frac{1}{2}$ *bei Eintritt in die Größenreduktion von* b_k. *Dann gilt mit den oben eingeführten Bezeichnungen für* $l = 1, \ldots, k-1$:

$$\|b_k^{(l)}\| \;\leq\; \|b_k^{(0)}\| + \sum_{i=k-l}^{k-1} \|b_i^{(0)}\| (M+1)^l (\tfrac{1}{2} + \max_{k-l \leq j \leq k-1} |\mu_{k,j}^{(0)}|) \;.$$

Beweis. Für die vollständige Größenreduktion von b_k bezüglich $b_{k-1}, \ldots, b_{k-l}$ erhalten wir mit Lemma 2.13

$$\begin{aligned}
\|b_k^{(l)}\| &= \|b_k^{(0)} - \sum_{i=k-l}^{k-1} \mu_{k,i}^{(k-1-i)} b_i^{(0)}\| \leq \|b_k^{(0)}\| + \sum_{i=k-l}^{k-1} \|b_i^{(0)}\| \, |\mu_{k,i}^{(k-1-i)}| \\
&\leq \|b_k^{(0)}\| + \sum_{i=k-l}^{k-1} \|b_i^{(0)}\| \, [\, |\mu_{k,i}^{(0)}| + [(M+1)^{k-1-i} - 1] \, (\tfrac{1}{2} + \max_{j=k-1,\ldots,i+1} |\mu_{k,j}^{(0)}|) \,] \\
&\leq \|b_k^{(0)}\| + \sum_{i=k-l}^{k-1} \|b_i^{(0)}\| (M+1)^l (\tfrac{1}{2} + \max_{k-l \leq j \leq k-1} |\mu_{k,j}^{(0)}|) \, . \qquad \square
\end{aligned}$$

Wir kommen nun zum Beweis von Satz 2.12 und betrachten zuerst die Auswirkung der Größenreduktion während der L^3–Reduktion in Schritt 2 auf die Länge der Basisvektoren und die Größe der Gram–Schmidt Koeffizienten. Vor der Größenreduktion eines Vektors b_k gilt nach Lemma 2.11 (2) mit $(\delta - \frac{1}{4}) \geq 2$:

$$\|b_i\| \leq 1.28 \, (\delta - \tfrac{1}{4})^{-(n/2-1)} \, n^{3/2} \left(\sum_{i=1}^{n} p_i^2 \right)^{\frac{1}{2} + \frac{1}{2\,(n-1)}} \tag{2.19}$$

$$\leq (\delta - \tfrac{1}{4})^{-n/2} \, n^{3/2} \left(\sum_{i=1}^{n} p_i^2 \right)^{\frac{1}{2} + \frac{1}{2\,(n-1)}} . \tag{2.20}$$

Mit Ungleichung (2.17) erhalten wir daher für $1 \leq j < k$:

$$\begin{aligned}
|\mu_{k,j}| &= \frac{|< \pi_{j,x}(b_k), \widehat{b}_{j,x} >|}{\|\widehat{b}_{j,x}\|^2} \leq \frac{\|b_k\|}{\|\widehat{b}_{j,x}\|} \\
&\leq (\delta - \tfrac{1}{4})^{-n/2} \, n^{3/2} \left(\sum_{i=1}^{n} p_i^2 \right)^{1 + \frac{1}{2\,(n-1)}} .
\end{aligned} \tag{2.21}$$

Da diese obere Schranke auch für alle $\mu_{i,j}$, $1 \leq j < i < k$ gilt, folgt mit Lemma 2.13 und Korollar 2.14 für die Größe der Gram–Schmidt Koeffizienten $\mu_{k,j}$, $1 \leq j < k$, und die Länge des Vektors b_k während der vollständigen Größenreduktion von b_k für $j = k - l, \ldots, 1$:

$$|\mu_{k,j}^{(l)}| \leq (\delta - \tfrac{1}{4})^{-(l+1)\,n/2} \, n^{3\,(l+1)/2} \left(\sum_{i=1}^{n} p_i^2 \right)^{(l+1) + \frac{l+1}{2\,(n-1)}} , \tag{2.22}$$

$$\|b_k^{(l)}\| \leq (\delta - \tfrac{1}{4})^{-(l+2)\,n/2} \, n^{3\,(l+2)/2} \left(\sum_{i=1}^{n} p_i^2 \right)^{(l+1.5) + \frac{l+2}{2\,(n-1)}} . \tag{2.23}$$

Wir betrachten nun die Größenreduktion der projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_n)$ in den Schritten 1 und 3. Vor der ersten Größenreduktion in Schritt 1 gilt $\|b_i\| = 1$, $i = 1, \ldots, n$. Vor jeder anderen Größenreduktion der projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_n)$ ist nach Lemma 2.11 (2) für $i = 1, \ldots, n$:

$$\|b_k\| \leq (\delta - \tfrac{1}{4})^{-n/2} \, n^{3/2} \left(\sum_{i=1}^{n} p_i^2 \right)^{\frac{1}{2} + \frac{1}{2\,(n-1)}} .$$

Damit gilt vor der Größenreduktion von b_k während der Größenreduktion der projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_n)$, daß $|\mu_{i,j}| \leq \frac{1}{2}$, $1 \leq j < i < k$. Wir erhalten

$$\begin{aligned} |\mu_{k,j}| &= \frac{| < \pi_{j,x}(b_k), \widehat{b}_{j,x} > |}{\|\widehat{b}_{j,x}\|^2} \leq \frac{\|b_k\|}{\|\widehat{b}_{j,x}\|} \\ &\leq (\delta - \tfrac{1}{4})^{-n/2}\, n^{3/2} \left(\sum_{i=1}^n p_i^2\right)^{1+\frac{1}{2\,(n-1)}} . \end{aligned}$$

Mit Lemma 2.13 und Korollar 2.14 erhalten wir daher für die Größe der Gram–Schmidt Koeffizienten $\mu_{k,j}$, $1 \leq j < k$, und die Länge des Vektors b_k während der vollständigen Größenreduktion von b_k für $l = 1, \ldots, k-1$:

$$|\mu_{k,j}^{(l)}| \leq 1.5^l\, (\delta - \tfrac{1}{4})^{-n/2}\, n^{3/2} \left(\sum_{i=1}^n p_i^2\right)^{1+\frac{1}{2\,(n-1)}} , \tag{2.24}$$

$$\|b_k^{(l)}\| \leq 1.5^{2\,l+1}\, (\delta - \tfrac{1}{4})^{-n}\, n^3 \left(\sum_{i=1}^n p_i^2\right)^{1.5+\frac{1}{n-1}} . \tag{2.25}$$

Aus den Ungleichungen (2.22), (2.23), (2.24), (2.25) sehen wir, daß die Absolutbeträge der Gram–Schmidt Koeffizienten und der Einträge der Basisvektoren während rat–SIRA durch $(\delta - \frac{1}{4})^{-n^2/2}\, n^{3\,n/2}\, (\sum_{i=1}^n p_i^2)^{n+3}$ beschränkt sind. □

Aus [LLL82], Formel (1.29) und Ungleichung (2.16) folgt, daß die binäre Länge der Nenner und Zähler der Gram–Schmidt Koeffizienten und der Einträge der Basisvektoren während rat–SIRA durch $O(n\,(n + \sum_{i=1}^n \lceil \log |p_i| \rceil) + \lceil \log |q| \rceil)$ beschränkt ist.

Größe der dualen Basisvektoren und Kosten der Matrixinvertierung:

Wir führen die Matrixinvertierung modulo einer genügend großen 2er–Potenz durch (vergleiche [HJLS89], Kapitel 6, Seite 879):

Vor dem letzten Austausch $b_{n-1} \leftrightarrow b_n$ sind die projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_n)$ größenreduziert und somit $|\mu_{i,j}| \leq \frac{1}{2}$, $1 \leq j < i \leq n$. Nach Satz 2.5 (1) gilt daher für die Länge der dualen Basisvektoren a_i, $1 \leq i \leq n$:

$$\|a_i\| \leq 1.5^{n-i} \left(\sum_{i=1}^n p_i^2\right)^{1/2} [0.64\,\sqrt{n}\,(\delta - \tfrac{1}{4})^{-(n/2-1)} + 1] \leq 1.5^{n-i}\,(\delta - \tfrac{1}{4})^{-n/2}\,\sqrt{n} \left(\sum_{i=1}^n p_i^2\right)^{1/2} .$$

Diese Abschätzung ist natürlich auch nach dem Austausch $b_{n-1} \leftrightarrow b_n$ erfüllt. Die Bitlänge der Einträge der dualen Basisvektoren ist daher durch $\lceil \log[1.5^n\,(\delta - \frac{1}{4})^{-n/2}\,\sqrt{n}\,(\sum_{i=1}^n p_i^2)^{1/2}] \rceil$ beschränkt. Somit genügt es, die inverse Matrix $[a_1, \ldots, a_n]^\top := [b_1, \ldots, b_n]^{-1}$ nach der Schulmethode und modulo der Primzahlpotenz 2^e mit $e := 1 + \lceil \log[1.5^n\,(\delta - \frac{1}{4})^{-n/2}\,\sqrt{n}\,(\sum_{i=1}^n p_i^2)^{1/2}] \rceil$ zu berechnen. Für die Matrixinvertierung benötigt rat–SIRA dann n Divisionen modulo 2^e, wobei jede Division mit dem erweiterten euklidischen Algorithmus in $O(e)$ arithmetischen Schritten auf $O(e)$–Bit langen ganzen Zahlen durchgeführt werden kann. Damit kostet die Matrixinvertierung $O(n^3 + n\,\lceil \log[1.5^n\,(\delta - \frac{1}{4})^{-n/2}\,\sqrt{n}\,(\sum_{i=1}^n p_i^2)^{1/2}] \rceil)$ arithmetische Operationen auf ganzen Zahlen der Bitlänge $O(n\,(n + (\sum_{i=1}^n \lceil \log |p_i| \rceil)))$.

Insgesamt erhalten wir folgendes Resultat, welches die polynomiale Bitkomplexität von rat–SIRA beweist:

Satz 2.15 *Auf Eingaben* $x := (p_1, \ldots, p_n)/q \in \mathbb{Q}^n$, $p_i \in \mathbb{Z}$, $i = 1, \ldots, n$, $q \in \mathbb{Z}$, $\delta \in (\frac{1}{2}, 1)$ *und* $\epsilon \in \mathbb{Q}_+$, *führt rat–SIRA höchstens* $O(n^5\,(n + \lceil|\log\epsilon|\rceil))$ *arithmetische Operationen auf ganzen Zahlen der Bitlänge* $O(n\,(n + \sum_{i=1}^{n}\lceil\log|p_i|\rceil) + \lceil\log|q|\rceil)$ *aus.*

2.4 Numerische Stabilität

Zur Beschleunigung der Abarbeitung der Algorithmen SIRA und rat–SIRA ersetzen wir die exakte Arithmetik auf den rationalen Zahlen $\mu_{i,j}$, $\|\widehat{b}_{j,x}\|^2$ durch Gleitpunktarithmetik. Die Vektoren $x, b_1, \ldots, b_n$, $a_1, \ldots, a_n$ der dualen Basen führen wir in exakter ganzzahliger Arithmetik mit. Um Rundungsfehler zu minimieren, verwenden wir statt den Gram–Schmidt Größen $\mu_{i,j}$, $\|\widehat{b}_{j,x}\|^2$ die normalisierten Gram–Schmidt Koeffizienten $\tau_{i,j} := \mu_{i,j}\,\|\widehat{b}_{j,x}\|$. Wir berechnen die $\tau_{i,j}$ mit Hilfe der Givens Rotation. Dies erfordert Quadratwurzelberechnung; somit sind die $\tau_{i,j}$ im allgemeinen nicht rational.

Numerische Fehleranalyse. Wir beschreiben im folgenden das Modell der numerischen Fehleranalyse bei Gleitpunktarithmetik nach Wilkinson [Wi63]. Für jede arithmetische Operation $+$, $-$, $\cdot$, $/$, $<$, $\lceil\ \rfloor$, $\sqrt{\ }$ entstehen Rundungsfehler dadurch, daß das Ergebnis der arithmetischen Operation auf die (im Computer) nächste darstellbare Gleitpunktzahl gerundet wird. Sei hierzu t' der Gleitpunktwert einer reellen Zahl t und $t - t'$ der *absolute,* sowie $(1 - t'/t)$ der *relative (Rundungs–) Fehler.* Sei weiter r die Anzahl der Genauigkeitsbits der Gleitpunktarithmetik und 2^{-r} der *maximale relative Fehler.* r ist durch die jeweilige CPU–Architektur vorgegeben. Im allgemeinen wird eine Gleitpunktzahl als Produkt einer binären Gleitpunktzahl $m := b_0.b_1 \ldots b_i \ldots b_{r-1}$, $b_i \in \{0, 1\}$ und einer 2er Potenz 2^e dargestellt. m wird dabei als *Mantisse* und e als *Exponent* bezeichnet. r ist dann die Anzahl der Binärstellen der Mantisse.

Für unsere Experimente verwenden wird den IEEE 754–Standard für Gleitpunktzahlen mit doppelter Genauigkeit[2]. Jede Gleitpunktzahl wird durch 64 Bits dargestellt, wobei 11 Bits für den Exponenten reserviert sind, 1 Bit für das Vorzeichen und 53 Bits für die Mantisse[3].

Rundungsfehleranalyse bei der Orthonormalisierung. Die $n \times (n+1)$–Matrix $B := (b_{i,j}) = [x, b_1, \ldots, b_n]$ besitzt eine eindeutige Zerlegung $B = U \cdot L^\top$ in eine orthogonale $n \times n$–Matrix U und eine obere Dreiecksmatrix $L^\top$. Dann ist $L = (\tau_{i,j})_{\substack{0 \le i \le n \\ 1 \le j \le n}}$ und im

[2] In der Programmiersprache C ist dies 'double precision format'.

[3] Hierbei wird ein sogenanntes 'Hidden Bit' gespart, indem das Vorkommabit der Mantisse vereinbarungsgemäß gesetzt wird.

Elementare Rotation $G_{i,j}$

EINGABE Matrix $\overline{B} = (\overline{b}_{i,j})_{\substack{0 \le i \le n \\ 1 \le j \le n}}$

$discr := \overline{b}_{j,j}^2 + \overline{b}_{i,j}^2$;
IF $discr \neq 0$ THEN
  $c := \overline{b}_{j,j} / \sqrt{discr}$; $s := \overline{b}_{i,j} / \sqrt{discr}$;
  FOR $k = 1, \ldots, n$ DO   * rotiere $\overline{b}_{i,j}$ zu 0 *
    $t_1 := \overline{b}_{j,k}$; $t_2 := \overline{b}_{i,k}$;
    $\overline{b}_{i,k} := c * t_2 - s * t_1$;
    $\overline{b}_{j,k} := c * t_1 + s * t_2$;

AUSGABE $\overline{B}$

Abbildung 2.1: Elementare Rotation $G_{i,j}$

Falle $\widehat{b}_{n,x} = 0$ ist $U = \left[\frac{x}{\|x\|}, \frac{\widehat{b}_{1,x}}{\|\widehat{b}_{1,x}\|}, \ldots, \frac{\widehat{b}_{n-1,x}}{\|\widehat{b}_{n-1,x}\|} \right]$. (Für uns ist nur der Fall $\widehat{b}_{n,x} = 0$ interessant, da im Falle $\widehat{b}_{n,x} \neq 0$ SIRA und rat–SIRA terminieren.) Aus der Orthogonalität von U folgt $L^\top = U^\top B$. $U^\top = U^{-1}$ ist dabei Produkt von *elementaren Rotationen* (ER) $G_{i,j}$, $1 \le j < i \le n$, die wie folgt definiert sind:
Sei $\overline{B} = (\overline{b}_{i,j}) = [x, \overline{b}_1, \ldots, \overline{b}_n]$. Dann bewirkt die Matrixmultiplikation $\overline{B} \mapsto G_{i,j}\,\overline{B}$, daß durch Transformation der Spaltenvektoren $\overline{b}_j, \overline{b}_i$ der Eintrag $\overline{b}_{i,j}$ zu 0 rotiert wird. Die Matrix $G_{i,j}$ sieht wie folgt aus:

$$G_{i,j} = \begin{array}{c} \\ 1 \\ \\ j \\ \\ i \\ \\ n \end{array} \begin{array}{c} \begin{array}{cccccccc} 1 & & j & & i & & n \end{array} \\ \begin{pmatrix} 1 & \ldots & 0 & \ldots & 0 & \ldots & 0 \\ \vdots & 1 & \vdots & 0 & \vdots & 0 & \vdots \\ 0 & \ldots & c & \ldots & s & \ldots & 0 \\ \vdots & 0 & \vdots & 1 & \vdots & 0 & \vdots \\ 0 & \ldots & -s & \ldots & c & \ldots & 0 \\ \vdots & 0 & \vdots & 0 & \vdots & 1 & \vdots \\ 0 & \ldots & 0 & \ldots & 0 & \ldots & 1 \end{pmatrix} \end{array} \qquad \text{mit} \quad s := \frac{\overline{b}_{i,j}}{\sqrt{\overline{b}_{j,j}^2 + \overline{b}_{i,j}^2}} \quad \text{und} \quad c := \frac{\overline{b}_{j,j}}{\sqrt{\overline{b}_{j,j}^2 + \overline{b}_{i,j}^2}} \quad .$$

Offensichtlich ist $G_{i,j}$ orthogonal. Die Transformation $\overline{B} \mapsto G_{i,j}\,\overline{B}$ ist in Abbildung 2.1 als Routine beschrieben (vergleiche [GoL89]).

Sei $|\overline{B}| := \max_{1 \le i \le n} \|\overline{b}_i\|$. Wegen der Orthogonalität von $G_{i,j}$ ist $|\overline{B}| = |G_{i,j}\,\overline{B}|$. Insbesondere ist die Länge der Vektoren $\overline{b}_i$ gegenüber elementaren Rotationen $G_{i,j}$ invariant. Nach der Fehleranalyse gemäß Wilkinson ([Wi63], Seiten 131–139, vergleiche auch [H95], Seiten 58–60) gilt für den Gesamtfehler bei der Anwendung einer elementaren Transformation $G_{i,j}$ folgendes

Lemma 2.16 *[Wi63]*
Seien $\overline{b}_0, \ldots, \overline{b}_n$ *die Spaltenvektoren der Matrix* $\overline{B}$, $|\overline{B}| := \max_{0 \le i \le n} \|\overline{b}_i\|$ *und* $G_{i,j}$ *die oben definierte orthogonale Matrix, welche den Eintrag* $\overline{b}_{i,j}$ *von* $\overline{B}$ *zu 0 rotiert. Dann gilt für den Fehler der transformierten Spaltenvektoren* $G_{i,j}\,\overline{b}_k$, $k = 0, \ldots, n$:

$$\|(G_{i,j}\,\overline{b}_k) - (G_{i,j}\,\overline{b}_k)'\| \;\le\; 7 \cdot 2^{-r}\,|\overline{B}| \,. \qquad \Box$$

Die Fehlerabschätzung für die transformierten Spaltenvektoren der Matrix $\overline{B}$ nach Lemma 2.16 gilt auch für den Fall, daß eine Folge von elementaren Rotationen $G_{i,j}$ mit disjunkten Indextupeln (i,j) auf die Matrix $\overline{B}$ angewendet werden. Wir nennen die elementaren Rotationen einer solchen Sequenz im folgenden *disjunkt* .

Für die Orthonormalisierung der Matrix $B := [x, b_1, \ldots, b_n]$ wird eine Folge von elementaren Rotationen $G_{i,j}$ mit $1 \le j < i \le n$ auf B angewendet. Insgesamt sind dies $\sum_{k=1}^{n-1}(n-k) = (n-2)\,(n-1)/2$ viele. Einer Betrachtung von Gentleman zufolge [Ge75] (vergleiche auch die exakte Analyse in [H95], Seiten 60–61) können die für die Orthonormalisierung erforderlichen $(n-2)\,(n-1)/2$ elementaren Rotationen derart auf $2\,n-3$ 'Stufen' verteilt werden, daß jede Stufe nur aus disjunkten elementaren Rotationen besteht. Die Routine *Givens Rotation* von Abbildung 2.2 führt die vollständige Orthonormalisierung von $B := [x, b_1, \ldots, b_n]$ aus; die mittlere Schleife der Routine stellt dabei jeweils eine Stufe disjunkter elementarer Rotationen dar. Man beachte, daß bei elementarer Rotation des Elementes $\overline{b}_{i-j,j}$ zu 0 die Elemente $\overline{b}_{i-j,k}$, $1 \le k < i-j$ schon zu 0 rotiert worden sind.

Gibt $(i_1, j_1) \to (i_2, j_2)$ die lexikographische Ordnung der Indextupel (i,j) an, in der die äußere Doppelschleife der Routine *Givens Rotation* abgearbeitet wird, so läßt sich die Reihenfolge der Abarbeitung der Indextupel (i,j) durch folgende Pfeilbewegung in der

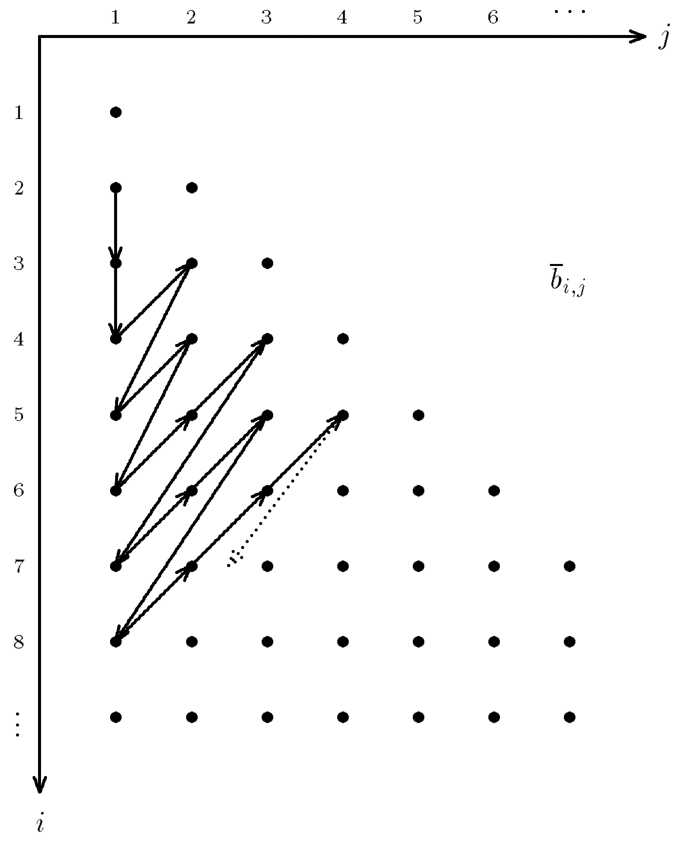
Givens Rotation
EINGABE Matrix $B = (b_{i,j})_{\substack{1 \le i \le n \\ 0 \le j \le n}} = [x, b_1, \ldots, b_n]$

Setze $\overline{B} := B$;
FOR $i = 2, \ldots, 2\,n-2$ DO * führe disjunkte elementare Rotationen aus *
  FOR $j = \max\{i-n, 1\}, \ldots, \lceil \frac{i-1}{2} \rceil$ DO * führe elementare Rotation $G_{i-j,j}$ aus *
    $\overline{B} := G_{i-j,j}\,\overline{B}$;

AUSGABE $\overline{B}$

Abbildung 2.2: Routine zur Givens Rotation von $B = [x, b_1, \ldots, b_n]$

durch die Schleifenindizes i, j definierten Matrix beschreiben:



Man sieht unmittelbar, daß die Pfeilbewegung auf einer Diagonalen von links unten nach rechts oben der Abarbeitung disjunkter Indextupel (i, j) für elementare Rotationen $G_{i,j}$ entspricht. Die Fehleranalyse von Gentleman [Ge75, H95] zeigt insbesondere, daß nach s Stufen disjunkter elementarer Rotationen der Gleitpunktfehler der Elemente der transformierten Matrix $[\overline{b}_0, \overline{b}_1, \ldots, \overline{b}_n]$ von der Eingabematrix $B := [b_0 := x, b_1, \ldots, b_n]$ wie folgt abhängt:

$$\|\overline{b}_i' - \overline{b}_i\| \quad \leq \quad 7 \cdot 2^{-r}\, s\, (1 + 7 \cdot 2^{-r})^{s-1}\, \|b_i\|\,,\ i = 0, \ldots, n\ .$$

Da $d\,2^{-r} \ll 1$ für Konstanten $d > 0$, kann man nach dem Modell der Fehleranalyse von Wilkinson jeden Term $(1 + d\,2^{-r})^s$, $s \in \mathbb{N}$ durch $(1 + s\,d\,2^{-r})$ abschätzen. Wir erhalten somit unter Vernachlässigung quadratischer Terme in 2^{-r} für den Fehler der Einträge der Ausgabematrix $(\tau_{j,i})_{\substack{1 \leq j \leq n \\ 0 \leq i \leq n}} = [\overline{b}_0, \ldots, \overline{b}_n]$ der gesamten Routine Givens Rotation:

$$\begin{aligned} |\tau_{j,i}' - \tau_{j,i}| \quad &\leq \quad \|\overline{b}_j' - \overline{b}_j\|\ 7\,(2\,n - 3)\,2^{-r}\,(1 + (2\,n - 4)\,2^{-r})\,|B| \\ &\leq \quad (14\,n\,2^{-r} - 21 \cdot 2^{-r})\,(1 + 2\,n\,2^{-r} - 4)\,|B| \ \leq \ 14\,n\,2^{-r}\,|B|\ . \end{aligned}$$

Wir haben daher folgendes Lemma gezeigt.

Lemma 2.17 *Sei* $B = [b_0 := x, b_1, \ldots, b_n]$ *die Eingabematrix der Givens Rotation mit* $|B| := \max_{0 \leq i \leq n} \|b_i\|$ *und* $[\overline{b}_0, \ldots, \overline{b}_n] := (\tau_{j,i})_{\substack{1 \leq j \leq n \\ 0 \leq i \leq n}} := U^\top B$ *die Ausgabematrix. Dann* *gilt unter Vernachlässigung von Termen 2^{-2r} für den Fehler der Einträge $\tau_{j,i}$, $0 \le j < i \le n$:*

$$|\tau'_{j,i} - \tau_{j,i}| \;\le\; \|\overline{b}'_j - \overline{b}_j\| \;\le\; 14\, n\, 2^{-r}\, |B|\;. \qquad \square$$

Bemerkungen. *1. Das Verfahren der Givens Rotation wurde schon in den parallelen L^3–Algorithmen von Heckler, Thiele [HT93] und Joux [Jo93] verwendet. Hier führt Givens Rotation zu einer effizienten parallelen Version des L^3–Algorithmus.*

Ferguson und Bailey [FB92] benutzen eine Variante der Givens Rotation, die Householder Reflektion, *in Verbindung mit dem HJLS–Algorithmus*[4]*. Während Householder Reflektionen jeweils nur Komponenten eines Spaltenvektors einer Matrix annihilieren, kann dies bei Givens Rotationen selektiv für beliebige Matrixeinträge geschehen. Ferguson und Bailey benutzen das Verfahren der Householder Reflektion, um beim HJLS–Algorithmus nach Austauschen $b_{k-1} \leftrightarrow b_k$ die normalisierten Gram–Schmidt Koeffizienten $\tau_{i,j}$, $i,j \in \{k-1,k\}$ durch Annihilierung von $\tau_{k-1,k}$ zu aktualisieren.*

2. Während die Berechnung der normalisierten Gram–Schmidt Koeffizienten $\tau_{i,j}$ die Quadratwurzelfunktion verwendet, kann die Bestimmung der Gram–Schmidt Größen $\mu_{i,j}$, $\|\widehat{b}_{i,x}\|^2$ mit Hilfe der Givens Rotation auch in exakter Arithmetik auf rationalen Zahlen durchgeführt werden. Dies geschieht durch Anwendung quadratwurzelfreier Varianten der Givens Rotation [Ge73, GoL89, GS91]. Die diesen Verfahren zugrundeliegende Idee ist, daß bei der Berechnung der Gram–Schmidt Größen $\mu_{i,j} = \tau_{i,j}/\tau_{j,j}$, $\|\widehat{b}_{i,x}\|^2 = \tau_{i,i}^2$ die Quadratwurzeln im Nenner der $\tau_{i,j}$ sich herauskürzen bzw. quadrieren und somit jeweils herausfallen. Allerdings können die Zähler und Nenner der Gram–Schmidt Größen bei Verwendung exakter rationaler Arithmetik wie bei GSO (vergleiche Abschnitt 2.3) außerordentlich groß werden.

Vermeidung fehlerhafter Austausche bei SIRA und rat–SIRA. Wir geben hinreichende Begingungen für die Vermeidung fehlerhafter Austausche in SIRA und rat–SIRA an. Ein Austausch $b_{k-1} \leftrightarrow b_k$ ist dann fehlerhaft, wenn die Länge $\tau_{k-1,k-1}$ der Höhe $\widehat{b}_{k-1,x}$ nicht erniedrigt wird.

Wir nennen einen Austausch $b_{k-1} \leftrightarrow b_k$ *gut,* falls $\tau_{k-1,k-1}$ erniedrigt wird. Vor einem guten Austausch gilt $\tau_{k-1,k-1} > (\tau_{k,k}^2 + \tau_{k,k-1}^2)^{1/2}$.

Nach jedem Austausch $b_{k-1} \leftrightarrow b_k$ der L^3–Reduktion mit δ in Schritt 2 von SIRA und rat–SIRA werden die Gram–Schmidt Größen mittels Givens Rotation von $[x, b_1, \ldots, b_n]$ neu berechnet. Hierbei genügen die Gleitpunktwerte der normalisierten Gram–Schmidt Koeffizienten vor Abtesten der L^3–Bedingung der Fehlerabschätzung von Lemma 2.17.

[4] Householder Reflektionen bilden Vektoren $x \in \mathbb{R}^n$ in den Orthogonalraum eines Vektors $v \in \mathbb{R}^n - \{0\}$ ab. Man sagt in diesem Falle, der Vektor x wird in die Hyperebene *$span(v)^\perp$ reflektiert* [GoL89], Seiten 195–201.

Satz 2.18 *Sei* $(\tau_{i,j})_{\substack{0\le i\le n\\ 1\le j\le n}}$ *die Transponierte der Matrix* $L^\top = U^\top B$, *die durch Anwendung der Givens Rotation* $U^\top$ *auf die Matrix* $B := [b_0 := x, b_1, \ldots, b_n]$ *entsteht, und* $|B| := \max_{0\le i\le n} \|b_i\|$. *Falls* $\delta\, \tau'^2_{k-1,k-1} > \tau'^2_{k,k} + \tau'^2_{k,k-1}$ *für die gerundeten Gleitpunktwerte* $\tau_{i,j}$ *gilt und* $14\, n\, 2^{-r}\, |B|\; (1+\sqrt{\delta})^2/(1-\delta) \le \tau_{k-1,k-1}$, *dann ist unter Vernachlässigung von Termen* 2^{-2r} *der Austausch* $b_{k-1}\leftrightarrow b_k$ *gut.*

Beweis. Nach Lemma 2.17 erfüllen die gerundeten Gleitpunktwerte der Matrix $[\overline{b}_1, \ldots, \overline{b}_n] := (\tau_{i,j})^\top$ die Fehlerabschätzung $\|\overline{b}_{k-1} - \overline{b}'_{k-1}\|$, $\|\overline{b}_k - \overline{b}'_k\| \le 7\,(2\,n-3)\, 2^{-r}\, |B|$. Somit ist

$$\tau_{k-1,k-1} - (\tau^2_{k,k} + \tau^2_{k,k-1})^{1/2} \;\ge$$

$$(1-\sqrt{\delta})\, \tau_{k-1,k-1} + \sqrt{\delta}\, \tau'_{k-1,k-1} - (\tau'^2_{k,k-1} + \tau'^2_{k,k})^{1/2} - (\sqrt{\delta}+1)\, 7\,(2\,n-3)\, 2^{-r}\, |B| \,.$$

Nach Annahme war $\sqrt{\delta}\, \tau'_{k-1,k-1} - \sqrt{\tau'^2_{k,k-1} + \tau'^2_{k,k}} + 2^{-r+1} > 0$, wobei 2^{-r+1} den Fehler bei der Berechnung der Austauschbedingung für die gerundeten Gleitpunktwerte darstellt. Wir erhalten somit $\tau_{k-1,k-1} > (\tau^2_{k,k} + \tau^2_{k,k-1})^{1/2}$, da $14\,(1+\sqrt{\delta})/(1-\sqrt{\delta}) = 14\,(1+\sqrt{\delta})^2/(1-\delta)$. □

Umgekehrt zeigt folgender Satz, daß Austausche $b_{k-1}\leftrightarrow b_k$ in SIRA durchgeführt werden, falls für die exakten τ–Werte $\overline{\delta}\, \tau_{k-1,k-1} > (\tau^2_{k,k} + \tau^2_{k,k-1})^{1/2}$ mit einem $\overline{\delta} < \delta - 14\, n\, 2^{-r} \max_{0\le i\le n} \|b_i\|\, (1+\sqrt{\delta})^2/\tau_{k-1,k-1}$ gilt.

(Man betrachte zum Beispiel die Werte $\overline{\delta} = \frac{1}{2}$ und $\delta = \frac{3}{4}$. Dann implizieren die Ungleichungen $\frac{1}{2}\, \tau_{k-1,k-1} > (\tau^2_{k,k} + \tau^2_{k,k-1})^{1/2}$ und $165\, n\, 2^{-r} \max_{0\le i\le n} \|b_i\| < \tau_{k-1,k-1}$, daß $\frac{3}{4}\, \tau'_{k-1,k-1} > (\tau'^2_{k,k} + \tau'^2_{k,k-1})^{1/2}$ für die Gleitpunktwerte der normalisierten Gram–Schmidt Koeffizienten gilt und somit ein Austausch $b_{k-1}\leftrightarrow b_k$ in SIRA ausgeführt wird.)

Satz 2.19 *Sei* $(\tau_{i,j})_{\substack{0\le i\le n\\ 1\le j\le n}}$ *die Transponierte der Matrix* $L^\top = U^\top B$, *die durch Anwendung der Givens Rotation* $U^\top$ *auf die Matrix* $B := [b_0 := x, b_1, \ldots, b_n]$ *entsteht, und* $|B| := \max_{0\le i\le n} \|b_i\|$. *Dann implizieren unter Vernachlässigung von Termen* 2^{-2r} *die Ungleichungen* $\overline{\delta}\, \tau^2_{k-1,k-1} > \tau^2_{k,k} + \tau^2_{k,k-1}$ *und* $\overline{\delta} < \delta - 14\, n\, 2^{-r}\, |B|\, (1+\sqrt{\delta})^2/\tau_{k-1,k-1}$ *die Ungleichung* $\delta\, \tau'^2_{k-1,k-1} > \tau'^2_{k,k} + \tau'^2_{k,k-1}$.

Beweis. Wie im Beweis von Satz 2.18 gilt mit Lemma 2.17

$$\sqrt{\delta}\, \tau'_{k-1,k-1} - (\tau'^2_{k,k} + \tau'^2_{k,k-1})^{1/2} \;\ge$$

$$(\sqrt{\delta} - \sqrt{\overline{\delta}})\, \tau_{k-1,k-1} + \sqrt{\overline{\delta}}\, \tau_{k-1,k-1} - (\tau^2_{k,k-1} + \tau^2_{k,k})^{1/2} - (1+\sqrt{\delta})\, 14\, n\, 2^{-r}\, |B| \;,$$

wobei nach Voraussetzung $\sqrt{\overline{\delta}}\, \tau_{k-1,k-1} - \sqrt{\tau^2_{k,k-1} + \tau^2_{k,k}} + 2^{-r+1} > 0$. Die Behauptung folgt mit

$$\frac{14\, n\, 2^{-r}\, |B|\, (1+\sqrt{\delta})}{(\sqrt{\delta} - \sqrt{\overline{\delta}})} \;=\; 14\, n\, 2^{-r}\, |B|\; \frac{(1+\sqrt{\delta})^2}{(\delta - \overline{\delta})} \;<\; \tau_{k-1,k-1} \,. \qquad \square$$

Wir geben im folgenden Beispiele an, für die Austausche $b_{k-1} \leftrightarrow b_k$ von rat–SIRA bei rationalen Eingaben $x := (p_1, \ldots, p_n)/q \in \mathbf{Q}^n$ gut sind. Vor jedem Austausch $b_{k-1} \leftrightarrow b_k$ sind die Vektoren $b_1, \ldots, b_n$ größenreduziert. Sei $|B| := \max_{0 \le i \le n} \|b_i\|$. Dann ist nach Ungleichung (2.18)

$$|B| \;\le\; 2\,(\delta - \tfrac{1}{4})^{-(n/2-1)}\, \epsilon^{-1}\, n^{3/2} \left(\sum_{i=1}^{n} p_i^2 \right)^{\frac{1}{2}} .$$

Mit Satz 2.18 ist daher ein Austausch $b_{k-1} \leftrightarrow b_k$ gut, falls $\tau_{k-1,k-1} > \epsilon$ und

$$28 \left(\frac{1}{\sqrt{\delta - \frac{1}{4}}} \right)^{n-2} \frac{1 + \sqrt{\delta}}{1 - \sqrt{\delta}}\, \epsilon^{-2}\, n^{5/2} \max_{1 \le i \le n} |p_i|\, 2^{-r} \;<\; 1 \,. \tag{2.26}$$

Bei fester Eingabe $x = (p_1, \ldots, p_n)/q \in \mathbf{Q}^n$, $\epsilon \in \mathbf{Q}_+$ ist die linke Seite dieser Ungleichung bei Eingabewerten $\delta \in (\frac{1}{2}, 1)$ minimal für

$$\sqrt{\delta} \;=\; \frac{2}{3\,(2-n)} - \frac{8\, 2^{1/3}}{3\, C_n^{1/3}\, (2-n)} - \frac{C_n^{1/3}}{6\, 2^{1/3}\, (2-n)} \,,$$

mit $\quad C_n := -128 + 4\,[-1024 + (32 + 27\,(3 - 2\,n)\,(2-n)^2)^2]^{1/2} - 108\,(3 - 2\,n)\,(2-n)^2$, das heißt approximativ

$$\sqrt{\delta} \;\approx\; \left(\frac{n/2 - 3/4}{n/2 - 1} \right)^{1/3} - \frac{1}{3\,(n/2-1)} \;\approx\; 1 - \frac{2}{3\,n} \,,$$

also für genügend große n :

$$\delta \;\approx\; 1 - \frac{4}{3\,n} \,.$$

Bei $\epsilon^{-1} = 16 = 2^4$ und $\max_{1 \le i \le n} |p_i| \le 2^{20}$ gilt Ungleichung (2.26) für $\delta = \frac{3}{4}$ bis Dimension 15 , für $\delta = 0.95$ bis Dimension 25 .

Da die Ungleichung $\tau_{k-1,k-1} > \epsilon$ mit Ausnahme einiger weniger Austausche gilt, ist rat–SIRA für Eingaben $\epsilon^{-1} = 16 = 2^4$, $\max_{1 \le i \le n} |p_i| \le 2^{20}$, $\delta = 0.95$ stabil bis Dimension 25 .

Die praktischen Ergebnisse zeigen für $\delta = 0.95$ sogar eine weitaus bessere Stabilität.

Vermeidung fehlerhafter Größenreduktionsschritte bei SIRA und rat–SIRA. Wir geben hinreichende Bedingungen für die Vermeidung fehlerhafter Größenreduktionsschritte in SIRA und rat–SIRA an. Fehlerhaft sind Größenreduktionsschritte $b_k := b_k - \lceil \mu_{k,j} \rfloor\, b_j$, die den Absolutbetrag des Gram–Schmidt Koeffizienten $\mu_{k,j} = \; < \pi_{j,x}(b_k), \pi_{j,x}(b_j) > / \|\pi_{j,x}(b_j)\|^2$ nicht genügend verkleinern. (Dadurch werden die Vektoren $\pi_{j,x}(b_j), \pi_{j,x}(b_k)$ nicht ausreichend orthogonal zueinander.)

Wir nennen einen Größenreduktionsschritt $b_k := b_k - \lceil \mu_{k,j} \rfloor\, b_j$ *gut* , falls dadurch $|\mu_{k,j}| \le \frac{3}{4}$ wird.

In der L^3–Reduktion mit δ und in der Größenreduktion von SIRA und rat–SIRA wird nach jeder Größenreduktion von b_k die Orthonormalisierung $(\tau_{i,j})_{\substack{0 \le i \le n \\ 1 \le j \le n}}$ von $[x, b_1, \ldots, b_n]$

mittels Givens Rotation neu berechnet. Die Gleitpunktwerte der normalisierten Gram–Schmidt Koeffizienten $\tau_{i,j}$ erfüllen die Fehlerabschätzung von Lemma 2.17. Wir ermitteln nun den Fehler der nach der Größenreduktion mit $\lceil \mu_{k,j} \rfloor$ neu berechneten Gram–Schmidt Koeffizienten $\mu_{k,i}$, $0 \leq i \leq j$.

Lemma 2.20 *Sei $(\tau_{i,j})_{\substack{0 \leq i \leq n \\ 1 \leq j \leq n}}$ die Transponierte der Matrix, die durch Anwendung der Givens Rotation auf die Matrix $B := [b_0 := x, b_1, \ldots, b_n]$ entsteht, und $|B| := \max_{0 \leq i \leq n} \|b_i\|$. Falls $120\,(\delta - \frac{1}{4})^{-(2k+1)\,n/2}\, n^{3\,(k+1)/2}\, \left(\sum_{i=1}^{n} p_i^2\right)^{2k+\frac{1}{2}+\frac{2k+1}{2\,(n-1)}}\, 2^{-r} < 1$, dann ist unter Vernachlässigung von Termen 2^{-2r} der Größenreduktionsschritt $b_k := b_k - \lceil \mu_{k,j} \rfloor\, b_j$ gut.*

Beweis. Nach Lemma 2.17 gilt für den Fehler bei der Berechnung der Gram–Schmidt Koeffizienten $\mu_{i,j}$, $0 \leq i, j \leq n$:

$$\mu'_{i,j} \;=\; \frac{\tau'_{i,j}}{\tau'_{j,j}}\,(1+\kappa) \;=\; \frac{\tau_{i,j}\,(1+\epsilon_1)}{\tau_{j,j}\,(1+\epsilon_2)}\,(1+\kappa)$$

mit $|\epsilon_i| \leq 14\,n\,2^{-r}\,|B|$, $i = 1, 2$ und $|\kappa| \leq 2^{-r}$, also für $n\,2^{-r} \ll 1$:

$$|\mu'_{i,j} - \mu_{i,j}| \;\leq\; |\mu_{i,j}|\,(1 + 28\,n\,2^{-r}\,|B|) + 2^{-r} \;\leq\; 28.5\,n\,2^{-r}\,|B|\,,\; 0 \leq i, j \leq n\;. \tag{2.27}$$

Der Fehler bei der Berechnung des Reduktionskoeffizienten $\lceil \mu_{k,j} \rfloor$ resultiert aus dem Fehler bei der Operation $\lceil\,.\,\rfloor$ und dem numerischen Fehler des Operanden $\mu_{k,j}$.

Für ein falsches Ergebnis $\lceil a \rfloor'$ muß der Operand a aber von der Form $a = m + \frac{1}{2} + \kappa$ mit $m \in \mathbb{Z}$ und $|\kappa| \leq 2^{-r}$ sein. Dann gilt

$$|\lceil a \rfloor' - a| \;=\; |\lceil a \rfloor - a| + \kappa$$

für ein $|\kappa| \leq 2^{-r}$. Wir erhalten somit für den Fehler des Absolutbetrages des aktualisierten Gram–Schmidt Koeffizienten $\mu_{k,j}^{(neu)} := \mu_{k,j} - \lceil \mu_{k,j} \rfloor$:

$$\begin{aligned} \big|\,|\mu_{k,j}^{\prime\,(neu)}| - |\mu_{k,j}^{(neu)}|\,\big| \;&=\; \big|\,|\mu'_{k,j} - \lceil \mu'_{k,j} \rfloor'| - |\mu_{k,j} - \lceil \mu_{k,j} \rfloor|\,\big| \\ &\leq\; \big|\,|\mu'_{k,j} - \lceil \mu'_{k,j} \rfloor| - |\mu_{k,j} - \lceil \mu_{k,j} \rfloor|\,\big| + 2^{-r} \\ &\leq\; |\mu'_{k,j} - \mu_{k,j}| + 2^{-r} \;\leq\; 29\,n\,u\,|B|\,|\mu_{k,j}|\,,\; 0 \leq j < k \leq n\;. \end{aligned}$$

Nach Ungleichungen (2.22), (2.23) folgt die Behauptung aus $29\,n\,2^{-r}\,|B|\,|\mu_{k,j}| + 2^{-r} < \frac{1}{4}$ und $|\mu_{k,j}^{\prime\,(neu)}| \leq \frac{1}{2} + 2^{-r}$. □

Damit SIRA gute Größenreduktionsschritte ausführt, müssen insbesondere nach Lemma 2.20 vor jedem Größenreduktionsschritt $b_k := b_k - \lceil \mu_{k,j} \rfloor\, b_j$ die Gram–Schmidt Koeffizienten $\mu_{i,j} = \tau_{i,j} / \tau_{j,j}$ aus der Orthonormalisierung $(\tau_{i,j})_{\substack{0 \leq i \leq n \\ 1 \leq j \leq n}}$ der Matrix $[x, b_1, \ldots, b_n]$ berechnet werden. Dies erfordert für jede Größenreduktion von b_k maximal $k-1$ Givens Rotationen.

Falls die normalisierten Gram–Schmidt Koeffizienten $\tau_{i,j}$ mittels Givens Rotation von $[x, b_1, \ldots, b_n]$ nur vor der Größenreduktion von b_k berechnet werden und die Gram–Schmidt Koeffizienten $\mu_{k,i}$, $i = 0, \ldots, k-1$, gemäß der Größenreduktions–Routine aus

Abbildung 1.1 aktualisiert werden, kann sich der numerische Fehler in der Neuberechnung der Gram–Schmidt Koeffizienten stark aufschaukeln.

Dies liegt daran, daß für einen Größenreduktionsschritt $b_k := b_k - \lceil \mu_{k,j} \rfloor b_j$ der Fehler bei der Berechnung von $\lceil \mu_{k,j} \rfloor$ sich aufspaltet in den Fehler bei der Operation $\lceil . \rfloor$ und in den Fehler des Operanden $\mu_{k,j}$. Der Fehler bei der Operation $\lceil . \rfloor$ ist im Extremfall 1 . Damit läßt sich der Fehler bei der Berechnung von $\lceil \mu_{k,j} \rfloor$ für $0 \leq j < k \leq n$ abschätzen durch

$$|\lceil \mu'_{k,j} \rfloor' - \lceil \mu_{k,j} \rfloor| \quad \leq \quad 1 + |\mu'_{k,j} - \mu_{k,j}| + 2^{-r} \quad \leq \quad 1 + 28.5\, n\, 2^{-r}\, |B|\, |\mu_{k,j}| + 2^{-r} \; .$$

Sei $(\tau_{i,j})_{\substack{0 \leq i \leq n \\ 1 \leq j \leq n}}$ die Transponierte der Matrix, die durch Anwendung der Givens Rotation auf die Matrix $B := [b_0 := x, b_1, \ldots, b_n]$ vor der vollständigen Größenreduktion von b_k entsteht und $\mu_{i,j}$, $0 \leq i, j \leq n$, die aus den normalisierten Gram–Schmidt Koeffizienten $\tau_{i,j}$ berechneten Gram–Schmidt Koeffizienten. Vor der Größenreduktion von b_k sei außerdem $|B| := \max_{0 \leq i \leq n} \|b_i\|$, $M := \max_{1 \leq j < i < k} |\mu_{i,j}|$ und $M_k := \max_{1 \leq j < k} |\mu_{k,j}|$. Wie in Lemma 2.13 seien $\mu^{(l)}_{k,i}$ die Gram–Schmidt Koeffizienten nach Ausführung von $b_k - \lceil \mu_{k,j} \rfloor b_j$ für $j = k-1, \ldots, k-l$. Wir vernachlässigen Terme 2^{-2r} . Dann gilt mit Lemma 2.13 nach Induktion über $l = 1, \ldots, k-1$

$$\begin{aligned} |\mu'^{(l)}_{k,i} - \mu^{(l)}_{k,i}| \quad \leq \quad & 28.5\, n\, 2^{-r}\, |B| \left[M_k\, (M+1) + (l+1)\, (M_k + \tfrac{1}{2})\, M + l\, M \right] \\ & + M\, (M+1)^{l+1}\, (1 + 2^{-r}) \quad , \; i = k-l-1, \ldots, 1 \; . \end{aligned} \tag{2.28}$$

Die von 2^{-r} unabhängige Größe $M\,(M+1)^{l+1}$ kommt durch die fehlerhafte Berechnung von $\lceil . \rfloor$ zustande.

Während der Größenreduktion von $b_1, \ldots, b_n$ in Schritt 3 von SIRA und rat–SIRA läßt sich $M\,(M+1)^{l+1}$ durch $\frac{1}{2} \cdot 1.5^{l+1}$ abschätzen, im Falle der L^3–Reduktion von rat–SIRA durch $(\delta - \frac{1}{4})^{-(l+2)\,n/2}\, n^{3\,(l+2)/2}\, \left(\sum_{i=1}^{n} p_i^2\right)^{l+2+\frac{2\,(l+2)}{2\,(n-1)}}$ (vergleiche Ungleichung (2.21)).

Im nächsten Abschnitt werden wir bei der Auswertung der Experimente sehen, daß es für die numerische Stabilität in Dimensionen $n \geq 10$ nicht mehr ausreicht, die Orthonormalisierung $(\tau_{i,j})_{\substack{0 \leq i \leq n \\ 1 \leq j \leq n}}$ von $[x, b_1, \ldots, b_n]$ jeweils nur einmal vor der Größenreduktion von b_k zu berechnen.

2.5 Implementierung und experimentelle Resultate

Die Algorithmen SIRA und rat–SIRA sind in der Programmiersprache C und unter dem Betriebssystem HP–UX, ein UNIX–Derivat der Firma HP (Hewlett Packard), auf einer HP Workstation ‘Apollo 9000’ der Serie 715 implementiert. Der Prozessor der Workstation ist mit 50 MHz und 62 MIPS getaktet.

Wir untersuchen die Performanz der Algorithmen SIRA und rat–SIRA auf rationalen Eingaben $x = (p_1, \ldots, p_n)/q \in \mathbb{Q}^n$, $\epsilon \in \mathbb{Q}_+$ und $\delta = 0.95$.

Wir halten die Vektoren $x, b_1, \ldots, b_n$, $a_1, \ldots, a_n$ in exakter ganzzahliger Arithmetik. Hierzu verwenden wir eine in der Arbeitsgruppe Mathematische Informatik der Univer-

n	ϵ^{-1}	Q	$\oslash \|a_n\|$	$\oslash \max \lambda(x)$	$\oslash \max_i \|b_i\|$	$\oslash \|x' - x\|$	$\oslash$ Sek.	$\sharp\ x'$
			SIRA					
5	8	45	8.02	5376.70	7643.90	5.465e+8	0.04	10
5	16	45	16.63	5376.70	4.799e+5	1.551e+7	0.06	10
5	32	45	24.50	5376.70	2.674e+6	3.321e+5	0.07	10
5	64	45	105.46	5376.70	4.885e+8	1.478e+4	0.07	10
5	128	45	229.48	5376.70	3.042e+9	654.84	0.08	10
5	256	45	322.09	5376.70	1.843e+10	5.72	0.09	10
5	512	45	550.89	5376.70	2.224e+11	0.21	0.09	10
5	1024	45	981.99	5376.70	2.739e+13	0.02	0.10	10
5	2048	45	1672.04	5376.70	1.931e+13	0.02	0.15	9
8	4	45	23.39	229.30	1.189e+9	2.188e+7	0.25	10
8	8	45	46.58	229.30	8.200e+13	2722.61	0.27	10
8	16	45	61.55	229.30	1.067e+14	2.73	0.37	10
8	32	45	78.05	229.30	8.587e+18	1.34	0.45	8
			rat–SIRA					
5	8	45	8.52	5376.70	7635.83	2.631e+8	0.07	10
5	16	45	15.91	5376.70	5.726e+5	3.987e+6	0.06	10
5	32	45	41.40	5376.70	9.765e+6	2.274e+5	0.08	10
5	64	45	94.12	5376.70	1.192e+8	2841.39	0.10	10
5	128	45	115.88	5376.70	5.986e+12	194.04	0.08	10
5	256	45	200.17	5376.70	6.808e+12	5.21	0.09	10
5	512	45	460.43	5376.70	2.404e+14	0.12	0.10	10
5	1024	45	950.49	5376.70	6.245e+16	0.01	0.12	10
5	2048	45	1976.83	5376.70	1.931e+17	0.001	0.12	9
8	4	45	4.126	229.30	4.263e+4	8.033e+8	0.20	10
8	8	45	9.781	229.30	4.030e+6	4.730e+5	0.33	10
8	16	45	16.16	229.30	3.640e+12	9810.24	0.37	10
8	32	45	30.59	229.30	3.030e+15	48.477	0.42	10

Tabelle 2.1: Ergebnisse von SIRA und rat–SIRA für Dimensionen 5 und 8 und $\delta = 0.95$

sität Frankfurt entwickelte Programmbibliothek LARIFARI, mit deren Hilfe man in C–Programmen mit großen ganzen und rationalen Zahlen bis zu einer Bitlänge von maximal 825 rechnen kann. Es zeigt sich bei Computer–Experimenten, daß die dualen Basisvektoren $a_1, \ldots, a_n$ um den Faktor n kleiner als die primären Basisvektoren $b_1, \ldots, b_n$ sind.

Wir berechnen die Orthonormalisierung von $[x, b_1, \ldots, b_n]$ nach jedem Austausch $b_{k-1} \leftrightarrow b_k$ und nach der Größenreduktion eines Vektors b_k jeweils mit der Givens Rotation neu.

In der Tabelle 2.1 sind die experimentellen Resultate für SIRA und rat–SIRA mit $n = 5, 8$ und $\delta = 0.95$ zusammengetragen. Jede Zeile mit Einträgen n, ϵ^{-1} der Tabelle entspricht den Ergebnissen von 10 Eingaben $x := (p_1, \ldots, p_n)/q$, wobei die p_i, $i = 1, \ldots, n$ und q mit dem x^2 *mod* N–Generator von Blum, Blum und Schub generierte [BBS92] pseudozufällige ganze Zahlen aus $[-2^Q + 1, 2^Q - 1]$ sind.

Wir unterscheiden die zwei möglichen Fälle der Terminierung:

- ein Austausch $b_{n-1} \leftrightarrow b_n$ führt zu einer nicht verschwindenden Höhe $\widehat{b}_{n,x} = \widehat{b}_n$ und somit zu einer Relation $a_n = \widehat{b}_n / \|\widehat{b}_n\|^2$ für x ;
- es ist $\max_{i=1,\ldots,n} \|\widehat{b}_{i,x}\| \leq \epsilon$ und SIRA bzw. rat–SIRA stoppt mit '$\lambda(x) \geq 1/\epsilon$' und gibt einen Nahebeipunkt $x' \neq x$ und eine Relation a_n für x' aus.

n	ϵ^{-1}	Q	$\oslash \|a_n\|$	$\oslash \max \lambda(x)$	$\oslash \max_i \|b_i\|$	$\oslash \|x' - x\|$	$\oslash$ Sek.	$\sharp\ x'$
5	4	45	5.20	5376.70	873.62	3.766e+9	0.04	10
5	8	45	8.02	5376.70	7643.90	5.465e+8	0.05	10
5	16	45	16.63	5376.70	4.799e+5	1.551e+7	0.06	10
5	32	45	24.50	5376.70	2.674e+6	3.321e+5	0.06	10
5	64	45	105.46	5376.70	4.885e+8	1.478e+4	0.08	10
5	128	45	229.48	5376.70	3.042e+9	654.84	0.10	10
5	256	45	322.09	5376.70	1.844e+10	5.72	0.11	10
5	512	45	550.89	5376.70	2.224e+11	0.21	0.12	10
5	1024	45	981.99	5376.70	2.739e+13	0.02	0.13	10
5	2048	45	1672.04	5376.70	1.931e+13	0.001	0.14	9
10	4	45	5.02	61.36	2.347e+6	7.628e+6	0.7	10
10	8	45	13.57	61.36	3.499e+9	6381.64	1.02	10
10	16	45	20.42	61.36	1.323e+12	5.33	1.35	10
10	32	45	32.74	61.36	1.857e+13	0.01	1.59	5
10	64	45	33.30	61.36	2.506e+13	0.0	1.60	0
10	128	45	33.30	61.36	2.506e+13	0.0	1.77	0
15	4	45	7.81	22.50	5.174e+10	222.10	6.18	10
15	8	45	11.68	22.50	2.336e+13	0.01	8.52	2
15	16	45	11.49	22.50	2.558e+13	0.0	8.55	0
15	32	45	11.49	22.50	2.558e+13	0.0	8.53	0
20	2	45	4.08	14.66	7.225e+9	1.581e+4	17.44	10
20	4	45	7.55	14.66	2.879e+13	0.152	29.34	3
20	8	45	7.68	14.66	3.804e+13	0.0	29.38	0
20	16	45	7.68	14.66	3.804e+13	0.0	29.43	0
30	2	45	5.96	10.28	5.898e+13	0.03	84.17	2
30	4	45	5.74	10.28	6.016e+13	0.0	84.71	0
40	4	45	6.77	9.04	5.623e+13	0.0	243.63	0
60	4	45	6.45	8.46	6.972e+13	0.0	1371.09	0
80	4	45	5.51	8.54	8.896e+13	0.0	4105.09	0
100	4	45	5.92	8.82	1.107e+14	0.0	9988.76	0
120	4	45	5.63	9.16	1.087e+14	0.0	22399.28	0

Tabelle 2.2: Ergebnisse von SIRA für $\delta = 0.95$

n	ϵ^{-1}	Q	$\oslash \|a_n\|$	$\oslash \max \lambda(x)$	$\oslash \max_i \|b_i\|$	$\oslash \|x' - x\|$	$\oslash$ Sek.	$\sharp\ x'$
5	4	45	3.26	5376.70	466.03	1.413e+10	0.04	10
5	8	45	8.18	5376.70	9166.21	4.017e+8	0.05	10
5	16	45	14.82	5376.70	4.695e+5	6.210e+6	0.07	10
5	32	45	28.88	5376.70	4.136e+6	2.292e+5	0.07	10
5	64	45	73.44	5376.70	2.526e+8	3.498e+3	0.09	10
5	128	45	354.37	5376.70	3.451e+9	542.38	0.10	10
5	256	45	334.37	5376.70	4.758e+10	19.55	0.11	10
5	512	45	457.80	5376.70	1.323e+11	0.26	0.12	10
5	1024	45	762.26	5376.70	9.493e+11	0.02	0.16	10
5	2048	45	1849.20	5376.70	2.899e+13	0.001	0.18	10
10	4	45	4.44	61.36	5.685e+5	3.059e+7	0.95	10
10	8	45	7.69	61.36	2.222e+8	9173.23	1.18	10
10	16	45	18.31	61.36	2.450e+11	25.10	1.52	10
10	32	45	31.24	61.36	1.916e+13	0.01	1.84	7
10	64	45	31.35	61.36	2.384e+13	0.0	1.87	0
10	128	45	31.35	61.36	2.384e+13	0.0	1.85	0
15	4	45	9.52	22.50	2.787e+10	463.68	7.68	10
15	8	45	10.56	22.50	2.283e+13	0.02	10.65	5
15	16	45	10.37	22.50	2.568e+13	0.0	10.83	0
15	32	45	10.37	22.50	2.568e+13	0.0	10.84	0
20	2	45	4.06	14.66	4.406e+8	1.447e+5	20.40	10
20	4	45	6.39	14.66	3.086e+13	0.12	36.66	5
20	8	45	6.30	14.66	3.952e+13	0.0	37.79	0
20	16	45	6.30	14.66	3.952e+13	0.0	37.50	0
30	2	45	4.97	10.28	2.862e+13	0.04	124.98	2
30	4	45	4.92	10.28	3.191e+13	0.0	125.36	0
40	4	45	5.19	9.04	6.048e+13	0.0	447.33	0
60	4	45	4.99	8.46	6.912e+13	0.0	2171.60	0
80	4	45	4.80	8.54	8.234e+13	0.0	7287.19	0
100	4	45	4.85	8.82	8.050e+13	0.0	17690.77	0
120	4	45	5.55	9.16	1.341e+15	0.0	36313.27	0

Tabelle 2.3: Ergebnisse von rat–SIRA für $\delta = 0.95$

Der Test auf $\tau_{n,n} \neq 0$ erfolgt in der Implementierung von SIRA und rat–SIRA durch $|\tau_{n,n}| > 2^{-r}$, wobei $r = 53$ ($2^{-53} \approx 10^{-16}$) die maximale Anzahl der Genauigkeitsbits im IEEE 754–Standard für Gleitpunktzahlen in doppelter Genauigkeit ist. Im ersten Fall der Terminierung testen wir jeweils noch ab, ob $< x, a_n >= 0$. Ist dies nicht der Fall, so liefert der Abbruchtest $|\tau_{n,n}| > 2^{-53}$ ein falsches Prädikat. Wir nehmen dann eine neue zufällige Eingabe x .

In der Spalte $\sharp\, x'$ der Tabelle 2.1 steht die Anzahl der Fälle 2 der Terminierung. Die Werte jeder anderen Spalte sind Durchschnittswerte, gemittelt über die Ergebniswerte für 10 Eingaben x . Die Spalte $\oslash \|a_n\|$ gibt die durchschnittliche euklidische Länge des

Ausgabevektors a_n an. $\oslash$ $\max \lambda(x)$ ist die durchschnittliche (in Abschnitt 1.4 bewiesene) theoretische obere Schranke für die kürzeste Relation (vergleiche Satz 1.27). In der Spalte $\oslash$ $\max_i \|b_i\|$ sind die durschnittlichen maximalen euklidischen Längen der Basisvektoren $b_1, \ldots, b_n$ eingetragen. $\oslash$ $\|x' - x\|$ bezeichnet den durchschnittlichen euklidischen Abstand von x' zu x. In der Spalte $\oslash$ Sek. steht die durchschnittliche Laufzeit des Algorithmus in Sekunden.

Bei $Q = 45$ werden die 53 Genauigkeitsbits der Gleitpunktzahlen fast vollständig aufgebraucht. Für $Q > 45$ ist der Algorithmus nicht mehr stabil. Dann wird die numerische Genauigkeit kleiner als die Länge der Basisvektoren $b_1, \ldots, b_n$ und der Anteil der guten Austausche nimmt ab. Der Algorithmus bleibt für Werte $Q > 45$ in Endlosschleifen stecken.

Die in Abschnitt 2.2 (Satz 2.3 (3)) theoretisch nachgewiesene lineare Beziehung zwischen der Länge der Relation und dem Parameter ϵ^{-1} bei fester Dimension ist an Tabelle 2.1 gut erkennbar.

Für Dimensionen $n \geq 10$ können wir die numerische Stabilität von SIRA und rat–SIRA dadurch erhöhen, daß wir nicht erst nach der vollständigen Größenreduktion eines Vektors b_k, sondern jeweils nach den einzelnen Größenreduktionsschritten $b_k := b_k - \lceil \tau_{k,j} / \tau_{j,j} \rfloor \, b_j$ eine neue Orthonormalisierung mit Givens Rotation durchführen. Mit dieser Modifikation werden beide Algorithmen numerisch stabil bis Dimension 120. Die experimentellen Ergebnisse finden sich in den Tabellen 2.2 und 2.3.

Für gleiche Eingaben $x \in \mathbb{R}^n$, $\epsilon \in \mathbb{Q}_+$ ist die Laufzeit für rat–SIRA in hohen Dimensionen wesentlich größer. Dieser Effekt kommt durch die zusätzliche vollständige Größenreduktion des Vektors b_k vor Austauschen $b_{k-1} \leftrightarrow b_k$ während der L^3–Reduktion der projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_{n-1})$ in Schritt 2 zustande. Mit der Givens Rotation nach jedem Größenreduktionsschritt $b_k := b_k - \lceil \tau_{k,j} / \tau_{j,j} \rfloor \, b_j$ kostet die Größenreduktion von b_k dann $O(k\, n^3)$ statt $O(n^3)$ arithmetische Operationen. Die durchschnittliche Länge der gefundenen Relation ist dafür bei rat–SIRA etwas kürzer. Dies liegt an der besseren Reduziertheit der primären und somit auch dualen Basisvektoren während der L^3–Reduktion der projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_{n-1})$ in Schritt 2. Die Anzahl der gefundenen Nahebeipunkte ist hingegen für beide Algorithmen im wesentlichen gleich.

Die guten Laufzeiten von SIRA und rat–SIRA dokumentieren nicht nur die Effizienz des Verfahrens, sondern auch dessen numerische Stabilität; denn im Falle kurzer Laufzeiten führt der Algorithmus überwiegend gute Austausche aus. Die Tatsache, daß die Algorithmen SIRA und rat–SIRA Relationen für die Eingaben wie erwartet finden, beweist damit auch die numerische Stabilität der beiden Algorithmen.

Die Werte in der Spalte $\oslash$ $\|a_n\|$ belegen durch Vergleich mit der theoretischen oberen Schranke aus Satz 1.27 die Korrektheit des Verfahrens hinsichtlich der zu erwartenden Länge der Relation für die Eingabe x. Beispielsweise ist für $n = 10$ nach Tabelle 2.2 und Satz 1.27, respektive, $16 \leq \lambda(x) \leq 61.36$ für alle 10 Eingaben; entsprechend erhalten wir für Dimension $n = 15$ die Abschätzung $4 \leq \lambda(x) \leq 22.50$ für alle 10 Eingaben.

Numerische Stabilität anderer Orthonormalisierungsverfahren. Das Verfahren der Givens Rotation zur Berechnung der Orthonormalisierung von $b_0 := x, b_1, \ldots, b_n$ besitzt einen numerischen Fehler der linear in $n\, 2^{-r}\, |B|$ ist, wobei $\|B| := \max_{0 \leq i \leq n} \|b_i\|$. Verwenden wir das GSO–Verfahren zur Orthonormalisierung von $x, b_1, \ldots, b_n$, so ist der numerische Fehler der normalisierten Gram–Schmidt Koeffizienten $\tau_{i,j}$, $0 \leq i, j \leq n$, proportional zu

$$n\, 2^{-r}\, |B| / \min_{\substack{1 \leq i \leq n \\ \widehat{b}_{i,x} \neq 0}} \|\widehat{b}_{i,x}\|^{-1} \qquad \text{(vergleiche [GoL89], Seite 219.)}$$

In dem Beweis von Proposition 2.8 haben wir gesehen, daß Austausche $b_{n-1} \leftrightarrow b_n$ von SIRA das von den projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_{n-1})$ aufgespannte Volumen $d(L(\pi_x(b_1), \ldots, \pi_x(b_{n-1})))$ um den Faktor $\frac{1}{2}$ verkleinern. Die Längen $\tau_{i,i} = \|\widehat{b}_{i,x}\|$ der Höhen können während des Algorithmus damit beliebig klein werden. Dies hat zur Folge, daß die Berechnung der Gram–Schmidt Koeffizienten $\mu_{i,j} = \tau_{i,j}/\tau_{j,j}$ ungenau wird. Inkorrekt berechnete Gram–Schmidt Koeffizienten ergeben fehlerhafte und große Reduktionskoeffizienten. Dadurch werden die Basisvektoren $b_1, \ldots, b_n$ ebenfalls sehr groß und nur unzureichend orthogonal zueinander. Beides vergrößert wiederum den Fehler bei der Berechnung der Gram–Schmidt Koeffizienten. Der Algorithmus gerät schließlich in Endlosschleifen, in denen für Indizes $2 \leq k \leq n$ unendlich oft Vektoren b_{k-1}, b_k miteinander vertauscht werden.

Für das Verfahren der GSO, wie es im L^3–Algorithmus zur Orthonormalisierung der Gitterbasisvektoren verwendet wird [LLL82], können wir keine numerische Stabilität beweisen.

Wir haben für die Algorithmen SIRA und rat–SIRA das Verfahren der GSO getestet, wie es von Schnorr und Euchner in der Gleitpunktversion des L^3–Algorithmus vorgeschlagen wurde [SE94]. In [SE94] wurde gezeigt, daß durch zusätzliche Korrekturschritte bei Skalarproduktberechnungen[5] das GSO–Verfahren für nicht degenerierte Gitter numerisch stabil ist bis Dimension $n = 125$. Führen wir jedoch in SIRA und rat–SIRA die Orthonormalisierung von $x, b_1, \ldots, b_n$ mit der in [SE94] vorgestellten modifizierten GSO durch, so gerät der Algorithmus für zufällige Eingabevektoren $x = (p_1, \ldots, p_n)/q$ mit $2^{19} < \max_{1 \leq i \leq n} |p_i| \leq 2^{20}$, $q = 1$ schon bei Dimension $n = 3$ und $\epsilon^{-1} = 2^6$ in Endlosschleifen.

Schnorr [Sc88] gab eine Variante dieses Verfahrens an, die numerische Fehler bei der Berechnung der Gram–Schmidt Koeffizienten $\mu_{i,j}$ durch Selbstkorrekturschritte ausgleicht. Hierzu wird die inverse Matrix $(\nu_{i,j})_{1 \leq i,j \leq n} := (\mu_{i,j})^{-1}_{\substack{0 \leq i,j \leq n \\ \widehat{b}_{j,x} \neq 0}}$ durch Skalarprodukte $< b_i, b_j >$, $0 \leq i, j \leq n$ der in exakter ganzzahliger Arithmetik mitgeführten Basisvektoren aktualisiert. Mit dieser Form der Selbstkorrektur wird die Orthogonalisierung $(\mu_{i,j})_{0 \leq i,j \leq n}$ von $b_0 := x, b_1, \ldots, b_n$ neu berechnet.

[5] Bei der Berechnung von Skalarprodukten werden unterschiedlich große Summanden aufaddiert. Durch Rundung des Ergebnisses in der Gleitpunktarithmetik können die führenden Bits wegfallen [GoL89], Seiten 63–65.

Der Fehler der 'selbstkorrigierten' Gleitpunktwerte $\nu'_{i,j}$, $1 \leq i, j \leq n$ konvergiert quadratisch gegen 0 [Sc88], sofern die Selbstkorrektur in exakter rationaler Arithmetik implementiert und der absolute Fehler der Gleitpunktwerte der $\nu_{i,j}$, $1 \leq i, j \leq n$ zu Beginn der Selbstkorrektur nicht größer als 1 ist. Das Verfahren benötigt $12\,n + 3\,\lceil \log \max\{|p_i|\,,\ 1 \leq i \leq n\,,\ |q|\}\rceil$ Genauigkeitsbits und erschöpft damit die uns beim IEEE 754–Standard für Gleitpunktzahlen in doppelter Genauigkeit zur Verfügung stehenden 53 Präzisionsbits schon für Dimension $n = 3$ und zufälligen Eingabevektoren $x = (p_1, p_2, p_3)/q$ mit $\max_{1 \leq i \leq n} |p_i| \geq 2^{10}$, $q = 1$.

Wir haben das Orthogonalisierungverfahren von [Sc88] in SIRA und rat–SIRA getestet. Der Algorithmus läuft in Dimension $n = 3$ für zufällige 18–bit lange Eingabevektoren $x = (p_1, p_2, p_3)/q$ mit $q = 1$ und $\epsilon^{-1} = 2^6$ noch numerisch stabil, ebenso für Dimension $n = 5$, $\epsilon^{-1} = 2^6$ und 12–bit lange Eingabevektoren. Für höhere Dimensionen und größere Bitlängen der p_i , $1 \leq i \leq n$ bleibt der Algorithmus mit endlosen Austauschen benachbarter Vektoren b_{k-1}, b_k stecken; die Gleitpunktwerte der jeweiligen Gram–Schmidt Koeffizienten $\mu_{k,k-1}$ werden dann für das Abtesten der L^3–Bedingung nicht mehr genau genug berechnet. Dann hilft das Verfahren der Selbstkorrektur zur Angleichung der Gleitpunktwerte $\mu'_{i,j}$ an die exakten Werte $\mu_{i,j}$ nicht, da der Fehler der Gleitpunktwerte $\nu'_{i,j}$, $1 \leq i, j \leq n$ zu Beginn der Selbstkorrektur schon wesentlich größer als 1 ist.

Zusammenfassend läßt sich festhalten, daß für die Orthonormalisierung von fast linear abhängigen Systemen, wie sie in SIRA und rat–SIRA vorliegen, die Givens Rotation der GSO und allen GSO–ähnlichen Verfahren vorzuziehen ist; der numerische Fehler bei der Givens Rotation nimmt im Gegensatz zum Verfahren der GSO nicht mit kleiner werdender Determinante des Systems zu.

Kapitel 3

Ein stabiler höherdimensionaler Kettenbruchalgorithmus

Das Problem der (simultanen) diophantischen Approximation besteht darin, gegebene reelle Zahlen $x_1, \ldots, x_{n-1}$ durch rationale Zahlen $\frac{p_1}{q}, \ldots, \frac{p_{n-1}}{q}$ mit Hauptnenner q möglichst gut zu approximieren. Konvergiert der erste Vektor $(p_1, \ldots, p_{n-1}, q)$ einer Gitterbasis $\{b_1, \ldots, b_n\}$ des $\mathbb{Z}^n$ gegen die Gerade $(x_1, \ldots, x_{n-1}, 1)\,\mathbb{R}$ so sind die Brüche $\frac{p_1}{q}, \ldots, \frac{p_{n-1}}{q}$ eine 'gute' (simultane) diophantische Approximation an die reellen Zahlen $x_1, \ldots, x_{n-1}$. Nach einem Satz von Dirichlet gibt zu reellen Zahlen $x_1, \ldots, x_{n-1}$ unendlich viele (simultane) diophantische Approximationen $(p_1, \ldots, p_{n-1}, q) \in \mathbb{Z}^n$ mit $\max_{1 \le i \le n-1} |q\, x_i - p_i| < |q|^{\frac{1}{n-1}}$. Der Kettenbruchalgorithmus löst das Problem der diophantischen Approximation in Dimension 2. Der Kettenbruchalgorithmus berechnet eine Folge $((p^{(k)}, q^{(k)}))_{k \in \mathbb{N}}$ von *Bestapproximationen* an die Eingabe $x \in \mathbb{R}$, das heißt Zahlen $p^{(k)}, q^{(k)} \in \mathbb{Z}$, so daß $|q'\, x - p'| > |q^{(k)}\, x - p^{(k)}|$ für alle $p', q' \in \mathbb{Z}$ mit $|q'| < |q|$.

Wir zeigen in diesem Kapitel, daß SIRA für einen Eingabevektor $x = (x_1, \ldots, x_{n-1}, 1)$ gute diophantische Approximationen an $x_1, \ldots, x_{n-1}$ konstruiert, welche die Schranke des Existenzsatzes von Dirichlet bis auf einen Faktor $2^{(n+2)/4} \max_{1 \le i \le n-1} \sqrt{1 + x_i^2}$ erfüllen. Wir verbessern damit die von Just [Ju92] angegebene Schranke

$$\max_{1 \le i \le n-1} |x_i - p_i/q| \;\le\; 2^{\frac{n+2}{4}} \max_{1 \le i \le n-1} \sqrt{1 + x_i^2}\, /\, |q|^{1 + \frac{1}{2\,n\,(n-1)}} \,.$$

3.1 Diophantische Approximationen

Zu reellen Zahlen $x_1, \ldots, x_{n-1}$, $\epsilon > 0$ nennen wir einen ganzzahligen Vektor $b =: (p_1, \ldots, p_{n-1}, q)$ *(Simultane) Diophantische Approximation der Güte* ϵ, falls $\max_{1 \le i \le n-1} |q\, x_i - p_i| < \epsilon$. Nach Dirichlet gilt folgender

Satz 3.1 *[Di1842] Zu beliebigen reellen Zahlen* $x_1, \ldots, x_{n-1} \in \mathbb{R}$ *existieren unendlich viele Vektoren* $b =: (p_1, \ldots, p_{n-1}, q) \in \mathbb{Z}^n$, *so daß*

$$\max_{1 \leq i \leq n-1} |q\, x_i - p_i| \;<\; 1/|q|^{\frac{1}{n-1}} \;.$$

Bemerkung. *Die Schranke* $q^{-\frac{1}{n-1}}$ *ist bis auf einen konstanten Faktor optimal. Die Aussage des Satzes wird falsch, wenn man die rechte Seite der Ungleichung durch einen Term* $\frac{n-1}{n}/q^{\frac{1}{s}}$ *mit einem Exponenten* $s > 1/(n-1)$ *ersetzt [Bor03].*

Folgendes Lemma zeigt, daß eine Folge $(p_1^{(k)}, \ldots, p_{n-1}^{(k)}, q^{(k)})_{k \in \mathbb{N}}$ ganzzahliger Vektoren, die gegen die Gerade $(x_1, \ldots, x_{n-1}, 1)\,\mathbb{R}$ konvergiert, gute simultane diophantische Approximationen an die reellen Zahlen $x_1, \ldots, x_{n-1}$ liefert. Wir zeigen im nächsten Abschnitt, daß der in Kapitel 2 behandelte Stabile Relationenalgorithmus SIRA auf Eingaben $x \in \mathbb{R}^n$ genau eine solche gegen die Gerade $x\,\mathbb{R}$ konvergente Folge ganzzahliger Vektoren konstruiert.

Lemma 3.2 *[Ju92] Sei* $x := (x_1, \ldots, x_{n-1}, 1) \in \mathbb{R}^n$, $b := (p_1, \ldots, p_{n-1}, q) \in \mathbb{Z}^n$ *und* $\pi_x(b)$ *die zu* x *orthogonale Projektion des Vektors* b . *Dann gilt für* $i = 1, \ldots, n-1$:

$$|q\, x_i - p_i| \;\leq\; \|\pi_x(b)\| \sqrt{1 + x_i^2} \;.$$

Der Beweis folgt unmittelbar aus der Betrachtung der zu $x := (x_1, \ldots, x_{n-1}, 1)^\top$ orthogonalen Projektionen in der Ebene und geht auf Just [Ju92] (Ungleichung (18) und Gleichung (19), Seite 918) zurück.

3.2 Diophantische Approximation mit dem stabilen Relationenalgorithmus

Methode. Wir ändern den Algorithmus SIRA dahingehend ab, daß der Algorithmus nach der Größenreduktion der projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_n)$ den ersten Basisvektor b_1 ausgibt. Der Algorithmus gleicht damit im wesentlichen dem höherdimensionalen Kettenbruchalgorithmus von [Ju92]. In Analogie zum Stabilen Relationenalgorithmus aus Abschnitt 2.1 nennen wir den daraus resultierenden Algorithmus SCFA (Stable Continued Fraction Algorithm). Mit Hilfe von Satz 2.9 (2) zeigen wir, daß SCFA für Eingaben $x = (x_1, \ldots, x_{n-1}, 1) \in \mathbb{R}^n$ eine Folge von Vektoren $(p_1^{(k)}, \ldots, p_{n-1}^{(k)}, q^{(k)}) \in \mathbb{Z}^n$, $k = 1, 2, \ldots$ mit wachsendem Nenner $|q^{(k)}|$ berechnet, so daß $\max_{1 \leq i \leq n} |x_i\, q^{(k)} - {p_i}^{(k)}| \leq 2^{(n+2)/4} \sqrt{1 + x_i^2} \,/\, |p_n^{(k)}|^{1+\frac{1}{n-1}}$ gilt.

Stabiler Kettenbruchalgorithmus (SCFA)

EINGABE $x \in \mathbb{R}^n - \{0\}$, $\epsilon > 0$.

1. Initialisierung. $[b_0, b_1, \ldots, b_n] := [x, e_1, \ldots, e_n]$, $s := 1$,

berechne die Orthonormalisierung $(\tau_{i,j})_{\substack{0 \le i \le n \\ 1 \le j \le n}}$ von $[x, b_1, \ldots, b_n]$.

Falls $\tau_{n,n} > 0$, dann ist e_n eine Relation für x. Gebe den Punkt $x' := x$ und die Relation $a_n := e_n$ für x aus, und stoppe.

2. *L^3-Reduktion von $\pi_x(b_1), \ldots, \pi_x(b_{n-1})$.*

WHILE ($\exists 1 < k < n$: $\frac{3}{4}\tau_{k-1,k-1}^2 > \tau_{k,k}^2 + \tau_{k,k-1}^2$) DO

Setze $b_k := b_k - \lceil \tau_{k,k-1}/\tau_{k-1,k-1} \rfloor b_{k-1}$;

vertausche b_{k-1} und b_k und berechne die Orthonormalisierung $(\tau_{i,j})_{\substack{0 \le i \le n \\ 1 \le j \le n}}$ neu.

Größenreduziere alle projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_n)$.

WHILE $|\tau_{s,s}| \le \epsilon$ DO $s := s + 1$.

Gebe $(p_1, \ldots, p_{n-1}, q) := b_1$, die nächste Approximation an x aus.

3. *Austausch $b_{n-1} \leftrightarrow b_n$.*

Vertausche b_{n-1} und b_n und berechne die Orthonormalisierung $(\tau_{i,j})_{\substack{0 \le i \le n \\ 1 \le j \le n}}$ neu.

Falls $\tau_{n,n} = 0$ und $s < n$ dann gehe nach 2.

4. *Abbruch.* Berechne $[a_1, \ldots, a_n]^\top := [b_1, \ldots, b_n]^{-1}$.

Im Falle $\tau_{n,n} > 0$ ist a_n Relation für x. Gebe den Punkt $x' := x$ und a_n aus, und stoppe. Falls $s = n$, dann gilt $\tau_{i,i} \le \epsilon$ für $i = 1, \ldots, n$, und es existiert keine Relation für x mit Länge kleiner als ϵ^{-1}. Berechne in diesem Fall $\pi_n(x) = < x, \widehat{b}_n/\|\widehat{b}_n\| > \widehat{b}_n/\|\widehat{b}_n\| \in span(b_1, \ldots, b_{n-1})^\perp$, und gebe den Punkt $x' := x - \pi_n(x)$, die Relation a_n für x' sowie '$\lambda(x) \ge 1/\epsilon$' aus.

Falls $\epsilon = 0$, gibt SCFA eine eventuell unendliche Folge von Vektoren b_1 nach der L^3–Reduktion in Schritt 2 aus, welche gute diophantische Approximationen an x sind. SCFA stoppt im Falle $\epsilon = 0$ genau dann, wenn $\dim(L_x) \ge 1$ ist und findet in diesem Falle eine Relation a_n zu x.

Satz 3.3 *Für Eingaben $x = (x_1, \ldots, x_{n-1}, 1)$, $\epsilon \in \mathbb{R}_+$ berechnet SCFA eine Folge von Vektoren $(p_1, \ldots, p_{n-1}, q) := b_1$ und gibt diese nach der L^3-Reduktion in Schritt 2 aus. Die Vektoren $(p_1, \ldots, p_{n-1}, q)$ erfüllen für $i = 1, \ldots, n-1$:*

$$|x_i - p_i/q| \quad \le \quad 2^{\frac{n+2}{4}} \sqrt{1 + x_i^2} \,/\, |q|^{1+\frac{1}{n-1}} \, . \tag{3.1}$$

Nach dem Existenzsatz von Dirichlet [Di1842] sind die Vektoren $(p_1, \ldots, p_{n-1}, q)$ in dem Sinne bis auf den Faktor $2^{(n+2)/4}\|x\|$ *optimal,* als daß der Exponent $1 + 1/(n-1)$ im allgemeinen nicht erhöht werden kann. Wir verbessern damit die von Just [Ju92] bewiesene Schranke von $|x_i - p_i/q| \le 2^{\frac{n+2}{4}} \sqrt{1 + x_i^2} \,/\, |q|^{1+\frac{1}{2\,n\,(n-1)}}$.

Beweis. Für $n = 2$ ist die Folge p_1/q der rationalen Zahlen nach einem Austausch $b_1 \leftrightarrow b_2$ der Bruch der Kettenbruchentwicklung von x_1. In diesem Falle gilt die bessere und scharfe Schranke $|x_1 - p_1/q| \le 1/|q|^2$ (vergleiche [Kh63], Satz 9, Seite 9).

Sei nun $n \geq 3$. Wegen der L^3–Reduziertheit der projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_{n-1})$ folgt aus den Sätzen 1.22 (1) und 2.5 (2)

$$\|b_1\| \;\leq\; 2^{n/2}\,\epsilon^{-1} \prod_{j=2}^{n-1} \|\widehat{b}_{j,x}\|^{-1} + \|\widehat{b}_{1,x}\| \;\leq\; 2^{n/2}\,\epsilon^{-1}\,\|\widehat{b}_{1,x}\|^{-n+2}\,2^{(n-1)\,(n-2)/4} + 1\;.$$

Diese Ungleichung kann auch für $\epsilon = \|\widehat{b}_{1,x}\|$ gezeigt werden (vergleiche die Bemerkung im Anschluß an Satz 2.5 (2)). Wir erhalten

$$\|b_1\| \;\leq\; \|\widehat{b}_{1,x}\|^{-n+1}\,2^{n/2}\cdot 2^{(n-1)\,(n-2)/4} + 1 \;\stackrel{n\geq 3}{\leq}\; \|\widehat{b}_{1,x}\|^{-n+1}\,2^{(n-1)\,(n+2)/4}\,,$$

$$\|\widehat{b}_{1,x}\| \;\leq\; \|b_1\|^{-\frac{1}{n-1}}\,2^{(n+2)/4} \;\leq\; |q|^{-\frac{1}{n-1}}\,2^{(n+2)/4}\,.$$

Mit den Ungleichungen

$$|q\,x_i - p_i| \;\leq\; \|\widehat{b}_{1,x}\|\,\sqrt{1+x_i^2}\;,\;\; i = 1, \ldots, n-1\,,$$

aus Lemma 3.2 folgt die Behauptung. □

Bemerkungen. *1. Für den Fall $\|x\| = \sqrt{2}$ bzw. $\sum_{i=1}^{n-1} x_i^2 = 1$ verlieren wir in Ungleichung (3.1) gegenüber der Dirichlet–Schranke aus Satz 3.1 den Faktor $2^{n/4+1}$. Dieser exponentielle Faktor wird durch das in Schritt 2 verwendete L^3–Reduktionsverfahren bedingt. Bei Verwendung von Algorithmen zur Block–Reduktion der projizierten Vektoren $\pi_x(b_1), \ldots, \pi_x(b_n)$ kann man den Faktor $2^{(n+2)/4}$ auf einen Faktor $(1+\varepsilon)^{n/4}$ mit beliebig kleinem $\varepsilon > 0$ reduzieren [Sc87]. Diese Algorithmen sind allerdings nicht mehr polynomial in der Dimension n.*

*2. Umgekehrt ist das Problem, zu gegebenen $x = (x_1, \ldots, x_{n-1}, 1) \in \mathbf{Q}^n$, $\epsilon \in \mathbf{Q}_+$ und $N \in \mathbb{N}$ zu entscheiden, ob ganze Zahlen $p_1, \ldots, p_{n-1}, q$ mit $q \in [1, N]$ und $\max_{1\leq i\leq n} |q\,x_i - p_i| \leq \epsilon$ existieren, **NP**–vollständig [Lag85]. Es ist daher nicht zu erwarten, daß es polynomial–Zeit Algorithmen gibt, die für Eingaben $x = (x_1, \ldots, x_{n-1}, 1) \in \mathbf{Q}^n$ Vektoren $(p_1, \ldots, p_{n-1}, q)$ berechnen, die die Schranke aus Satz 3.3 mit rechter Seite $\sqrt{1+x_i^2}/q^{1+\frac{1}{n-1}}$ ohne einen exponentiellen Faktor in n erfüllen.*

3. Ein Vektor $b =: (p_1, \ldots, p_{n-1}, q)^\top \in \mathbb{Z}^{n-1}\times\mathbb{N}$ heißt gemäß [Bre81] Bestapproximation *an $x := (x_1, \ldots, x_{n-1}, 1)^\top \in \mathbb{R}^n$, falls für alle $\overline{b} =: (\overline{p}_1, \ldots, \overline{p}_{n-1}, \overline{q}) \in \mathbb{Z}^{n-1}\times\mathbb{N}$ mit $\overline{q} \leq q$ gilt, daß*

$$\max_{1\leq i\leq n-1} |\overline{q}\,x_i - \overline{p}_i| \;\geq\; \max_{1\leq i\leq n-1} |q\,x_i - p_i|\;.$$

Es wäre interessant zu untersuchen, inwieweit der Algorithmus SCFA Bestapproximationen an x ausgibt. Da die Schranke des Existenzsatzes von Dirichlet im oben angegebenen Sinne scharf ist, liegt die Vermutung nahe, daß SCFA Bestapproximationen an x bis auf einen Faktor $2^{(n+2)/4}\max_{1\leq i\leq n-1}\sqrt{1+x_i^2}$ berechnet.

Kapitel 4

Approximierbarkeit kürzester Relationen

In diesem Kapitel zeigen wir unter der Annahme $\mathbf{NP} \not\subseteq \mathbf{QP}$[1] eine untere Schranke für die Approximierbarkeit kürzester Relationen in der Maximum–Norm:

Es existiert kein polynomial–Zeit Algorithmus, der zu gegebenem $x \in \mathbf{Q}^n$ eine in der ∞–Norm bis auf den Faktor $2^{\log^{0.5-\zeta} bin(x)}$ kürzeste Relation findet, wobei ζ eine beliebig kleine positive Konstante und $bin(x)$ die binäre Länge der Eingabe x ist.

Im Gegensatz zu diesem Resultat kann die in der euklidischen Länge kürzeste Relation bis auf den Faktor $2^{n/2-1}$ approximiert werden ([HJLS89], Kapitel 6).

Der Beweis setzt sich aus 3 Reduktionen zusammen. Als erstes geben wir eine quasipolynomial–Zeit Reduktion der Erfüllbarkeit einer 3–Term konjunktiven Normalform auf ein kombinatorisches Optimierungsproblem auf Graphen an (Paragraph 2). Dieses Ergebnis geht auf Feige, Lovasz [FL92] zurück und wurde von Arora, Babai, Stern, Sweedyk [ABSS93] in exakter Form dargestellt. Als nächstes reduzieren wir dieses Optimierungsproblem auf das Problem, die in der ∞–Norm minimale ganzzahlige Lösung eines homogenen linearen Gleichungssystem zu finden (Paragraph 3). Wir verwenden hierzu die Techniken aus [ABSS93]. Schließlich transformieren wir den in der ∞–Norm beschränkten Anteil der ganzzahligen Lösungsmenge des homogenen linearen Gleichungssystems in den in der ∞–Norm beschränkten Anteil der ganzzahligen Lösungsmenge einer einzelnen homogenen linearen Gleichung (Paragraph 4). Wir erhalten damit eine quasipolynomial–Zeit Reduktion des Problems der Erfüllbarkeit einer 3–Term konjunktiven Normalform auf die Approximierbarkeit kürzester Relationen in der ∞–Norm. Aus der **NP**–Vollständigkeit des erstgenannten Problems folgt unser Resultat. Im 1. Paragraphen führen wir grundlegende Begriffe ein.

Der Inhalt dieses Kapitels entspricht im wesentlichen dem der Arbeit [RSe96a].

[1] Da für die Entscheidbarkeit von Sprachen aus **NP** keine polynomial–Zeit und keine quasipolynomial–Zeit Algorithmen bisher bekannt sind, wird vermutet, daß $\mathbf{NP} \neq \mathbf{P}$ sowie $\mathbf{NP} \not\subseteq \mathbf{QP}$ gilt. Die Annahme $\mathbf{NP} \neq \mathbf{P}$ ist schwächer, da aus $\mathbf{NP} = \mathbf{P}$ natürlich $\mathbf{NP} \subseteq \mathbf{QP}$ folgt.

4.1 Grundlagen

Folgende Begriffe gehen auf Creszenzi, Panconesi [CP91] und Papadimitriou, Yannakakis [PY91] zurück.

Definition 4.1 *Ein* Optimierungsproblem Π *ist eine Menge* $\mathcal{I}_\Pi \subseteq \{0,1\}^*$ *von* Eingaben I, *eine Menge* $\mathcal{S}_\Pi \subseteq \{0,1\}^*$ möglicher Lösungen S *für die Eingaben* $I \in \mathcal{I}_\Pi$, *eine polynomial–Zeit berechenbare* Maßfunktion $m_\Pi : \mathcal{I}_\Pi \times \{0,1\}^* \to \mathbb{R}_+$ *und das* Ziel *der Optimierung von* m_Π, *entweder* Minimierung *oder* Maximierung. *Die Funktion* m_Π *weist einer Eingabe* $I \in \mathcal{I}_\Pi$ *und einer Lösung* $S \in \mathcal{S}_\Pi(I)$ *von* I *einen* Lösungswert $m_\Pi(I,S)$ *zu.*

Das Optimierungsproblem besteht darin, für gegebenes $I \in \mathcal{I}_\Pi$ *eine Lösung* $S \in \mathcal{S}_\Pi(I)$ *für* I *zu finden, so daß* $m_\Pi(I,S)$ *optimal für alle möglichen* $S \in \mathcal{S}_\Pi(I)$ *ist, das heißt* $m_\Pi(I,S) = \min_\Pi\{m_\Pi(I,S') : S' \in \mathcal{S}_\Pi(I)\}$ *im Falle eines* Minimierungsproblems *und* $m_\Pi(I,S) = \max_\Pi\{m_\Pi(I,S') : S' \in \mathcal{S}_\Pi(I)\}$ *im Falle eines* Maximierungsproblems.

*Für **NP**–Optimierungsprobleme muß folgende weitere Bedingung erfüllt sein:*

Für die Menge $\mathcal{S}_\Pi$ *muß ein* $k \in \mathbb{N}$ *und eine polynomial–Zeit berechenbares Prädikat* $P : \mathcal{I} \times \{0,1\}^* \to \{0,1\}$ *existieren, so daß für alle* $I \in \mathcal{I}_\Pi$:

$$\mathcal{S}_\Pi \;=\; \{y \in \{0,1\}^* \,:\, |y| \le |I|^k \wedge P(I,y)\} \;.$$

Die **NP**–Minimierungsprobleme *Kürzeste Relation in der* ∞*–Norm* SIR_∞ (Shortest Integer Relation in ∞–norm) und *Kürzeste Simultane Relation in der* ∞*–Norm* $SSIR_\infty$ (Shortest Simultaneous Integer Relation in ∞–norm) sind wie folgt definiert:

$\underline{SIR_\infty\ :}$
GEGEBEN ein nicht trivialer rationaler Vektor $x \in \mathbb{Q}^n$.
FINDE einen ganzzahligen Vektor $m \neq 0$ mit $< m, x >= 0$ und minimaler ∞–Norm $\|m\|_\infty := \max_{1 \le i \le n} |m_i|$.

$\underline{SSIR_\infty\ :}$
GEGEBEN r nicht triviale rationale Vektoren $y_1, \ldots, y_r \in \mathbb{Q}^n$.
FINDE eine in der ∞–Norm minimale simultane (ganzzahlige) Relation m für $y_1, \ldots, y_r$, das heißt einen ganzzahligen Vektor $m \neq 0$ mit $< m, y_j >= 0$, $j = 1, \ldots, r$, und minimaler ∞–Norm $\|m\|_\infty$.

Statt $SSIR_\infty$ werden wir im folgenden das zu $SSIR_\infty$ (hinsichtlich der Lösungsmenge) äquivalente Minimierungsproblem *Minimale* $\mathbb{Z}$*–Lösung von Homogenem Gleichungssystem in der* ∞*–Norm* $Min\ \mathbb{Z} - HLS_\infty$ (Minimal $\mathbb{Z}$–Solution of Homogeneous Linear System of Equations in ∞–norm) betrachten:

$\underline{Min\ \mathbb{Z} - HLS_\infty\ :}$
GEGEBEN ein homogenes System $Ax = 0$ in r Gleichungen und n Variablen mit $A \in M(r, n, \mathbb{Q})$.
FINDE einen in der ∞–Norm minimalen ganzzahligen Vektor $x \neq 0$ mit $Ax = 0$.

Wir betrachten im folgenden Optimierungsprobleme dieser Art.

Definition 4.2 *Sei Π ein Minimierungs–(bzw. Maximierungs–)Problem. Wir sagen, Π wird in Zeit $t(n)$ bis auf einen Faktor $f(n) \geq 1$ approximiert, falls ein Algorithmus A existiert, der für alle $n \in \mathbb{N}$ und alle Eingaben $I \in \Pi$ mit $n := |I|$ sowie optimalem Lösungswert $opt_\Pi(I)$ $O(t(n))$ Rechenschritte ausführt und eine Lösung $A(I) \in \mathcal{S}_\Pi(I)$ ausgibt, so daß*

$$m_\Pi(I, A(I)) \leq opt_\Pi(I)\, f(n) \qquad (\text{ respektive } m_\Pi(I, A(I)) \geq opt_\Pi(I)/f(n)\ ) \quad .$$

Für die Betrachtung der Approximierbarkeit von Optimierungsproblemen benutzen wir folgenden Reduktionsbegriff, der auf Arora [Ar94] zurückgeht.

Definition 4.3 *Seien Π und Π' zwei Minimierungs–(bzw. Maximierungs–)Probleme und ρ, $\rho' \geq 1$. Eine* gap–erhaltende Reduktion *von Π auf Π' mit Parametern $((c, \rho), (c', \rho'))$ ist eine polynomial–Zeit Abbildung τ, welche Eingaben $I \in \Pi$ auf Eingaben $I' = \tau(I) \in \Pi'$ abbildet, so daß für die Minima (bzw. Maxima) $opt_\Pi(I)$ und $opt_{\Pi'}(I')$ von I bzw. I' folgendes gilt:*

$$\begin{aligned} opt_\Pi(I) \leq c &\implies opt_{\Pi'}(I') \leq c' && (4.1)\\ (\text{ respektive } opt_\Pi(I) \geq c &\implies opt_{\Pi'}(I') \geq c'\ ) \\ opt_\Pi(I) > c\,\rho &\implies opt_{\Pi'}(I') > c'\,\rho' && (4.2)\\ (\text{ respektive } opt_\Pi(I) < c/\rho &\implies opt_{\Pi'}(I') < c'/\rho'\ ) \quad . \end{aligned}$$

c, ρ bzw. c', ρ' hängen dabei von der jeweiligen binären Länge der Eingaben I bzw. I' ab.

Bemerkung. *Eine gap–erhaltende Reduktion τ unterliegt bei Eingaben I eines Minimierungs–(bzw. Maximierungs–)Problems Π für $c\,\rho \geq opt_\Pi(I) > c$ (respektive $c/\rho \leq opt_\Pi(I) < c$) keinerlei Bedingungen. Daher ist dieser Reduktionsbegriff schwächer als der von Papdimitriou, Yannakakis [PY91] eingeführte Begriff der* L–Reduktion[2]*, für den Bedingung (4.2) mit $\rho = \rho' = 1$ gilt.*

Ein Approximationsalgorithmus für ein Optimierungsproblem Π zusammen mit einer L–Reduktion τ von Π auf Π' definiert einen Approximationsalgorithmus für Π. Dies gilt jedoch nicht für gap–erhaltende Reduktionen. Falls jedoch kein polynomial–Zeit Algorithmus Eingaben I eines Minimierungsproblems Π mit $opt_\Pi(I) \leq 1$ und $opt_\Pi(I) > \rho$ unterscheiden kann, dann kann bei Existenz einer gap–erhaltenden Reduktion von Π auf Π' mit Parametern $((1, \rho), (1, \rho'))$ kein polynomial–Zeit Algorithmus Eingaben I' von Π' mit

[2]Eine L–Reduktion von einem Optimierungsproblem Π auf ein Optimierungsproblem Π' ist ein Paar von polynomial–Zeit Abbildungen (τ, κ), für die Konstanten $\alpha, \beta > 0$ existieren, so daß folgendes gilt:

1. Jede Eingabe $I \in \Pi$ wird auf eine Eingabe $I' = \tau(I) \in \Pi'$ abgebildet, und für die Optima $opt_\Pi(I)$ und $opt_{\Pi'}(I')$ von I bzw. I' ist $opt_\Pi(I) \leq \alpha\, opt_{\Pi'}(I')$.
2. Jede Lösung $S' \in \mathcal{S}_{\Pi'}(I')$ von I' wird durch κ auf eine Lösung $S = \kappa(S') \in \mathcal{S}_\Pi(I)$ von I abgebildet und für die entsprechenden Lösungswerte $m_\Pi(I, S)$ bzw. $m_\Pi(I', S')$ ist $|m_\Pi(I, S) - opt_\Pi(I)| \leq \beta\, |m_{\Pi'}(I', S') - opt_{\Pi'}(I')|$.

.

$opt_{\Pi'}(I') \leq 1$ *und* $opt_{\Pi'}(I') > \rho'$ *unterscheiden.*

Definition 4.4 *Seien* $x_1, \ldots, x_n$ *Boole'sche Variablen. Ein Literal ist entweder eine Boole'sche Variable oder ihre Negation. Eine* 3–Term Konjunktive Normalform $3 - \mathcal{C}NF$ $\phi(x_1, \ldots, x_n)$ *ist die Konjunktion von Klauseln* C_i , $i = 1, \ldots, m$, *welche jeweils aus der Disjunktion von 3 Literalen* $l_{i_1}, l_{i_2}, l_{i_3}$ *bestehen:*

$$\phi(x_1, \ldots, x_n) \;=\; \bigwedge_{i=1}^{m} (l_{i_1} \vee l_{i_2} \vee l_{i_3}) \;.$$

Eine erfüllende Belegung $a_1, \ldots, a_n$ *einer* $3 - \mathcal{C}NF$ $\phi(x_1, \ldots, x_n)$ *ist eine Belegung* $a_1, \ldots, a_n \in \{0, 1\}$ *der Boole'schen Variablen* $x_1, \ldots, x_n$, *so daß* $\phi(a_1, \ldots, a_n) = 1$.

Das Entscheidungsproblem 3–Erfüllbarkeit 3–SAT *(Satisfiability for* $3 - \mathcal{C}NF$*) ist das Problem zu entscheiden, ob eine erfüllende Belegung für eine gegebene* $3 - \mathcal{C}NF$ *existiert.*

Satz 4.5 *[Co71] 3–SAT ist* ***NP****–vollständig.*

Bemerkung. *3–SAT ist ein grundlegendes* ***NP****–vollständiges Problem. Cook [Co71] zeigte, daß jede von nicht–deterministischen polynomial–Zeit Turing Maschinen akzeptierte Sprache auf 3–SAT reduziert werden kann.*

4.2 Minimale Pseudo–Markenüberdeckung

In diesem Abschnitt skizzieren wir kurz ein Hauptresultat der Arbeit [ABSS93], eine quasipolynomial–Zeit gap–erhaltende Reduktion von 3–SAT auf das Minimierungsproblem *Minimale Pseudo–Markenüberdeckung in* ∞*–Kosten* $Min\ PLC_\infty$ (Minimum Pseudo Label Cover in ∞–costs). Wir werden hierzu gemäß [ABSS93] einige Begriffe einführen:

Sei im folgenden $G = (V_1 \cup V_2, E)$ ein (ungerichteter) bipartiter Graph mit disjunkten Knotenmengen V_1, V_2 und Kantenmenge $E \subseteq V_1 \times V_2$. Sei $\mathcal{B}$ eine Menge von Marken für die Knoten in $V_1 \cup V_2$, und für jede Kante $e \in E$ existiere eine partielle Funktion $\Pi_e : \mathcal{B} \to \mathcal{B}$.

Wir nehmen weiter an, daß der Graph G regulär ist, das heißt, es existieren natürliche Zahlen $d_1, d_2 \in \mathbb{N}$, so daß für $i = 1, 2$ jeder Knoten in V_i genau zu d_i Kanten inzident ist. Diese Eigenschaft von G wird in der Reduktion von 3–SAT auf $Min\ PLC_\infty$ in [ABSS93] benötigt. Es genügt sogar die schwächere Voraussetzung, daß es natürliche Zahlen $d_1, d_2 \in \mathbb{N}$ gibt, so daß für $i = 1, 2$ jeder Knoten in V_i zu <u>mindestens</u> d_i Kanten inzident ist.

Definition 4.6 *Eine* Markierung *des Graphen* $G = (V_1 \cup V_2, E)$ *ist ein Tupel* $(\mathcal{P}_1, \mathcal{P}_2)$ *von Funktionen* $\mathcal{P}_i : V_i \to 2^{\mathcal{B}}$, $i = 1, 2$, *die jedem Knoten in* $V_1 \cup V_2$ *eine möglicherweise leere Menge von Marken zuweist.*

Definition 4.7 *Sei* $(\mathcal{P}_1, \mathcal{P}_2)$ *eine Markierung von* $G = (V_1 \cup V_2, E)$ *und* $e = (v_1, v_2) \in V_1 \times V_2$ *eine Kante von* G . *Wir nennen eine Kante* $e = (v_1, v_2)$

$$\begin{array}{rcl} \text{unberührt} & :\Leftrightarrow & \mathcal{P}_1(v_1) = \mathcal{P}_2(v_2) = \emptyset\,, \\ \text{überdeckt} & :\Leftrightarrow & \mathcal{P}_1(v_1) \neq \emptyset\,,\ \mathcal{P}_2(v_2) \neq \emptyset \\ & & \wedge\ \forall\, b_2 \in \mathcal{P}_2(v_2)\ \exists b_1 \in \mathcal{P}_1(v_1)\ :\ \Pi_e(b_1) = b_2\,, \\ \text{gelöscht} & :\Leftrightarrow & \mathcal{P}_2(v_2) = \emptyset\,,\ \mathcal{P}_1(v_1) \neq \emptyset \\ & & \wedge\ \forall\, b_1 \in \mathcal{P}_1(v_1)\ \exists b_1' \in \mathcal{P}_1(v_1)\ \mathit{mit}\ b_1' \neq b_1\ : \\ & & \Pi_e(b_1) = \ b_2\ = \Pi_e(b_1')\ \mathit{für\ eine\ Marke}\ b_2 \in \mathcal{B}\,. \end{array}$$

Eine Markierung $(\mathcal{P}_1, \mathcal{P}_2)$ *von* $G = (V_1 \cup V_2, E)$ *heißt* Pseudo–Überdeckung *von* G , *falls*

(i) $\bigcup_{(v_1,v_2) \in V_1 \times V_2} \mathcal{P}_1(v_1) \cup \mathcal{P}_2(v_2) \neq \emptyset$ *und*

(ii) jede Kante von G *ist durch die Markierung* $(\mathcal{P}_1, \mathcal{P}_2)$ *entweder unberührt, überdeckt oder gelöscht.*

Eine Pseudo–Überdeckung heißt Totalüberdeckung*, falls jede Kante von der Markierung* $(\mathcal{P}_1, \mathcal{P}_2)$ *überdeckt wird.*

Eine Intuition dieser sehr künstlichen Definition geben wir in Anschluß an Satz 4.9 in Verbindung mit der Konstruktion von [ABSS93].

Definition 4.8 *Die* ∞–Kosten einer Markierung $(\mathcal{P}_1, \mathcal{P}_2)$ *von* $G = (V_1 \cup V_2, E)$ *sind definiert durch*

$$\mathit{cost}(\mathcal{P}_1, \mathcal{P}_2) \quad := \quad \max_{v_1 \in V_1} |\mathcal{P}_1(v_1)|\ .$$

Wir definieren nun das Minimierungsproblem

$\underline{Min\ PLC_\infty\ :}$

GEGEBEN ein regulärer bipartiter Graph $G = (V_1 \cup V_2, E)$ mit disjunkten Knotenmengen V_1, V_2 und Kantenmenge $E \subseteq V_1 \times V_2$, eine Menge von Marken $\mathcal{B} = \{1, \ldots, \mathcal{N}\}$, $\mathcal{N} \in \mathbb{N}$, und für jede Kante $e \in E$ eine partielle Funktion $\Pi_e : \mathcal{B} \to \mathcal{B}$, so daß für alle $e \in E$ die Marke 1 ein Urbild unter Π_e besitzt, das heißt $\forall e \in E\ : \exists b \in \mathcal{B} : \Pi_e(b) = 1$.
FINDE eine Pseudo–Überdeckung $(\mathcal{P}_1, \mathcal{P}_2)$ von G mit minimalen ∞–Kosten $\mathit{cost}(\mathcal{P}_1, \mathcal{P}_2)$.

Offenbar existiert stets eine Pseudo–Überdeckung $(\mathcal{P}_1, \mathcal{P}_2)$ von G mit ∞–Kosten von maximal $\mathcal{N}$; man setzt $\mathcal{P}_2(v_2) := \{1\}$ für alle $v_2 \in V_2$ und $\mathcal{P}_1(v_1) := \mathcal{B}$ für alle $v_1 \in V_1$.

Lemma 4.9 *[ABSS93], Lemma 9*
Es existiert eine quasipolynomial–Zeit Abbildung, die jeder 3–Term Konjunktiven Normalformen ϕ eine Eingabe $\tau(\phi)$ von $Min\ PLC_\infty$ zuordnet, so daß folgendes gilt:
Ist $\phi \in 3-SAT$, so existiert eine Pseudo–Überdeckung $(\mathcal{P}_1, \mathcal{P}_2)$ von $\tau(\phi)$ mit $cost(\mathcal{P}_1, \mathcal{P}_2) = 1$. Falls $\phi \notin 3-SAT$, dann gilt für alle Pseudo–Überdeckungen $(\mathcal{P}_1, \mathcal{P}_2)$ von $\tau(\phi)$, daß $cost(\mathcal{P}_1, \mathcal{P}_2) > 2^{\log^{0.5-\zeta} N}$, wobei ζ eine beliebig kleine positive Konstante und N die binäre Länge von $\tau(\phi)$ ist.

Wir werden im folgenden nur *gültige Marken eines Knotens* $v_1 \in V_1$ benutzen; gültige Marken $b_1 \in \mathcal{B}$ eines Knotens $v_1 \in V_1$ sind Marken, für die der Wert $\Pi_e(b_1)$ der partiellen Funktion Π_e für alle zu v_1 inzidenten Kanten $e \in E$ definiert ist. Die Beschränkung auf gültige Marken ist keine Einschränkung: Falls $\phi \in 3-SAT$, dann existiert eine Total–Überdeckung $(\mathcal{P}_1, \mathcal{P}_2)$ des Graphen $\tau(\phi)$ mit (minimalen) ∞–Kosten $cost(\mathcal{P}_1, \mathcal{P}_2) = 1$. $(\mathcal{P}_1, \mathcal{P}_2)$ überdeckt dann alle inzidenten Kanten eines jeden Knotens $v_1 \in V_1$ mit jeweils genau einer Marke, die für v_1 jeweils gültig sein muß. Falls $\phi \notin 3-SAT$, so erhöht die Einschränkung der Menge der Marken $b_1 \in \mathcal{B}$ auf gültige Marken für Knoten $v_1 \in V_1$ lediglich die ∞–Kosten $cost(\mathcal{P}_1, \mathcal{P}_2)$ der minimalen Pseudo–Überdeckung $(\mathcal{P}_1, \mathcal{P}_2)$ von $\tau(\phi)$.

Bemerkung. *Im Beweis von [ABSS93] wird folgende in [FL92, ALMSS92] bewiesene Eigenschaft von Sprachen aus* ***NP*** *benutzt: Jede Sprache in* ***NP****, also insbesondere* 3*–SAT, besitzt ein* (Multi–) Interaktives Beweissystem mit 2 Beweisern und 1 Runde $MIP(2,1)$ *(2–prover 1–round (Multi–) Interactive Proof System). Ein $MIP(2,1)$ besteht aus einem probabilistisch beschränkten polynomial–Zeit Verifizierer —einer polynomial–Zeit Turing Maschine, welche mit polylogarithmisch vielen*[3] *internen Zufallsbits, arbeitet— und* 2 *Beweisern, welche über unbegrenzte Rechenkapazität verfügen, aber nicht miteinander kommunizieren dürfen. Die Beweiser versuchen, den Verifizierer davon zu überzeugen, daß eine $3-CNF$ ϕ in der* ***NP****–Sprache 3–SAT enthalten ist. Der Verifizierer prüft den Beweis für '$\phi \in 3-SAT$' durch (polylogarithmisch viele) zufällige Fragen an die Beweiser ab. (polylogarithmischer Länge) Falls $\phi \in 3-SAT$, akzeptiert der Verifizierer den Beweis für '$\phi \in 3-SAT$'. Falls $\phi \notin 3-SAT$, verwirft der Verifizierer alle Beweise für '$\phi \in 3-SAT$' mit überwältigender Wahrscheinlichkeit, wobei sich die Wahrscheinlichkeit auf die vom Verifizierer zufällig gewählten Fragen bezieht.*

Die Hauptidee der in [ABSS93] angegebenen quasipolynomial–Zeit Reduktion besteht in der Übersetzung der Strategie der Beweiser, den Verifizierer von der Gültigkeit ihres Beweises für '$\phi \in 3-SAT$' zu überzeugen und somit zum Akzeptieren ihres Beweises zu bewegen, in eine entsprechende Eingabe von $Min\ PLC_\infty$. Hierbei werden spezielle 'geometrische' Eigenschaften von $MIP(2,1)$ aus [FL92] ausgenutzt und mit einigen Modifikationen von [LY94] auf das Problem $Min\ PLC_\infty$ übertragen. Dadurch gelingt es [ABSS93], eine quasipolynomial–Zeit Reduktion τ zu konstruieren, welche die große Differenz $(1 - 2^{-\log^k |\phi|}$ für ein $k \in \mathbb{N}$) der Wahrscheinlichkeiten für Akzeptieren bzw. Ver-

[3] das heißt, für festes $k \in \mathbb{N}$ in $\lceil \log$ binäre Eingabelänge $\rceil^k$–vielen

werfen des Beweises in den Fällen $\phi \in 3-SAT$ bzw. $\phi \notin 3-SAT$ in eine entsprechende Differenz ($2^{\log^{0.5-1/(k+2)}|\tau(\phi)|}$) der jeweiligen ∞–Kosten für die Eingabe von $Min\ PLC_\infty$ in beiden Fällen transformiert.

Die disjunkten Knotenmengen V_1, V_2 der Eingabe $(V_1 \cup V_2, E, \Pi, \mathcal{N})$ von $Min\ PLC_\infty$ entsprechen dabei jeweils der Menge aller Fragen an die Beweiser 1 und 2, respektive, die Marken den Antworten der Beweiser; jede Kante steht für die vom Verifizierer zufällig gewählte Kombination eines Paares von Fragen, von denen jeweils eine an Beweiser 1 und die andere an Beweiser 2 gerichtet ist. Eine Kante wird überdeckt, wenn die Antworten beider Beweiser auf ein Paar von Fragen konsistent sind und damit vom Verifizierer akzeptiert werden. Sind die Antworten der Beweiser auf alle Fragen des Verifzierers konsistent bzw. $\phi \in 3-SAT$, so gibt es für jede mögliche Frage an Beweiser 1 nur ein Antwort; in diesem Falle ist $opt_{Min\ PLC_\infty}(\tau(\phi)) = 1$. Andernfalls, das heißt, falls $\phi \notin 3-SAT$, gibt es mindestens eine Frage, die bei der zufälligen Auswahl aller Fragenpaare durch den Verifizierer mehrmals an Beweiser 1 gestellt worden ist und mehr als $2^{\log^{0.5-1/(k+2)}|\tau(\phi)|}$ verschiedene Antworten von diesem erzwingt. Unter den Fragenpaaren sind nach Konstruktion von [LY94] allerdings auch solche, für die beide Beweiser keine Antwort liefern können und der Verifizierer leere Antworten beider Beweiser akzeptieren muß. Diese Fragenpaare entsprechen unberührten Kanten. Gemäß [ABSS93] ist der Anteil unberührter Kanten vernachlässigbar klein. Gelöschte Kanten sind mindestens 2 Paare von Fragen, für die Beweiser 2 nur einmal antwortet, aber Beweiser 1 jeweils konsistente Antworten liefert und den Verifizierer nach der Konstruktion von [FL92] zum Akzeptieren aller dieser Antworten bewegt. Der Anteil gelöschter Kanten ist nach [ABSS93] entweder sehr klein oder verursacht hohe ∞–Kosten der Pseudo–Überdeckung.

4.3 Minimale $\mathbb{Z}$–Lösung homogener linearer Gleichungssysteme

Folgender Satz zeigt die Existenz einer polynomial–Zeit gap–erhaltenden Reduktion mit Parametern $((1,\rho),(1,\sqrt{\rho}))$ von $Min\ PLC_\infty$ auf $Min\ \mathbb{Z}-HLS_\infty$ für alle $\rho \geq 1$:

Satz 4.10 *Es existiert eine polynomial–Zeit Abbildung τ die jeder Eingabe I von $Min\ PLC_\infty$ ein Eingabe $\tau(I)$ von $Min\ \mathbb{Z}-HLS_\infty$ zuordnet, so daß für alle Eingaben I von $Min\ PLC_\infty$ und alle $\rho \geq 1$:*

Ist $opt_{Min\ PLC_\infty}(I) = 1$, so ist $opt_{Min\ \mathbb{Z}-HLS_\infty}(\tau(I)) = 1$.
Falls $opt_{Min\ PLC_\infty}(I) > \rho$, dann gilt $opt_{Min\ \mathbb{Z}-HLS_\infty}(\tau(I)) > \sqrt{\rho}$.

Beweis. Sei $I = (V_1 \cup V_2, E, \Pi, \mathcal{N})$ eine Eingabe von $Min\ PLC_\infty$ mit einem regulären bipartiten Graphen $G = (V_1 \cup V_2, E)$, einer Menge von Marken $\mathcal{B} = \{1, \ldots, \mathcal{N}\}$, und der Familie $\Pi := \{\Pi_e : e \in E\}$ der partiellen Funktionen aller Kanten.

Wir konstruieren aus I ein homogenes lineares System $A\,x = 0$ mit einer Matrix A, die nur aus Einträgen $-1, 0, 1$ besteht.

Sei (v, b) ein Tupel mit $v \in V_1 \cup V_2$ und $b \in \mathcal{B}(v)$, wobei $\mathcal{B}(v_2) := \mathcal{B}$ für alle $v_2 \in V_2$ und $\mathcal{B}(v_1) := \{$ gültige Marken für v_1 $\}$. Ohne Beschränkung der Allgemeinheit nehmen wir an, daß $\mathcal{B}(v_1) := \mathcal{B}$ für alle Knoten $v_1 \in V_1$ gilt, das heißt insbesondere, die Anzahl gültiger Marken $|\mathcal{B}(v_1)|$ für einen Knoten $v_1 \in V_1$ ist gleich $\mathcal{N}$. Andernfalls ist in der folgenden Darstellung der Term $|V_1|\,\mathcal{N}$ durch $\sum_{v_1 \in V_1} |\mathcal{B}(v_1|$ und der Term $|V_1|\,2^{\lceil \log \mathcal{N} \rceil}$ durch $\sum_{v_1 \in V_1} 2^{\lceil \log |\mathcal{B}(v_1)| \rceil}$ zu ersetzen.

Wir definieren die Spaltenvektoren $a_{v,b} \in \{-1, 0, 1\}^r$ von A wie folgt:

Die ersten $|E|\,(\mathcal{N}+1)$ Koordinaten von $a_{v,b}$ bestehen aus $|E|$ Blöcken von *e–Projektionen* $p_e(a_{v,b}) \in \{-1, 0, 1\}^{\mathcal{N}+1}$, für jede Kante $e \in E$ ein $\mathcal{N}+1$–dimensionaler Block. Für jedes Tupel $(v_2, b_2) \in V_2 \times \mathcal{B}(v_2)$ ist die e–Projektion $p_e(a_{v_2,b_2})$ definiert durch

$$p_e(a_{v_2,b_2}) \;:=\; \begin{cases} u_{b_2} & \text{falls } e \text{ inzident zu } v_2 \\ \vec{0} & \text{sonst} \end{cases}$$

und für jedes Tupel $(v_1, b_1) \in V_1 \times \mathcal{B}(v_1)$ durch

$$p_e(a_{v_1,b_1}) \;:=\; \begin{cases} \vec{1} - u_{\Pi_e(b_1)} & \text{falls } e \text{ inzident zu } v_1 \\ \vec{0} & \text{sonst} \end{cases}$$

wobei u_j, $j = 1, \ldots, \mathcal{N}$, jeweils den j–ten Einheitsvektor und $\vec{0}, \vec{1}$ den 0–Vektor und 1–Vektor, respektive, in $\mathbb{R}^{\mathcal{N}+1}$ darstellen.

Für die Definition der restlichen Koordinaten von $a_{v,b}$ benötigen wir Hadamard Matrizen:

Definition 4.11 *Eine* Hadamard Matrix H_ℓ *der Ordnung* ℓ *ist eine* $\ell \times \ell$ *Matrix mit Einträgen* ± 1, *für die* $H_\ell\, H_\ell^\top = \ell\, I_\ell$ *gilt, wobei* I_ℓ *die Einheitsmatrix in* $\mathbb{R}^\ell$ *ist.*

Für ein $\ell \in \mathbb{N}$ stellen die Spalten von $\frac{1}{\sqrt{\ell}} H_\ell$ nach Definition eine Orthonormalbasis in $\mathbb{R}^\ell$ dar. Für jeden Vektor $z \in \mathbb{Z}^\ell$ gilt daher

$$\|\frac{1}{\sqrt{\ell}} H_\ell\, z\|_2 \;=\; \|z\|_2\,.$$

Falls $z \in \mathbb{Z}^\ell$ mindestens k nicht verschwindende Einträge hat, gilt damit

$$\|H_\ell\, z\|_\infty \;\geq\; \sqrt{k}\,. \tag{4.3}$$

Sei nun $H_\mathcal{N} = [h_1, \ldots, h_\mathcal{N}]$ die Hadamard Matrix mit Spaltenvektoren h_b, die jeweils eindeutig einer Marke $b \in \mathcal{B}$ zugeordnet sind. Hadamard Matrizen H_ℓ können in linearer Zeit in der Ausgabelänge konstruiert werden, wenn ℓ eine 2er–Potenz ist [WS77] (Seite 45). Falls $\mathcal{N}$ keine 2er Potenz ist, definieren wir $H_\mathcal{N}$ als die Matrix, die aus den ersten $\mathcal{N}$ Spalten der Hadamard Matrix H_ℓ mit $\ell = 2^{\lceil \log \mathcal{N} \rceil}$ besteht. Für die so konstruierte Matrix $H_\mathcal{N}$ bleibt Eigenschaft (4.3) für alle Vektoren $z \in \mathbb{Z}^\mathcal{N}$ erhalten.

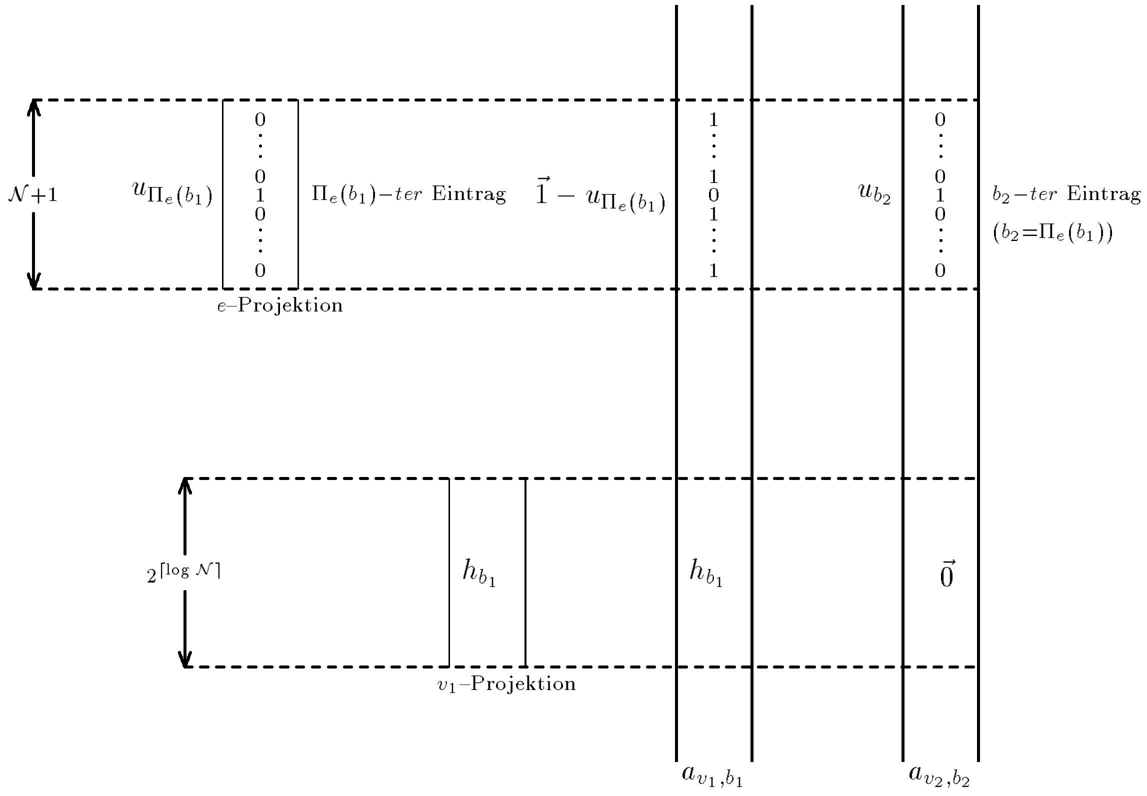


Abbildung 4.1: Spaltenvektoren von A gemäß [ABSS93]

Die letzten $|V_1|\,2^{\lceil \log \mathcal{N} \rceil}$ Koordinaten von $a_{v,b}$ werden nun in $|V_1|$ Blöcke von v_1–*Projektionen* $p_{v_1}(a_{v,b}) \in \{\pm 1\}^{2^{\lceil \log \mathcal{N} \rceil}}$ aufgeteilt, für jeden Knoten $v_1 \in V_1$ ein $2^{\lceil \log \mathcal{N} \rceil}$–dimensionaler Block. Für jedes Tupel $(v,b) \in (V_1 \cup V_2) \times \mathcal{B}$ wird die v_1–Projektion $p_{v_1}(a_{v,b})$ definiert durch

$$p_{v_1}(a_{v,b}) \quad := \quad \begin{cases} h_b & \text{falls } v = v_1 \\ \vec{0} & \text{sonst} \end{cases} \quad ,$$

wobei $\vec{0}$ der 0–Vektor in $\mathbb{R}^{2^{\lceil \log \mathcal{N} \rceil}}$ ist. Die Definition impliziert insbesondere $p_{v_1}(a_{v,b}) = \vec{0}$ für alle $v \in V_2$ und alle $b \in \mathcal{B}$.

Abbildung 4.1 veranschaulicht die Konstruktion der Vektoren $a_{v,b}$ mit $(v,b) \in (V_1 \cup V_2) \times \mathcal{B}$.

Wir definieren den $(|V_1|\,\mathcal{N} + |V_2|\,\mathcal{N} + 1)$–ten Spaltenvektor a_0 von A durch

$$a_0 \;=\; 1^{|E|\,(\mathcal{N}+1)} \times 0^{|V_1|\,2^{\lceil \log \mathcal{N} \rceil}} \;.$$

Die letzten $|V_1|\,2^{\lceil \log \mathcal{N} \rceil}$ Spaltenvektoren von A sind schließlich die Einheitsvektoren $u_{|E|\,(\mathcal{N}+1)+i}$, $i = 1, \ldots, |V_1|\,2^{\lceil \log \mathcal{N} \rceil}$ in $\mathbb{R}^{|E|\,(\mathcal{N}+1)+|V_1|\,2^{\lceil \log \mathcal{N} \rceil}}$. Die so konstruierte Matrix A ist in Abbildung 4.2 wiedergegeben. $I_{|V_1|\,2^{\lceil \log \mathcal{N} \rceil}}$ bezeichnet dabei die Einheitsmatrix in $GL_{|V_1|\,2^{\lceil \log \mathcal{N} \rceil}}(\mathbb{Z})$.

$$
\begin{array}{r c}
 & \begin{array}{cccc} |V_1|\,\mathcal{N}\ \ \textit{Spalten} & |V_2|\,\mathcal{N}\ \ \textit{Spalten} & 1\ \ \textit{Spalte} & |V_1|\,2^{\lceil \log \mathcal{N} \rceil}\ \ \textit{Spalten} \end{array} \\
\begin{array}{r} |E|\,(\mathcal{N}{+}1)\ \ \textit{Zeilen} \\ \\ \\ |V_1|\,2^{\lceil \log \mathcal{N} \rceil}\ \ \textit{Zeilen} \end{array} &
\left(\begin{array}{cccc}
[p_e(a_{v,b_1}), \ldots, p_e(a_{v,b_{\mathcal{N}}})]_{\substack{e \in E \\ v \in V_1}} & [p_e(a_{v,b})]_{\substack{e \in E \\ (v,b) \in V_2 \times \mathcal{B}}} & \vec{1} & \vec{0} \\
 & & & \\
 & & & \\
[p_{v_1}(a_{v,b_1}), \ldots, p_{v_1}(a_{v,b_{\mathcal{N}}})]_{\substack{v_1 \in V_1 \\ v \in V_1}} & \vec{0} & \vec{0} & I_{|V_1|\,2^{\lceil \log \mathcal{N} \rceil}}
\end{array} \right)
\end{array}
$$

Abbildung 4.2: Matrix A

Der Übersichtlichkeit wegen haben wir die Abkürzung

$$[p_e(a_{v,b_1}), \ldots, p_e(a_{v,b_{\mathcal{N}}})]_{\substack{e \in E \\ v \in V_2}} \quad =: \quad [p_e(a_{v,b})]_{\substack{e \in E \\ (v,b) \in V_2 \times \mathcal{B}}}$$

verwendet.

Für einen Vektor $y \in \mathbb{R}^{|E|\,(\mathcal{N}+1)+|V_1|\,2^{\lceil \log \mathcal{N} \rceil}}$ definieren wir $p_E(y) \in \mathbb{R}^{|E|\,(\mathcal{N}+1)}$ als die *Einschränkung* von y auf seine ersten $|E|\,(\mathcal{N}+1)$ Koordinaten.

Sei $x = \sum x_{v,b}\, p_E(a_{v,b})$ eine nicht triviale ganzzahlige Linearkombination der Spaltenvektoren $p_E(a_{v,b})$. Weist man jedem Knoten v, für den $x_{v,b} \neq 0$ ist, eine Marke $b \in \mathcal{B}$ zu, so wird dadurch eine Markierung $(\mathcal{P}_1^x, \mathcal{P}_2^x)$ definiert, die durch den Vektor x *induzierte Markierung.*

Falls die Kante $e = (v_1, v_2)$ von einem Tupel (b_1, b_2) von Marken überdeckt wird, gilt offensichtlich für die e–Projektion von $a_{v_1,b_1} + a_{v_2,b_2}$ wegen $\Pi_e(b_1) = b_2$:

$$p_e(a_{v_1,b_1}) + p_e(a_{v_2,b_2}) \;=\; 1^{\mathcal{N}+1} - u_{\Pi_e(b_1)} + u_{b_2} \;=\; 1^{\mathcal{N}+1} \;.$$

Für eine Totalüberdeckung von G existiert somit ein nicht trivialer Vektor $x \in \{0,1\}^{|E|(\mathcal{N}+1)}$ mit:

$$\sum_{(v,b) \in (V_1 \cup V_2) \times \mathcal{B}} x_{v,b}\, p_E(a_{v,b}) \;=\; 1^{|E|\,(\mathcal{N}+1)} \;.$$

Arora, Babai, Stern, Sweedyk haben folgende Quasi–Umkehrung gezeigt:

Lemma 4.12 *[ABSS93], Korollar 11*
Falls $x = \sum x_{v,b}\, p_E(a_{v,b})$ eine nicht triviale ganzzahlige Linearkombination der Spaltenvektoren $p_E(a_{v,b})$ ist und $x = \alpha\, 1^{|E|\,(\mathcal{N}+1)}$ für ein $\alpha \in \mathbb{Z}$ gilt, dann ist die von x induzierte Markierung $(\mathcal{P}_1^x, \mathcal{P}_2^x)$ eine Pseudo–Überdeckung.

Falls $\alpha \neq 0$, dann ist $(\mathcal{P}_1^x, \mathcal{P}_2^x)$ eine Totalüberdeckung von $G = (V_1 \cup V_2, E)$.

Beweis. Nach Annahme gilt

$$x \;=\; \sum_{x_{v,b} \in \mathbb{Z} - 0} x_{v,b}\, p_E(a_{v,b}) \;=\; \alpha\, 1^{|E|\,(\mathcal{N}+1)} \;. \tag{4.4}$$

Für jede Kante $e \in E$ sind die e–Projektionen $p_e(a_{v,b})$ mit $x_{v,b} \neq 0$ entweder u_b, $1^{\mathcal{N}+1} - u_{\Pi_e(b)}$ für eine Marke $b \in \mathcal{B}$ oder $0^{\mathcal{N}+1}$. Die Vektoren $u_1, \ldots, u_{\mathcal{N}}$ sind linear unabhängig und erzeugen nicht $span(1^{\mathcal{N}+1})$. Somit gilt im Falle

$$\sum_{b_2 \in \mathcal{B}} x_{b_2}\, u_{b_2} + \sum_{b_2 \in \mathcal{B}} y_{b_2}\, (1^{\mathcal{N}+1} - u_{b_2}) \;=\; \alpha\, 1^{\mathcal{N}+1} \;,$$

für alle $b_2 \in \mathcal{B}$: $x_{b_2} = y_{b_2}$.

Falls $b_2 \in \mathcal{B}$ und der Koeffizient von u_{b_2} nicht 0 ist, muß daher ein Vektor $1^{\mathcal{N}+1} - u_{b_2}$ mit nicht trivialem Koeffizienten in der Darstellung (4.4) von x als ganzzahlige Linearkombination von Vektoren $p_E(a_{v,b})$ existieren, das heißt, es gibt Knoten v_1, v_2, jeweils inzident zu e und eine Marke $b_1 \in \mathcal{B}$ mit $\Pi_e(b_1) = b_2$. Also wird e überdeckt.

Falls der Koeffizient von u_{b_2} gleich 0 ist, dann muß die Anzahl der Vektoren $1^{\mathcal{N}+1} - u_{b_2}$ mit nicht verschwindendem Koeffizienten in der Darstellung (4.4) von x entweder 0 oder mindestens 2 sein. Im ersten Falle ist die betreffende Kante von der Markierung $(\mathcal{P}_1^x, \mathcal{P}_2^x)$ unberührt, im letzteren Fall gelöscht.

Damit ist die durch x induzierte Markierung $(\mathcal{P}_1^x, \mathcal{P}_2^x)$ in jedem Falle eine Pseudo–Überdeckung. Falls $\alpha \neq 0$ gilt, so ist für jede Kante $e \in E$ der Koeffizient der e–Projektion $p_e(a_{v,b_2})$ in der Darstellung der jeweiligen e–Projektion von x als ganzzahlige Linearkombination der $p_e(a_{v,b})$ für eine Marke $b_2 \in \mathcal{B}$ ungleich 0. Da in diesem Falle die Kante e überdeckt wird, erhalten wir für $\alpha \neq 0$ durch $(\mathcal{P}_1^x, \mathcal{P}_2^x)$ sogar eine Totalüberdeckung. □

Aus obigem Lemma folgt nach Konstruktion der ersten $|V_1|\,\mathcal{N} + |V_2|\,\mathcal{N}$ Zeilen von A, daß jede nicht triviale ganzzahlige Lösung $x \in \mathbb{Z}^{|V_1|\,\mathcal{N}+|V_2|\,\mathcal{N}+1+|V_1|\,2^{\lceil \log \mathcal{N} \rceil}}$ des homogenen linearen Systems $A\,x = 0$ durch die ersten $|V_1|\,\mathcal{N} + |V_2|\,\mathcal{N}$ Koordinaten eine Pseudo–Überdeckung $(\mathcal{P}_1^x, \mathcal{P}_2^x)$ des Graphen $G = (V_1 \cup V_2, E)$ induziert. (Die Einschränkung $p_E(a_0)$ des $(|V_1|\,\mathcal{N} + |V_2|\,\mathcal{N} + 1)$–ten Spaltenvektors von A ist genau der 1–Vektor in $\mathbb{R}^{|V_1|\,\mathcal{N}+|V_2|\,\mathcal{N}}$, und die Einschränkung $p_E(a_j)$ der letzten $|V_1|\,2^{\lceil \log \mathcal{N} \rceil}$ Spaltenvektoren a_j, $j = |V_1|\,\mathcal{N} + |V_2|\,\mathcal{N} + 2, \ldots, |V_1|\,\mathcal{N} + |V_2|\,\mathcal{N} + 1 + |V_1|\,2^{\lceil \log \mathcal{N} \rceil}$ ist jeweils genau der 0–Vektor in $\mathbb{R}^{|V_1|\,\mathcal{N}+|V_2|\,\mathcal{N}}$).

Für die somit induzierte Pseudo–Überdeckung $({\mathcal{P}_1}^x, {\mathcal{P}_2}^x)$ existiert ein Knoten $v_1 \in V_1$ mit mindestens $opt_{Min\ PLC_\infty}(I)$ zugewiesenen Marken. Dies bedeutet jedoch umgekehrt, daß x mindestens $opt_{Min\ PLC_\infty}(I)$ nicht verschwindende Einträge hat. Mit Eigenschaft (4.3) von Hadamard Matrizen muß es daher einen Index $i^* \in \{|E|\,(\mathcal{N}+1)+1, \ldots, |E|\,(\mathcal{N}+1) + |V_1|\,2^{\lceil \log \mathcal{N} \rceil}\}$ geben mit

$$\left| \sum_{j=1}^{|V_1|\,\mathcal{N}} a_{i^*,j} x_j \right| \geq \sqrt{opt_{Min\ PLC_\infty}(I)}\;.$$

Da x eine Lösung von $A\,x = 0$ ist, muß nach Konstruktion von A für die letzten $|V_1|\,2^{\lceil \log \mathcal{N} \rceil}$ Koordinaten $x_{(|V_1|+|V_2|)\,\mathcal{N}+1+i}$, $i = 1, \ldots, |V_1|\,2^{\lceil \log \mathcal{N} \rceil}$ von x gelten, daß

$$x_{(|V_1|+|V_2|)\,\mathcal{N}+1+i} \;=\; -\sum_{j=1}^{|V_1|\,\mathcal{N}} a_{|E|\,(\mathcal{N}+1)+i,j}\, x_j\;, \quad i = 1, \ldots, |V_1|\,2^{\lceil \log \mathcal{N} \rceil}\;.$$

Somit hat jede nicht triviale ganzzahlige Lösung x von $A\,x \;=\; 0$ mindestens eine Koordinate $x_{|V_1|\,\mathcal{N}+|V_2|\,\mathcal{N}+1+i^*}$, $i^* \in \{1, \ldots, |V_1|\,2^{\lceil \log \mathcal{N} \rceil}\}$ mit

$$\|x\|_\infty \;\geq\; |x_{|V_1|\,\mathcal{N}+|V_2|\,\mathcal{N}+1+i^*}| \;\geq\; \sqrt{opt_{Min\ PLC_\infty}(I)}\ .$$

Sei nun $opt_{Min\ PLC_\infty}(I) = 1$ und $(\mathcal{P}_1, \mathcal{P}_2)$ die entsprechende Pseudo–Überdeckung, für die $opt_{Min\ PLC_\infty}(I) = 1$ angenommen wird. Dann stellt offensichtlich der $(|V_1|\,\mathcal{N} + |V_2|\,\mathcal{N} + 1 + |V_1|\,2^{\lceil \log \mathcal{N} \rceil})$–dimensionale Vektor $x = (x_1, \ldots, x_{|V_1|\,\mathcal{N}+|V_2|\,\mathcal{N}+1+|V_1|\,2^{\lceil \log \mathcal{N} \rceil}})$ mit

$$\begin{array}{rcll} x_{v_i, \mathcal{P}_i(v_i)} & := & 1 & \forall v_i \in V_i,\ i = 1, 2 \\ x_{v_i, b} & := & 0 & \forall v_i \in V_i,\ \forall b \in \mathcal{B} \setminus \mathcal{P}_i(v_i),\ i = 1, 2 \\ x_{|V_1|\mathcal{N}+|V_2|\mathcal{N}+1} & := & -1 & \\ x_{|V_1|\mathcal{N}+|V_2|\mathcal{N}+1+i} & := & -x_i & i = 1, \ldots, |V_1|\,2^{\lceil \log \mathcal{N} \rceil} \end{array}$$

eine Lösung des homogenen linearen Systems $A\,x \;=\; 0$ mit $\|x\|_\infty = 1$ dar.

Die Konstruktion der Matrix A aus der Eingabe $I \in Min\ PLC_\infty$ ist in polynomial–Zeit in der Anzahl der Spalten und Zeilen von A durchführbar. Die Anzahl der Spalten und Zeilen von A ist jeweils polynomial in der Eingabelänge der Eingabe I. Damit ist die Behauptung vollständig gezeigt. □

Korollar 4.13 *Falls* $\boldsymbol{NP} \not\subseteq \boldsymbol{QP}$, *dann existiert kein polynomial–Zeit Algorithmus, der das Problem* $SSIR_\infty$ *bis zu einem Faktor* $2^{\log^{0.5-\zeta} N}$ *approximiert, wobei* ζ *eine beliebig kleine positive Konstante und* N *die binäre Länge der Eingabe* I *von* SIR_∞ *ist.*

Die Aussage des Korollars bedeutet umgekehrt folgendes: Jeder Approximationsalgorithmus, der für Eingaben I von $SSIR_\infty$ den minimalen Wert $opt_{SSIR_\infty}(I)$ bis auf einen Faktor $2^{\log^{0.5-\zeta}}$ für ein beliebig kleines positives ζ approximiert, kann in einen Algorithmus transformiert werden, der $3-SAT$ in quasipolynomialer Zeit entscheidet. Die Approximation von $SSIR_\infty$ ist in diesem Sinne bis auf den Faktor $2^{\log^{0.5-\zeta} N}$ fast–**NP**–hart, wobei ζ eine beliebig kleine positive Konstante und N die binäre Länge der Eingabe von $SSIR_\infty$ ist.

4.4 Aggregation

Folgendes Lemma, implizit von Kannan [Ka83] bewiesen, gibt eine polynomial–Zeit Bijektion des Anteils der in der ∞–Norm beschränkten Lösungen eines homogenen linearen Gleichungssystems $A\,x \;=\; 0$ auf den Anteil gleichsam in der ∞–Norm beschränkter Lösungen einer einzelnen homogenen linearen Gleichung $< a, x > \;=\; 0$ an.

Lemma 4.14 *Sei* $A := (a_{i,j})_{\substack{1 \leq i \leq r \\ 1 \leq j \leq n}} \in M(r, n, \mathbb{Z})$, $\|A\|_\infty := \max_{\substack{1 \leq i \leq r \\ 1 \leq j \leq n}} |a_{i,j}|$ *und* $B_\mu := \{x \in \mathbb{R}^n \,:\, \|x\|_\infty \leq \mu\}$ *die* n*–dimensionale Kugel um den Nullpunkt mit* ∞*–Radius* μ *sowie* $k = n\,\|A\|_\infty\,\mu + 1$. *Dann ist*

$$B_\mu \cap \{x \in \mathbb{Z}^n \mid A\,x \;=\; 0\} \;=\; B_\mu \cap \left\{ x \in \mathbb{Z}^n \;\middle|\; \sum_{i=1}^{r} k^i \sum_{j=1}^{n} a_{i,j}\,x_j \;=\; 0 \right\}.$$

Beweis. Wir bezeichnen linke bzw. rechte Seite der obigen Gleichung mit S_r und S_1, respektive. Offensichtlich ist $S_r \subseteq S_1$.

Um die umgekehrte Inklusion zu zeigen, nehmen wir an, daß ein Element $x \in S_1$ existiert, welches mindestens eine der Gleichungen von $A\,x = 0$ nicht erfüllt. Seien $a_1, \ldots, a_r$ die Zeilenvektoren von A und i_{max} der größte Index, für den $< a_i, x > \neq 0$ gilt. Aus $\|x\|_\infty \leq \mu$ folgt

$$|< a_i, x >| \quad \leq \quad n\,\|A\|_\infty\,\mu \;=\; k-1\;. \tag{4.5}$$

Da $x \in S_1$ muß $\sum_{i=1}^{r} k^i \;< a_i, x > = \;0$ sein. Nach Definition von i_{max} erhalten wir

$$\sum_{i=1}^{i_{max}-1} k^i \;< a_i, x > \quad = \quad -k^{i_{max}} \;< a_{i_{max}}, x > \neq \;0\;.$$

Die linke Seite dieser Gleichung ist damit zum einen ein Vielfaches von $k^{i_{max}}$, zum anderen wegen (4.5) absolut beschränkt durch

$$(k-1)\sum_{i=1}^{i_{max}-1} k^i \;=\; (k-1)\,\frac{k^{i_{max}}-k}{k-1} \;=\; k^{i_{max}}-k \;\leq\; k^{i_{max}}-1\;;$$

dies ist ein Widerspruch und beweist damit die behauptete Inklusion. □

Wir zeigen nun das Hauptresultat dieses Kapitels:

Satz 4.15 *Falls* $\boldsymbol{NP} \not\subseteq \boldsymbol{QP}$, *dann existiert kein polynomial–Zeit Algorithmus, der das Problem* SIR_∞ *bis zu einem Faktor* $2^{\log^{0.5-\zeta} bin(x)}$ *approximiert, wobei* ζ *eine beliebig kleine positive Konstante und* $bin(x)$ *die binäre Länge der Eingabe* x *von* SIR_∞ *ist.*

Die Approximation von SIR_∞ ist in diesem Sinne bis auf einen Faktor $2^{\log^{0.5-\zeta} bin(x)}$ fast–**NP**–hart, wobei ζ eine beliebig kleine positive Konstante und $bin(x)$ die binäre Länge der Eingabe x von SIR_∞ ist.

Beweis. Nach Satz 4.10 können wir annehmen, daß eine Eingabe $I_1 = (V_1, V_2, E, \Pi, \mathcal{B}, \mathcal{N})$ von $Min\ PLC_\infty$ durch eine gap–erhaltende Reduktion τ mit Parametern $((1,\rho),(1,\sqrt{\rho}))$ in eine Eingabe I_2 von $Min\ \mathbb{Z}-HLS_\infty$ abgebildet wird, welche aus einer Matrix

$$A \;\in\; M(|E|\,(\mathcal{N}+1)+|V_1|\,2^{\lceil\log\mathcal{N}\rceil}\,,\;|V_1|\,\mathcal{N}+|V_2|\,\mathcal{N}+1+|V_1|\,2^{\lceil\log\mathcal{N}\rceil})\,,\;\{-1,0,1\})$$

besteht.

Für den Beweis des Satzes genügt es daher, eine gap–erhaltende Reduktion σ mit Parametern $((1,\rho),(1,\rho))$ von $Min\ \mathbb{Z}-HLS_\infty$ auf SIR_∞ anzugeben. Wir fixieren hierzu $\rho \geq 1$ und wenden Lemma 4.14 auf die Matrix A und $\mu = \sqrt{\rho}$ sowie $k = (|V_1|\,(\mathcal{N}+2^{\lceil\log\mathcal{N}\rceil}) + |V_2|\,\mathcal{N}+1)\,\sqrt{\rho}+1$ an. (Nach Konstruktion von A ist $\|A\|_\infty = 1$.) Wir erhalten eine Eingabe $I_3 := \sigma(I_2)$ von SIR_∞, die aus der Gleichung

$$\sum_{i=1}^{r}\sum_{j=1}^{n} k^i\,a_{i,j}\,x_j \quad = \quad 0 \tag{4.6}$$

besteht, wobei $r = |E|\,(\mathcal{N}+1) + |V_1|\,2^{\lceil \log \mathcal{N} \rceil}$ und $n = |V_1|\,\mathcal{N} + |V_2|\,\mathcal{N} + 1 + |V_1|\,2^{\lceil \log \mathcal{N} \rceil}$. Die binäre Länge von I_3 ist polynomial in $|I_2|$, da $a_{i,j} = 1$ für alle $1 \leq i \leq r$, $1 \leq j \leq n$ und $\lceil \log k^r \rceil = O(r\,|I_2|)$.

Sei nun $opt_{Min\ \mathbb{Z}-HLS_\infty}(I_2) > \sqrt{\rho}$. Nach Lemma 4.14 ist eine Lösung x von (4.6) mit $\|x\|_\infty \leq \sqrt{\rho}$ ebenfalls eine Lösung von $A\,x = 0$, im Widerspruch zu Satz 4.10. Somit muß für jede Lösung x von (4.6) gelten, daß

$$opt_{SIR_\infty}(I_3) > \sqrt{\rho}\,.$$

Falls $opt_{Min\ \mathbb{Z}-HLS_\infty}(I_2) = 1$, so ist nach Lemma 4.14 jede optimale Lösung für I_2 Zeuge für $opt_{SIR_\infty}(I_3) = 1$.

Mit Lemma 4.9 erhalten wir daher eine quasipolynomial–Zeit Reduktion $v := \sigma \circ \tau$, so daß für alle Eingaben I und alle $\zeta > 0$:

$$\begin{aligned} I \in 3-SAT &\implies opt_{SIR_\infty}(v(I)) = 1 \\ I \notin 3-SAT &\implies opt_{SIR_\infty}(v(I)) > \sqrt{2^{\log^{0.5-\zeta}|v(I)|}}\,. \end{aligned}$$

Ein polynomial–Zeit Algorithmus, der SIR_∞ bis auf einen Faktor $2^{\log^{0.5-\zeta}}$ für ein beliebiges $\zeta > 0$ approximiert, kann daher $3-SAT$ in quasipolynomialer Zeit entscheiden. □

Bemerkung. *Wir haben gezeigt, daß unter der Annahme* $\boldsymbol{NP} \not\subseteq \boldsymbol{QP}$ *kein polynomial–Zeit Algorithmus existiert, der* SIR_∞ *bis auf einen Faktor* $2^{\log^{0.5-\zeta} N}$ *für beliebig kleines* $\zeta > 0$ *approximiert, wobei* N *die binäre Länge der jeweiligen Eingabe von* SIR_∞ *angibt.*

Die Verbesserung des Nichtapproximierbarkeitsfaktors $2^{\log^{0.5-\zeta} N}$ *auf* N^ζ *für ein beliebiges* $\zeta > 0$ *ist ein immer noch offenes Problem. Ebenso offen ist, ob in Satz 4.15 der Nichtapproximierbarkeit* $2^{\log^{0.5-\zeta} N}$ *auch unter der Annahme* $\boldsymbol{NP} \neq \boldsymbol{P}$ *statt* $\boldsymbol{NP} \not\subseteq \boldsymbol{QP}$ *gilt.*

Kapitel 5

Approximierbarkeit minimaler diophantischer Approximationen

In Kapitel 3 haben wir einen Algorithmus angegeben, der für reelle Eingaben $x_1, \ldots, x_{n-1}$, $\epsilon > 0$ bis auf den Faktor $2^{(n+2)/4} \max_{1 \le i \le n-1} \sqrt{1 + x_i^2}$ im Sinne der Existenzaussage von Dirichlet optimale simultane diophantische Approximationen berechnet.

In diesem Kapitel untersuchen wir die Approximierbarkeit eines rationalen Vektors x durch Bestapproximationen mit beschränktem Hauptnenner, das heißt durch ganzzahlige Vektoren $(p_1, \ldots, p_{n-1}, q)$ mit minimalem $\max_{1 \le i \le n-1} |q\, x_i - p_i|$, wobei das Minimum über alle $q \in [1, N]$ für eine vorgegebene natürliche Zahl $N \in \mathbb{N}$ betrachtet wird. Mit Hilfe der in Kapitel 4 bewiesenen Sätze zeigen wir folgendes Resultat:

Falls $\mathbf{NP} \not\subseteq \mathbf{QP}$, dann existiert kein polynomial–Zeit Algorithmus, der für Eingaben $y = (y_1, \ldots, y_n)^\top \in \mathbf{Q}^n$, $\epsilon \in \mathbf{Q}_+$ und $N \in \mathbb{N}$ ganze Zahlen $p_1, \ldots, p_n, q^*$ berechnet, so daß $1 \le q^* \le 2^{\log^{0.5-\zeta} bin(y)} N$ und

$$\max_{1 \le i \le n} |q^* y_i - p_i| \;\le\; 2^{\log^{0.5-\zeta} bin(y)} \min_{1 \le q \le N} \|q\, y \bmod \mathbb{Z}\|_\infty ,$$

wobei ζ eine beliebig kleine positive Konstante ist und $\|(y_1, \ldots, y_n)^\top \bmod \mathbb{Z}\|_\infty = \max_{1 \le i \le n} \min_{m \in \mathbb{Z}} |y_i - m|$ bezeichnet.

Lagarias [Lag85] betrachtete das Problem zu entscheiden, ob zu gegebenen $y = (y_1, \ldots, y_n)^\top \in \mathbf{Q}^n$, $\epsilon \in \mathbf{Q}_+$, $N \in \mathbb{N}$ gute simultane diophantische Approximationen existieren, das heißt, ob es ganze Zahlen $p_1, \ldots, p_n, q^*$ gibt, so daß $q^* \in [1, N]$ und $\max_{1 \le i \le n} |q^* y_i - p_i| \le \epsilon$. Lagarias bewies die **NP**–Vollständigigkeit dieses Problems. Das hierzu kanonische Minimierungsproblem besteht darin, minimale simultane diophantische Approximationen zu finden, das heißt, ganze Zahlen $p_1, \ldots, p_n \in \mathbb{Z}$ $q^* \in [1, N]$ zu finden, die $\max_{1 \le i \le n} |q^* y_i - p_i|$ minimieren. Lagarias zeigte in [Lag85] die Existenz eines polynomial–Zeit Algorithmus, der für Eingaben $y = (y_1, \ldots, y_n)^\top \in \mathbf{Q}^n$ und $N \in \mathbb{N}$ ganze Zahlen $p_1, \ldots, p_n$ und $q^* \in [1, 2^{n/2} N]$ berechnet, so daß

$$\max_{1 \le i \le n} |q^* y_i - p_i| \;\le\; \sqrt{5\, n}\, 2^{(n-1)/2} \min_{1 \le q \le N} \|q\, y \bmod \mathbb{Z}\|_\infty .$$

Lagarias stellte hierzu folgende Vermutung auf: Unter der Annahme $\mathbf{P} \neq \mathbf{NP}$ gibt es keinen Algorithmus, der für ein Polynom $p(n)$ in n auf Eingaben $y = (y_1, \ldots, y_n)^\top \in \mathbf{Q}^n$ und $N \in \mathbb{N}$ ganze Zahlen $p_1, \ldots, p_n$ und $q^* \in [1, p(n)\, N]$ berechnet mit

$$\max_{1 \leq i \leq n} |q^* y_i - p_i| \;\leq\; p(n) \min_{1 \leq q \leq N} \|q\, y \bmod \mathbb{Z}\|_\infty \ ,$$

und polynomial viele Bitoperationen in der Eingabelänge unabhängig von der Dimension n durchführt; das heißt, die Approximation (des Nenners) der besten simultanen diophantischen Approximation $(p_1, \ldots, p_n, q)$ mit $q \in [1, N]$ ist bis auf einen Faktor $p(n)$ $\mathbf{NP}$–hart.

Das Resultat diese Kapitels ist daher ein Schritt dahin, die Lücke zwischen oberen und unteren Schranken für die Approximierbarkeit des Nenners der besten simultanen diophantischen Approximation zu schließen.

Wir führen im 1. Paragraphen das Minimierungsproblem Minimale Diophantische Approximation in der ∞–Norm $Min\ DA_\infty$ ein. Danach geben wir eine gap–erhaltende polynomial–Zeit Reduktion von dem in Kapitel 4 behandelten Problem SIR_∞ auf $Min\ DA_\infty$ an und zeigen damit das Hauptresultat dieses Kapitels. Die gap–erhaltende Reduktion verwendet 2 Lemmata aus der Analytischen Zahlentheorie, welche in Paragraph 2 separat bewiesen werden.

Der Inhalt diese Kapitels entspricht im wesentlichen dem der Arbeit [RSe96b].

5.1 Minimale Simultane Diophantinische Approximationen

Wir benutzen die gleiche Notation wie in Kapitel 4. Das Problem *Minimale Simultane Diophantische Approximation in* ∞*–Norm* $Min\ DA_\infty$ ist wie folgt definiert:

$\underline{Min\ DA_\infty\ :}$
GEGEBEN ein rationaler Vektor $y = (y_1, \ldots, y_n)^\top \in \mathbf{Q}^n$ und eine natürliche Zahl $N \in \mathbb{N}$.
FINDE eine natürliche Zahl $q \in [1, N]$, den *Nenner der diophantischen Approximation,* mit minimalem $\|q\, y \bmod \mathbb{Z}\|_\infty := \max_{1 \leq i \leq n} \min_{m \in \mathbb{Z}} |q\, y_i - m|$.

Satz 5.1 *Das Problem* $Min\ DA_\infty$ *ist* ***NP****–hart.*

Beweis. [Lag85], Theorem E, Seiten 207–208. □

Im restlichen Teil dieses Abschnitts zeigen wir, daß für eine beliebig kleine positive Konstante ζ die Approximierbarkeit von $Min\ DA_\infty$ bis auf einen Faktor $2^{\log^{0.5-\zeta} bin(y)}$ fast–$\mathbf{NP}$–hart ist, wobei $bin(y)$ die binäre Länge der Eingabe y von $Min\ DA_\infty$ ist. Wir modifizieren hierzu den von Lagarias in [Lag85] angegebenen Beweis der $\mathbf{NP}$–Vollständigkeit des Entscheidungsproblems *Gute Simultane Diophantische Approximation.*

Satz 5.2 *Es existiert eine polynomial–Zeit Abbildung* τ, *die Eingaben* I *von* SIR_∞ *auf Eingaben* $\tau(I) = (y := (a_0/b_0, \ldots, a_n/b_n)^\top,\ N)$ *abbildet, so daß für alle Eingaben* I *und alle* $\rho \in [1, |I|]$:

Ist $opt_{SIR_\infty}(I) = 1$, *so ist* $\min_{1 \le q \le N} \|q\, y \bmod \mathbb{Z}\|_\infty \le \frac{1}{b_1}$.
Falls $opt_{SIR_\infty}(I) > \rho$, *dann folgt* $\min_{1 \le q \le \rho N} \|q\, y \bmod \mathbb{Z}\|_\infty > \rho \frac{1}{b_1}$.

Beweis. Sei $I := (x_1, \ldots, x_n)^\top \in \mathbb{Z}^n - \{0\}$ die Eingabe von SIR_∞ und $\Lambda := n\, bin(\|x\|_\infty) \ge |I|$ sowie $\rho \in [1, \Lambda]$.

Wir folgen dem Beweis von Lagarias [Lag85] und führen in 3 Schritten eine gap–erhaltende polynomial–Zeit Reduktion τ von I auf eine Eingabe $\tau(I)$ von $Min\ DA_\infty$ durch:

Zuerst reduzieren wir das Problem, ein $m \in \mathbb{Z}^n - \{0\}$ mit $< m, x > = 0$ und $\|m\|_\infty \le \rho$ zu finden, auf das Problem eine in der ∞–Norm durch ρ beschränkte Lösung einer Kongruenz modulo einer Primzahlpotenz zu finden.

Seien hierzu $P := (p_k)_{k \in \mathbb{N}}$ die Folge der Primzahlen und p_{s_0} die kleinste Primzahl mit $p_{s_0} \not| \prod_{i=1}^n x_i$.

Seien weiter $X := \rho \sum_{i=1}^n |x_i|$ und $R := \lfloor \log_{p_{s_0}} X \rfloor + 1$.

Wir machen bei unserer Reduktion von den folgenden 2 Lemmata Gebrauch, deren Beweis auf Abschnitt 5.2 verschoben wird:

Lemma 5.3 *Es existiert ein* $s \in \mathbb{N}$, *so daß für eine Teilmenge* $\{p_{s_k},\ 1 \le k \le n\} \subseteq P \cap (p_s, \frac{\rho+1}{\rho} p_s)$ *die Zahlen* $R_i := \prod_{j=1}^s p_j + p_{s_i}$, $i = 1, \ldots, n$, *paarweise teilerfremd sind und folgende Bedingungen erfüllen:*

$$R_i < R_{i+1},\ i = 1, \ldots, n-1, \tag{5.1}$$

$$R_n < \frac{\rho+1}{\rho} R_1, \tag{5.2}$$

$$ggT(R_i, p_{s_0} \prod_{j=1}^n x_j) = 1,\ i = 1, \ldots, n, \tag{5.3}$$

$$\prod_{j=1}^{s-1} p_j > 2\rho\, p_{s_0}^R (n+1). \tag{5.4}$$

Die Zahlen R_i, $i = 1, \ldots, n$ *sind in polynomial vielen Bitoperationen in* Λ *konstruierbar.*

Lemma 5.4 *Seien die Bezeichnungen wie in Lemma 5.3 und* $R_1, \ldots, R_n$ *die gemäß Lemma 5.3 konstruierten paarweise teilerfremden Zahlen. Dann existieren ganze Zahlen* $r_1, \ldots, r_n$, *so daß für* $i = 1, \ldots, n$:

$$r_i \equiv 0 \bmod \prod_{\substack{j=1 \\ j \ne i}}^n R_j, \tag{5.5}$$

$$r_i \equiv x_i \bmod p_{s_0}^R, \tag{5.6}$$

$$r_i \not\equiv 0 \bmod R_i, \tag{5.7}$$

$$|r_i| < \frac{1}{2\rho(n+1)} \prod_{j=1}^n R_j. \tag{5.8}$$

Die Zahlen r_i, $i = 1, \ldots, n$ *sind in polynomial vielen Bitoperationen in* Λ *berechenbar.*

Aus (5.6) und $X < p_{s_0}^R$ folgt, daß die beiden Gleichungssysteme

$$\begin{aligned} \sum_{i=1}^n m_i\, x_i &= 0\,, & (5.9a) \\ 1 \le \|m\|_\infty &\le \rho & (5.9b) \end{aligned} \quad \text{und} \quad \begin{aligned} \sum_{i=1}^n m_i\, r_i &\equiv 0 \bmod p_{s_0}^R\,, & (5.10a) \\ 1 \le \|m\|_\infty &\le \rho & (5.10b) \end{aligned}$$

dieselbe ganzzahlige Lösungsmenge haben.

Im zweiten Schritt transformieren wir die Kongruenz (5.10a) in n Variablen in ein System von $n+1$ Kongruenzen in einer Variablen. Hierzu definieren wir für einen ganzzahligen nicht trivialen Vektor m die Größen

$$Z := \sum_{j=1}^n m_j\, r_j\ ,\ \ H := \sum_{j=1}^n |r_j|$$

und $B := \prod_{i=1}^n R_i$. Hieraus folgt für $1 \le \|m\|_\infty \le \rho$ insbesondere $Z \le \rho\, H$ und mit (5.8) aus Lemma 5.4

$$Z \ \le\ \rho\, n \frac{1}{2\,\rho\,(n+1)} B \ <\ B/2\,. \tag{5.11}$$

Lemma 5.5 *Sei* $opt_{\mathrm{mod}\ SIR_\infty}(r_1, \ldots, r_n\,;\ p_{s_0}^R)$ *die* ∞*–Norm der kürzesten nicht trivialen ganzzahligen Lösung von (5.10a). Dann gilt*

$$\begin{aligned} & opt_{\mathrm{mod}\ SIR_\infty}(r_1, \ldots, r_n\,;\ p_{s_0}^R) = 1 \\ \Longrightarrow\ & \exists Z \in [-H, H] \cap \mathbb{Z} - \{0\}\ : Z \equiv 0 \bmod p_{s_0}^R\ : \qquad (5.12) \\ & \Big(\forall m = (m_1, \ldots, m_n)^\top \in \mathbb{Z}^n - \{0\}\,,\ Z \equiv m_j\, r_j \bmod R_j\ ,\ j = 1, \ldots, n\ :\ \|m\|_\infty = 1\Big)\ , \\ & opt_{\mathrm{mod}\ SIR_\infty}(r_1, \ldots, r_n\,;\ p_{s_0}^R) > \rho \\ \Longrightarrow\ & \forall Z \in [-\rho\, H, \rho\, H] \cap \mathbb{Z} - \{0\}\,,\ Z \equiv 0 \bmod p_{s_0}^R\ : \qquad (5.13) \\ & \Big(\forall m = (m_1, \ldots, m_n)^\top \in \mathbb{Z}^n - \{0\}\,,\ Z \equiv m_j\, r_j \bmod R_j\ ,\ j = 1, \ldots, n\ :\ \|m\|_\infty > \rho\Big)\ . \end{aligned}$$

Beweis. (5.12): Sei $m = (m_1, \ldots, m_n)^\top$ eine Lösung von (5.10a) mit $\|m\|_\infty = opt_{\mathrm{mod}\ SIR_\infty}(r_1, \ldots, r_n\,;\ p_{s_0}^R) = 1$. Für $Z := \sum_{j=1}^n m_j\, r_j$ gilt nach (5.5) und (5.7) von Lemma 5.4 $Z \equiv m_j\, r_j \not\equiv 0 \bmod R_j$, denn mindestens ein m_j ist ungleich 0. Daher ist $Z \neq 0$. Aus $\|m\|_\infty = 1$ folgt weiterhin $|Z| \le H$. Außerdem gilt nach Definition $Z \equiv 0 \bmod p_{s_0}^R$ sowie wegen (5.5) aus Lemma 5.4

$$Z \ \equiv\ m_j\, r_j \bmod R_j\ ,\ j = 1, \ldots, n\,.$$

(5.13): Wir nehmen an, daß ein $Z \in [-\rho\, H, \rho\, H] \cap \mathbb{Z} - \{0\}$ existiert mit $Z \equiv 0 \bmod p_{s_0}^R$ und für alle $m = (m_1, \ldots, m_n)^\top \in \mathbb{Z}^n - \{0\}$ mit $Z \equiv m_j\, r_j \bmod R_j$, $j = 1, \ldots, n$ gilt, daß $\|m\|_\infty \le \rho$, das heißt $|m_j| \le \rho$, $j = 1, \ldots, n$. Um einen Widerspruch herzuleiten, beweisen wir die Existenz einer Lösung $m \in \mathbb{Z}^n - \{0\}$ für (5.10a) mit $1 \le \|m\|_\infty \le \rho$.

Sei $\overline{m} := (\overline{m}_1, \ldots, \overline{m}_n)^\top \in \mathbb{Z}^n - \{0\}$ ein Kandidat für eine solche Lösung und $Z := \sum_{j=1}^n \overline{m}_j\, r_j$. Dann gilt nach (5.5) aus Lemma 5.4

$$\overline{m}_j\, r_j \ \equiv\ Z \bmod\ R_j\ ,\ j = 1, \ldots, n\,.$$

Wir zeigen für das so gegebene Z, daß die Zahlen $\overline{m}_j \bmod R_j$, $j = 1, \ldots, n$ eindeutig bestimmt sind.

Nach (5.7) aus Lemma 5.4 ist $ggT(r_j, R_j) = 1$, $j = 1, \ldots, n$. Somit existiert für jedes $j = 1, \ldots, n$ ein eindeutig bestimmtes $r_j^\star \in [1, R_j] \cap \mathbb{Z}$ mit $r_j\, r_j^\star \equiv 1 \bmod R_j$. (Die Berechnung von $r_j^\star$ erfolgt mit Hilfe des Binären Euklidischen Algorithmus in $O((\lceil \log r_j \rceil)^2)$ Bitoperationen [Le38]. Die in Lemma 5.4 angegebene obere Schranke für die Bitkomplexität der Konstruktion der r_i, $i = 1, \ldots, n$ umfaßt diese obere Schranke.

Wir erhalten für alle $m_j \in [-\rho, \rho] \cap \mathbb{Z}$, $j = 1, \ldots, n$:

$$\overline{m}_j \;\equiv\; \overline{m}_j\, r_j\, r_j^\star \;\equiv\; m_j\, r_j\, r_j^\star \;\equiv\; m_j \bmod R_j \;.$$

Wir zeigen nun, daß sogar $\overline{m}_j \in [-\rho, \rho] \cap \mathbb{Z}$ gilt:

Nach dem Chinesichen Restsatz hat das System von n Kongruenzen

$$Z \;\equiv\; m_j\, r_j \bmod R_j \;,\; m_j \in [-\rho, \rho] \cap \mathbb{Z} \;,\; j = 1, \ldots, n$$

nach Definition von $B = \prod_{j=1}^n R_j$ genau $(2\,\rho + 1)^n$ mögliche Lösungen Z in dem Intervall $[-\frac{1}{2}\,B, \frac{1}{2}\,B]$.

Wegen (5.11) existieren höchstens $(2\,\rho + 1)^n$ Lösungen des Systems

$$\begin{aligned} Z \;&\equiv\; m_j\, r_j \bmod R_j \;,\; |m_j \bmod R_j| \leq \rho \;,\; 1 \leq j \leq n \\ |Z| \;&\leq\; \rho\, H \;. \end{aligned}$$

Andererseits können wir $(2\,\rho + 1)^n$ verschiedene Lösungen direkt angeben, nämlich genau alle $m := (m_1, \ldots, m_n)^\top \in \mathbb{Z}^n$ mit

$$m_j \in [-\rho, \rho] \cap \mathbb{Z} \;,\; j = 1, \ldots, n \;.$$

Alle diese Lösungen sind verschieden, denn falls $m_j^{(1)} \neq m_j^{(2)}$ für ein $1 \leq j \leq n$, so folgt für $Z^{(i)} := \sum_{j=1}^n m_j^{(j)}\, r_j$, $i = 1, 2$ mit $r_j \not\equiv 0 \bmod R_j$, daß

$$Z^{(1)} \;\equiv\; m_j^{(1)}\, r_j \;\not\equiv\; m_j^{(2)}\, r_j \;\equiv\; Z^{(2)} \bmod R_j \;,$$

ein Widerspruch.

Dies bedeutet, daß wir somit alle $(2\,\rho + 1)^n$ verschiedene Lösungen $m := (m_1, \ldots, m_n)^\top \in \mathbb{Z}^n$ mit $m_j \in [-\rho, \rho]$, $j = 1, \ldots, n$ gefunden haben.

Weiterhin gilt $Z = 0$ genau dann, wenn $m_j = 0$ für alle $j = 1, \ldots, n$. Aus $Z = \sum_{j=1}^n \overline{m}_j r_j$ und (5.6) von Lemma 5.4 folgt $Z \equiv 0 \bmod p_{s_0}^R$ und damit $opt_{\bmod\ SIR_\infty}(r_1, \ldots, r_n;\ p_{s_0}^R) \leq \rho$. □

Im dritten und letzten Schritt geben wir die Transformation des Systems von $n + 1$ Kongruenzen von Lemma 5.5 in eine Eingabe von $Min\ DA_\infty$ an:

Lemma 5.6 *Sei $((y_0, y_1, \ldots, y_n\,;\ N)$ die Eingabe von $Min\ DA_\infty$, die wie folgt definiert ist:*

$$\begin{aligned} N &:= R_1 \\ y_0 &:= \tfrac{1}{p^R_{s_0}} \\ y_j &:= \tfrac{r^\star_j}{R_j}\ ,\ j = 1, \ldots, n\ , \end{aligned}$$

wobei $r^\star_j$ das oben eindeutig bestimmte Inverse von $r_j \bmod R_j$ ist. Dann gilt

$$\begin{aligned} & \exists Z \in [-H, H] \cap \mathbb{Z} - \{0\}\ : Z \equiv 0 \bmod p^R_{s_0}\ : \\ & \Big(\forall m = (m_1, \ldots, m_n)^\top \in \mathbb{Z}^n - \{0\}\,,\ Z \equiv m_j\, r_j \bmod R_j\ ,\ j = 1, \ldots, n\ :\ \|m\|_\infty = 1\Big) \\ \Longrightarrow\ & \exists Z \in [-H, H] \cap \mathbb{Z} - \{0\}\ : \forall^n_{j=1} \min_{k\in\mathbb{Z}} |Z\, m_j - k|\ \le\ \frac{1}{N}\ , \end{aligned} \tag{5.14}$$

$$\begin{aligned} & \forall Z \in [-\rho\, H, \rho\, H] \cap \mathbb{Z} - \{0\}\,,\ Z \equiv 0 \bmod p^R_{s_0}\ : \\ & \Big(\forall m = (m_1, \ldots, m_n)^\top \in \mathbb{Z}^n - \{0\}\,,\ Z \equiv m_j\, r_j \bmod R_j\ ,\ j = 1, \ldots, n\ :\ \|m\|_\infty\ >\ \rho\Big) \\ \Longrightarrow\ & \forall Z \in [-\rho\, H, \rho\, H] \cap \mathbb{Z} - \{0\}\ : \exists_{j\in\{1,\ldots,n\}} \min_{k\in\mathbb{Z}} |Z\, m_j - k|\ >\ \frac{\rho}{N}\ . \end{aligned} \tag{5.15}$$

Beweis. (5.14): Sei $Z \in [-H, H] \cap \mathbb{Z} - \{0\}$ mit $Z \equiv 0 \bmod p^R_{s_0}$ und für alle $m := (m_1, \ldots, m_n)^\top \in \mathbb{Z}^n - \{0\}$ mit $Z \equiv m_j\, r_j \bmod R_j$, $j = 1, \ldots, n$ gelte $\|m\|_\infty = 1$. Offensichtlich ist dann

$$\min_{k\in\mathbb{Z}} |Z\, y_0 - k|\ =\ \min_{k\in\mathbb{Z}} |Z \frac{1}{p^R_{s_0}} - k|\ =\ 0\ .$$

Weiterhin folgt nach Voraussetzung sowie wegen (5.1) aus Lemma 5.3 für $j = 1, \ldots, n$:

$$\min_{k\in\mathbb{Z}} \left|Z \tfrac{r^\star_j}{R_j} - k\right| = \min_{k\in\mathbb{Z}} \left|\tfrac{m_j r_j r^\star_j}{R_j} - k\right| = \min_{k\in\mathbb{Z}} \left|\tfrac{m_j}{R_j} - k\right| \le \tfrac{1}{R_j} < \tfrac{1}{R_1}\ .$$

Somit existiert eine nicht triviale ganze Zahl Z mit den behaupteten Eigenschaften.

(5.15): Wir nehmen, es existiert eine ganze Zahl $Z \in [-\rho\, H, \rho\, H] \cap \mathbb{Z} - \{0\}$, so daß für alle $j = 0, 1, \ldots, n$

$$\min_{k\in\mathbb{Z}} |Z\, y_j - k| \le \tfrac{\rho}{R_1}\ ,$$

und zeigen, daß dann schon $\|m\|_\infty > \rho$ im Widerspruch zur Voraussetzung gelten muß. Aus (5.4) von Lemma 5.3 wissen wir, daß $\frac{1}{p^R_{s_0}} > \frac{\rho}{R_1}$. Mit $\min_{k\in\mathbb{Z}} |Z \frac{1}{p^R_{s_0}} - k| \le \frac{\rho}{R_1}$ folgt daher

$$\min_{k\in\mathbb{Z}} |Z \tfrac{1}{p^R_{s_0}} - k|\ =\ 0\ ,$$

und daher $Z \equiv 0 \bmod p^R_{s_0}$.

Nach (5.1) und (5.2) von Lemma 5.3 gilt $\frac{\rho+1}{R_j} > \frac{\rho}{R_1}$. In Verbindung mit $\min_{k\in\mathbb{Z}} |Z\, y_j - k| \le \frac{\rho}{R_1}$ erhalten wir

$$\min_{k\in\mathbb{Z}} |Z\, \tfrac{r^\star_j}{R_j} - k|\ \le\ \tfrac{\rho}{R_j}\ .$$

Diese Ungleichung kann jedoch nur erfüllt sein, wenn für alle $j = 1, \ldots, n$:

$$Z \equiv m_j r_j \mod R_j \ , \ |m_j| \leq \rho \ . \qquad \square$$

Beweis von Satz 5.2. Mit der Identität der Lösungsmengen der Systeme (5.9a) und (5.10a) und der in den Lemmata 5.5 und 5.6 beschriebenen polynomial–Zeit Transformationen des Systems (5.10a) auf die Eingabe $((y_0, y_1, \ldots, y_n) \, , \, N)$ von $Min\ DA_\infty$ erhalten wir eine gap–erhaltende Reduktion von $Min\ SIR_\infty$ auf $Min\ DA_\infty$.

Um zu beweisen, daß diese Reduktion polynomial–Zeit ist, genügt es zu zeigen, daß die Konstruktion der paarweise teilerfremden $p_{s_0}, R_1, \ldots, R_n$ und die Konstruktion der Zahlen $r_1, \ldots, r_n$ in polynomial vielen Bitoperationen in der binären Länge der Eingabe $(x_1, \ldots, x_n)^\top \in \mathbb{Z}^n - \{0\}$ von SIR_∞ möglich ist. Die Behauptung folgt daher mit den Lemmata 5.3 und 5.4. $\square$

Wir erhalten insgesamt folgendes Ergebnis:

Satz 5.7 *Falls* $\boldsymbol{NP} \not\subseteq \boldsymbol{QP}$, *dann existiert kein polynomial–Zeit Algorithmus, der für Eingaben* $y \in \mathbb{Q}^n$ *und* $N \in \mathbb{N}$ *eine ganze Zahl* $q^* \in [1, 2^{\log^{0.5-\zeta} bin(y)} N]$ *berechnet, so daß*

$$\|q^* y \mod \mathbb{Z}\|_\infty \ \leq 2^{\log^{0.5-\zeta} bin(y)} \min_{1 \leq q \leq N} \|q\, y \mod \mathbb{Z}\|_\infty \ .$$

Dabei ist ζ *eine beliebig kleine positive Kostante und* $bin(y)$ *die binäre Länge der Eingabe* y *von* $Min\ DA_\infty$.

5.2 Etwas Primzahltheorie

Für den Beweis der beiden Lemmata 5.3 und 5.4 benötigen wir folgendes Resultat aus der analytischen Zahlentheorie:

Satz 5.8 *(vergleiche [RoS62])*
Sei $\pi(n)$ *die Anzahl der Primzahlen kleiner oder gleich* $n \in \mathbb{N}$ *und* p_n *die* n*–te Primzahl. Dann gilt für* $n \geq 67$:

$$\frac{n}{\log n - \frac{1}{2}} \ < \ \pi(n) \ < \ 1.256 \frac{n}{\log n} \ , \tag{5.16}$$

$$n \log n \ < \ p_n \ < \ 1.34\, n \log n \ . \tag{5.17}$$

Lemma 5.3. *Seien* $x_1, \ldots, x_n \in \mathbb{Z}$, $\Lambda := n\, bin(\max_{1 \leq i \leq n} |x_i|)$, $\rho \in [1, \Lambda]$ *und* $X := \rho \sum_{i=1}^n |x_i|$. *Sei* $P := (p_k)_{k \in \mathbb{N}}$ *die Folge der Primzahlen und* p_{s_0} *die kleinste Primzahl mit* $p_{s_0} \not| \prod_{i=1}^n x_i$, *sowie* $R := \lfloor \log_{p_{s_0}} X \rfloor + 1$.

Dann existiert ein $s \in \mathbb{N}$, *so daß für eine Teilmenge* $\{p_{s_k} \, , \ 1 \leq k \leq n\} \subseteq P \cap (p_s, \frac{\rho+1}{\rho} p_s)$ *die Zahlen* $R_i := \prod_{j=1}^s p_j + p_{s_i}$, $i = 1, \ldots, n$, *paarweise teilerfremd sind und folgende*

Bedingungen erfüllen:

$$
\begin{aligned}
&(5.1) \qquad R_i \;<\; R_{i+1}\;,\; i=1,\ldots,n-1\;,\\
&(5.2) \qquad R_n \;<\; \frac{\rho+1}{\rho}\,R_1\;,\\
&(5.3) \qquad ggT(R_i, p_{s_0}\prod_{j=1}^{n} x_j) \;=\; 1\;,\; i=1,\ldots,n\;,\\
&(5.4) \qquad \prod_{j=1}^{s-1} p_j \;>\; 4\,\rho\, p_{s_0}^R\,(n+1)\;.
\end{aligned}
$$

Die Zahlen R_i , $i=1,\ldots,n$ *sind in* $O(\Lambda^5\,(\lceil\log\Lambda\rceil)^2)$ *vielen Bitoperationen konstruierbar.*

Beweis. Wir zeigen zuerst, daß für Primzahlen $p_{s_1},\ldots,p_{s_n}\in P\cap(p_s,\frac{\rho+1}{\rho}\,p_s)$ die Zahlen $R_i=\prod_{j=1}^{s}p_j+p_{s_i}$, $i=1,\ldots,n$ paarweise teilerfremd sind. Angenommen, dies ist nicht der Fall; dann existieren Indizes $1\le l<k\le n$ und eine Primzahl p, die $ggT(R_l,R_k)$ teilt. Dann teilt p auch R_l-R_k. Nach Konstruktion ist $R_k-R_l\in[2,\frac{p_s}{\rho})$. Primzahlen $p\in[2,\frac{p_s}{\rho})$ die R_l für ein $1\le l\le n$ teilen, würden nach Konstruktion von R_l aber p_{k_l} teilen, womit sofort der Widerspruch folgt.

Ungleichung (5.1) folgt nach Konstruktion der R_i und (5.2) ergibt sich aus der Bedingung $p_s\;<\;p_{s_i}\;<\frac{\rho+1}{\rho}\,p_s$, $i=1,\ldots,n$:

$$
R_n \;=\; \prod_{j=1}^{s}p_j+p_{s_n} \;<\; \prod_{j=1}^{s}p_j+\frac{\rho+1}{\rho}\,p_s \;<\; \frac{\rho+1}{\rho}\,R_1\;.
$$

(5.3): p_{s_0} ist die kleinste Primzahl mit $p_{s_0}\not|\prod_{i=1}^{n}x_i$. Die binäre Länge von $\prod_{i=1}^{n}x_i$ ist kleiner als $\sum_{i=1}^{n}(\lceil\log|x_i|\rceil+1)\le\Lambda$. Damit hat $\prod_{i=1}^{n}x_i$ höchstens Λ Primfaktoren. p_{s_0} ist daher eine der ersten $\Lambda+1$ Primfaktoren.

Wir wählen s groß genug, damit es mindestens n Zahlen $\prod_{j=1}^{s}p_j+p_{s_i}$ mit $p_{s_i}\in(p_s,\frac{\rho+1}{\rho}\,p_s)$ gibt, die (5.3) erfüllen. Das Produkt $p_{s_0}\prod_{j=1}^{n}x_j$ hat maximal $\Lambda+1$ Primfaktoren, von denen Primfaktoren $p\in\{p_1,\ldots,p_s\}$ die Zahlen $\prod_{j=1}^{s}p_j+p_{s_i}$ nicht teilen können. Das Intervall, in dem die R_i gewählt werden, hat Länge kleiner als $\frac{p_s}{\rho}$. Jeder der $\Lambda+1$ Primfaktoren, der $p_{s_0}\prod_{i=1}^{n}x_i$ teilt und größer als p_s ist, kann daher nur eine Zahl in diesem Intervall teilen. Unter $n+\Lambda+1$ Zahlen $\prod_{j=1}^{s}p_j+p_{s_i}$ mit $p_{s_i}\in(p_s,\frac{\rho+1}{\rho}\,p_s)$ sind daher mindestens n Zahlen, die (5.3) erfüllen. Wir nehmen als die R_i genau diese Zahlen. Im Anschluß an den Beweis von (5.4) geben wir explizit die Größe von s an.

(5.4): Nach Definition von R ist

$$
R \;\le\; \lceil\log_{p_{s_0}}(\rho\sum_{j=1}^{n}|x_i|)\rceil \;\le\; (2\,\log n+1.5\,\log\max_{1\le i\le n}|x_i|)/\log p_{s_0}+1\;.
$$

Nach (5.17) von Satz 5.8 gilt für $s\ge(\Lambda+3)+2\,\log n+1.5\,\log\max_{1\le i\le n}|x_i|$ daher insbesondere

$$
p_{s_0}^R \;=\; p_{s_0}^{R-1}\,p_{s_0} \;\le\; \prod_{j=1}^{s-1}p_j/(4\,\rho\,(n+1))\;.
$$

Wahl von s. Bei der Wahl von s muß sichergestellt sein, daß Bedingungen (5.3) und (5.4) erfüllt sind und mindestens $n+\Lambda+1$ verschiedene Primzahlen $p_{s_i} \in (p_s, \frac{\rho+1}{\rho} p_s)$, $i = 1, \ldots, n$ bzw. in $[1, \frac{\rho+1}{\rho} p_s)$ mindestens $n+\Lambda+1+s$ verschiedene Primzahlen existieren. Die Anzahl der Primzahlen in $[1, \frac{\rho+1}{\rho} p_s)$ ist nach (5.16) von Satz 5.8 mindestens

$$\frac{\rho+1}{\rho} \frac{p_s}{\log p_s} .$$

Nach (5.16) aus Satz 5.8 ist s daher so zu wählen, daß

$$s \quad > \quad 1.256\, \rho\, (n+\Lambda+1) . \tag{5.18}$$

Für $s := 2\,(\Lambda+1)^2$ erfüllt s Bedingungen (5.3), (5.4) sowie (5.18).

Anzahl der Bitoperationen. Wir führen den Primzahltest einer Zahl $z \in \mathbb{Z}$ mittels *Trial Division*, das heißt durch Division mit allen ungeraden Zahlen kleiner oder gleich $\lfloor\sqrt{|z|}\rfloor$ durch. Dies erfordert $\lfloor\sqrt{|z|}/2\rfloor\,(\lfloor\log\sqrt{|z|}\rfloor)^2$ viele Bitoperationen[1]. Um p_{s_0} zu finden, sind nach (5.17) aus Satz 5.8 damit höchstens $O(\lfloor\sqrt{\Lambda\,\log\Lambda}\rfloor\,(\lfloor\log\sqrt{\Lambda\,\log\Lambda}\rfloor)^2) = O(\lceil\sqrt{\Lambda}\,(\log\Lambda)^3\rceil)$ Bitoperationen erforderlich.

Um die $n+\Lambda+1$ Primzahlen in $[1, \frac{\rho+1}{\rho} p_s)$ zu finden, führen wir wiederum Trial Division durch. Dies erfordert nach (5.17) aus Satz 5.8 höchstens $O(\lceil\sqrt{p_s}\,(\log\sqrt{p_s})^2\rceil) = O(\lceil\sqrt{s\,\log s}\,(\log\sqrt{s\,\log s})^2\rceil) = O(\lceil\sqrt{s}\,(\log s)^3\rceil)$ Bitoperationen.

Wir zählen außerdem die Bitoperationen, die erforderlich sind, um $n+\Lambda+1$ Zahlen $\prod_{j=1}^{s} p_j + p_{s_i}$ auf Teilerfremdheit mit $p_{s_0}\,\prod_{i=1}^{n}|x_i| \;\leq\; 2^{\Lambda+1}$ zu testen. Den Test auf Teilerfremdheit führen wir mit dem Binären Euklidischen Algorithmus[2]durch. Nach (5.17) von Satz 5.8 ist

$$\prod_{j=1}^{s} p_j + p_{s_i} \quad\leq\quad 1.5\,\prod_{j=1}^{s} p_j \;\leq\; 2^{1.5\,s\,\log s} .$$

Ein Test auf Teilerfremdheit benötigt wegen $\Lambda+1 < s$ daher $O(s^2\,(\lceil\log s\rceil)^2)$ Bitoperationen. Die $(n+\Lambda+1)$–malige Anwendung eines solchen Tests kostet somit $O(2\,\Lambda\,s^2\,(\lceil\log s\rceil)^2)$ Bitoperationen. Die Anzahl der Bitoperationen für die Konstruktion der n Zahlen R_i ist nach obiger Wahl von s damit durch $O(\Lambda^5\,(\lceil\log\Lambda\rceil)^2 + \Lambda^2\,\lceil(\log\Lambda)^3\rceil) = O(\Lambda^5\,(\lceil\log\Lambda\rceil)^2)$ beschränkt. □

Die im Beweis von Lemma 5.3 angegebene Wahl von s impliziert folgendes

Lemma 5.4. *Seien die Bezeichnungen wie in Lemma 5.3 und R_i, $i = 1, \ldots, n$, die gemäß Lemma 5.3 konstruierten paarweise teilerfremden Zahlen $R_i := \prod_{j=1}^{s} p_j + p_{s_i}$ mit $p_s < p_{s_i} < \frac{\rho+1}{\rho} p_s$ für ein $s \;\geq\; 2\,(\Lambda+1)^2$.*

---

[1] Es werden $\lfloor\sqrt{|z|}/2\rfloor$ Divisionen durchgeführt, von denen jede nach der Schulmethode maximal $(\lfloor\log(\sqrt{|z|}/2)\rfloor)^2$ viele Bitoperationen kostet, siehe zum Beispiel [Ko87], Seite 126.

[2] Der Binäre Euklidische Algorithmus (nach [Le38]), angewendet auf Zahlen der Bitlänge ℓ benötigt $O(\ell^2)$ Bitoperationen auf Zahlen mit Bitlänge ℓ, [Kn81], Seiten 321–323.

Dann existieren ganze Zahlen $r_1, \ldots, r_n$ *, so daß für* $i = 1, \ldots, n$ *:*

$$\begin{aligned}
&(5.5) \qquad r_i \;\equiv\; 0 \mod \prod_{\substack{j=1\\ j\neq i}}^{n} R_j \ , \\
&(5.6) \qquad r_i \;\equiv\; x_i \mod p_{s_0}^R \ , \\
&(5.7) \qquad r_i \;\not\equiv\; 0 \mod R_i \ , \\
&(5.8) \qquad |r_i| \;<\; \frac{1}{2\,\rho\,(n+1)} \prod_{j=1}^{n} R_j \ .
\end{aligned}$$

Die Zahlen r_i *,* $i = 1, \ldots, n$ *sind in* $O(\Lambda^{13}\,(\lceil\log\Lambda\rceil)^7)$ *Bitoperationen berechenbar.*

Beweis. Nach Lemma 5.3 sind für jedes $i = 1, \ldots, n$ die Zahlen p_{s_0}, R_i und $S_i := \prod_{\substack{j=1\\ j\neq i}}^{n} R_j$ paarweise teilerfremd. Wir können daher den Chinesischen Restsatz für $i = 1, \ldots, n$ auf die Kongruenzen

$$\begin{aligned}
z_i &\equiv 0 \mod S_i \ , \\
z_i &\equiv x_i \mod p_{s_0}^R
\end{aligned}$$

anwenden. Dann existiert ein eindeutig bestimmtes $r_i^0 \in [1, S_i\,p_{s_0}]$, so daß

$$\begin{aligned}
r_i^0 &\equiv 0 \mod S_i \ , \\
r_i^0 &\equiv x_i \mod p_{s_0}^R \ .
\end{aligned}$$

Für jedes $i = 1, \ldots, n$ setzen wir nun $r_i^k := r_i^0 + k\,p_{s_0}^R\,S_i$ mit $k \in \mathbb{N} \cup 0$ und wählen als r_i das r_i^k mit minimalem Index k, so daß $ggT(r_i^k, R_i) = 1$ bzw. $r_i^k \not\equiv 0 \mod R_i$.

Da $ggT(R_i, p_{s_0}\,S_i) = 1$, kann ein Primteiler von R_i nur jeweils eines der r_i^k, $k = 0, 1, \ldots, R_i - 1$ teilen. R_i hat höchstens $\lfloor\log R_i\rfloor$ verschiedene Primteiler. Somit existiert ein Index $k \in [0, \ldots, \lfloor\log R_i\rfloor]$ mit $r_i^k \not\equiv 0 \mod R_i$. Nach (5.17) von Satz 5.8 und Konstruktion von R_i ist

$$\log R_i \;\leq\; \log\frac{\rho+1}{\rho} + \sum_{j=1}^{s}\log p_j \;\leq\; 1.5\,s\,\log s \;\leq\; 1.5\,p_s \ . \tag{5.19}$$

Da R_i keinen Primteiler kleiner oder gleich p_s besitzt, gibt es daher höchstens $1.5\,s\,\log s - s$ viele Primzahlen in $[p_s, 1.5\,p_s]$, die R_i und r_i^k teilen; dies bedeutet umgekehrt, daß wir nur höchstens $1.5\,s\,\log s - s$ viele r_i^k auf $ggT(R_i, r_i^k) = 1$ testen müssen, um eines zu finden, welches $r_i^k \not\equiv 0 \mod R_i$ erfüllt.

Mit obiger Wahl von s folgt wegen (5.4)

$$\begin{aligned}
|r_i^k| &\leq |r_i^0| + (1.5\,p_s - s)\,p_{s_0}^R \prod_{\substack{j=1\\ j\neq i}}^{n} R_j \\
&\leq \frac{2\,p_s\,p_{s_0}^R}{R_i} \prod_{j=1}^{n} R_j \;\leq\; \frac{1}{2\,\rho\,(n+1)} \prod_{j=1}^{n} R_j \ .
\end{aligned}$$

Anzahl der Bitoperationen. Für jedes $i = 1, \ldots, n$ ist nach (5.19) und (5.17) von Satz 5.8 die binäre Länge des Moduls $p_{s_0} \prod_{\substack{j=1 \\ j \neq i}} R_j$ durch

$$n + n \sum_{j=1}^{s} \log p_j \;\leq\; 1.5\, n\, s^2 \log s$$

beschränkt. Die n–malige Anwendung des Chinesischen Restsatzes mit Hilfe des Binären Euklidischen Algorithmus kostet damit $O(n\,(n\,s^2\,\lceil \log s \rceil)^2)$ Bitoperationen. Das Testen eines r_i^k auf $ggT(r_i^k, R_i) = 1$ geht mit dem Binären Euklidischen Algorithmus in $O((\lceil \log r_i^k \rceil)^2)$ Bitoperationen und erfordert wegen $r_i^k \leq \lfloor \log R_i \rfloor\, p_{s_0}^R \prod_{\substack{j=1 \\ j \neq i}} R_j$ und (5.19) insgesamt $O((n\,s^2\,(\lceil \log s \rceil)^3)^2))$ Bitoperationen. Dies schließt die Berechnung des Inversen $r_i^\star \equiv (r_i^k)^{-1} \bmod R_i$ mit ein. Da wir diesen Test für jedes $i = 1, \ldots, n$ maximal $\lfloor \log R_i \rfloor$ oft durchführen müssen, ist die Anzahl der Bitoperationen für alle Tests durch $O(n^3\,s^5\,(\lceil \log s \rceil)^7)$ beschränkt.

Nach Wahl von s benötigt daher die Berechnung der r_i, $i = 1, \ldots, n$ insgesamt $O(\Lambda^{13}\,(\lceil \log \Lambda \rceil)^7)$ Bitoperationen. $\square$

Literaturverzeichnis

[ABSS93] S. Arora, L. Babai, J. Stern, Z Sweedyk: The Hardness of Approximate Optima in Lattices, Codes and Systems of Linear Equations. Proc. 34th IEEE Symp. on Foundations of Computer Science (1993), Seiten 724–730.

[ALMSS92] S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy: Proof Verification and Hardness of Approximation Problems. Proc. 33rd IEEE Symp. on Foundations of Computer Science (1992), Seiten 14–23.

[Ar94] S. Arora: Probabilistic Checking of Proofs and Hardness of Approximation Problems. PhD Dissertation, University of California at Berkeley (1994).

[Berg80] G. Bergman: Notes on Ferguson and Forcade's Generalized Euclidean Algorithm. TR, Department of Mathematics, University of California, Berkeley, CA (1980).

[Bern71] L. Bernstein: The Jacobi–Perron Algorithm. Lecture Notes in Mathematics 207, Berlin–Heidelberg–New York (1971), Seiten 1–161.

[BBS92] L. Blum, M. Blum & M. Schub: A Simple Unpredictable Pseudo–random Number Generator. SIAM J. Comput., Vol. 15, No. 2 (1986), Seiten 364–383.

[BJM88] L. Babai, B. Just, F. Meyer auf der Heide: On the Limits of Computations with the Floor Function. Inform. Comput., 78 (1988), Seiten 99–107.

[Bor03] É. Borel: Contribution à l'analyse arithmétique du continu. Journal de Mathématiques Pures et Appliquées (5), 9 (1903), Seiten 329–375.

[Bre81] A.J. Brentjes: Multi–dimensional Continued Fraction Algorithm. Math. Centre Tracts 155, Amsterdam (1981).

[Bru20] V. Brun: En Generalisation av Kjedebrøken I+II. Norske Videnskapsselskapets Skrifter I. Matematisk–Naturvidenskapelig Klasse 6 (1919) und 6 (1920), Seiten 1–24.

[BK93] J. Buchmann und V. Kessler: Computing a Reduced Lattice Basis from a Generating System. Preprint (1993).

[Cas71] J.W.S. Cassels: An Introduction to the Geometry of Numbers, 2nd Printing. Springer–Verlag Berlin, Heidelberg, New York (1971).

[Cl92] K.L. CLARKSON: Safe and Effective Determinant Evaluation. Proc. 33rd IEEE Symp. on Foundations of Computer Science (1992), Seiten 387–395.

[Co71] S. A. COOK: The Complexity of Theorem Proving. Proc. 3rd ACM Symp. Theory of Computing (1971), Seiten 151–158.

[CP91] P. CRESCENZI, A. PANCONESI: Completeness in Approximation Classes. Inform. and Computation, Vol. 93 (1991), Seiten 241–262.

[Di1842] G.L. DIRICHLET: Verallgemeinerung eines Satzes aus der Lehre von den Kettenbrűchen nebst einigen Anwendungen auf die Theorie der Zahlen. Bericht űber die zur Bekanntmachung geigneten Verhandlungen der Kőniglich Preussischen Akademie der Wissenschaften zu Berlin (1842), pp. 93–95.

[Di1850] G.L. DIRICHLET: Über die Reduction der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. J. reine angew. Math. 40 (1850), Seiten 228–232.

[vEB81] P. VAN EMDE BOAS: Another **NP**–complete Partition Problem and the Complexity of Computing Short Vectors in a Lattice. Mathematics Department Report 81–04, University of Amsterdam, Amsterdam (1981).

[FB92] H.R.P. FERGUSON and D.H. BAILEY: A Polynomial Time, Numerically Stable Integer Relation Algorithm. RNR Technical Report RNR-91-032, NASA Ames Research Center, Moffett Field, CA (1992).

[FF79] H. FERGUSON and R. FORCADE: Generalization of the Euclidean Algorithm for Real Numbers to All Dimensions Higher than Two, Bull. Amer. Math. Soc., (New Series) 1 (1979), Seiten 912–914.

[FF82] H. FERGUSON and R. FORCADE: Multidimensional Euclidean Algorithm. J. Reine Angew. Math. 334 (1982), Seiten 171–181.

[FGLSS91] U. FEIGE, S. GOLDWASSER, L. LOVÁSZ, S. SAFRA, M. SZEGEDY: Approximating Clique is almost **NP**–complete. Proc. 32nd IEEE Symp. on Foundations of Computer Science (1991), Seiten 2–12.

[FL92] U. FEIGE, L. LOVÁSZ: Two–prover One–round Proof Systems: Their Power and Their Problems. Proc. 24th ACM Symp. Theory of Computing (1992), Seiten 643–654.

[Ga1801] C.F. GAUSS: Disquisitiones Arithmeticae. Leipzig 1801. Deutsche Übersetzung: Untersuchungen über die höhere Arithmetik. Springer, Berlin (1889). (reprint: Chelsea, New York, 1981.)

[Ge73] W.M. GENTLEMAN: Least Squares Computations by Givens Transformations Without Square Roots. J. Inst. Maths Applics, Vol. 12 (1973), Seiten 329–336.

[Ge75] W.M. GENTLEMAN: Error Analysis of QR Decomposition by Givens Transformations. Linear Algebra and its Applications, Vol. 10 (1975), Seiten 189–197.

[GoL89] G.H. GOLUB and C.F. VAN LOAN: Matrix Computations. The Johns Hopkins University Press, London (1989).

[GS91] J. GÖTZE und U. SCHWIEGELSOHN: A Square Root and Division Free Givens Rotation for Solving Least Squares Problems on Systolic Arrays. SIAN J. Sci. Stat. Comput., 12(4) (1991), Seiten 800–807.

[GrL87] P.M. GRUBER and C.G. LEKKERKERKER: Geometry of Numbers, 2nd Edition. North Holland, Amsterdam, New York, Oxford, Tokyo (1987).

[H95] C. HECKLER: Automatische Parallelisierung und parallele Gitterbasisreduktion. Dissertation an der Technischen Fakultät der Universität des Saarlandes, Saarbrücken (1995).

[HT93] C. HECKLER and L. THIELE: A Parallel Lattice Basis Reduction for Mesh–connected Processor Arrays and Parallel Complexity. Proceedings of the 5th Symposium on Parallel and Distributed Processing, Dallas (1993).

[He1845] C. HERMITE: Extraits de lettres de M. Ch. Hermite à M. Jacobi sur différents objets de la théorie des nombres. Deuxième lettre du 6 août 1845. J. Reine Angew. Math. 40 (1850), Seiten 279–290.

[HJLS89] J. HASTAD, B. JUST, J.C. LAGARIAS and C.P. SCHNORR: Polynomial Time Algorithms for Finding Integer Relations among Real Numbers. SIAM J. Comput., Vol. 18, No. 5 (1989), Seiten 859–881.

[Ja1868] C.G.J. JACOBI: Allgemeine Theorie der kettenbruchähnlichen Algorithmen, J. Reine Angew. Math. 69 (1868), Seiten 29–64.

[Jo93] A. JOUX: A Fast Parallel Lattice Basis Reduction Algorithm. Proceedings of the 2nd Gauss Symposium, Munich (1993).

[Ju92] B. JUST: Generalizing the Continued Fraction Algorithm to Arbitrary Dimensions. SIAM J. Comput., Vol. 21, No. 5 (1992), Seiten 909–926.

[Ka83] R. KANNAN: Polynomial–time Aggregation of Integer Programming Problems. J. ACM, Vol. 30 (1983), Seiten 133–145.

[Kh63] A. Y. KHINTCHINE: Continued Fractions. P. Noordhoff Groningen (1963).

[Kn81] D.E. KNUTH: The Art of Computer Programming, Vol. 2 (Seminumerical Algorithms). Second Edition, Addison Wesley (1981).

[Ko87] N. KOBLITZ: A Course in Number Theory and Cryptography. Graduate Texts in Mathematics 114, Springer Verlag, New York (1987).

[KZ1873] A. Korkine, G. Zolotarev: Sur les formes quadratiques. Math. Ann., Vol. 6 (1873), Seiten 336–389.

[Lag85] J. C. Lagarias: The Computational Complexity of Simultaneous Diophantine Approximation Problems. SIAM J. Comput. 14 (1985), Seiten 196–209.

[La1773] L. Lagrange: Recherches d'arithmetique. Nouv. Mém. Acad. Berlin (1773), Seiten 265–312.

[Le38] D.H. Lehmer: AMM, Vol. 45 (1938), Seiten 227–233.

[LLL82] A.K. Lenstra, H.W. Lenstra,Jr. and L. Lovász: Factoring Polynomials with Rational Coefficients. Math. Ann. 21 (1982), Seiten 515–534.

[LY94] C. Lund, M. Yannakakis: On the Hardness of Minimization Problems. J. ACM 41 (1994), Seiten 960–981.

[Mi05] H. Minkowski: Über die positiven quadratischen Formen und über kettenbruchähnliche Algorithmen, Ges. Abh. II, Seiten 243–260.

[PY91] C. H. Papadimitriou, M. Yannakakis: Optimization, Approximation and Complexity Classes. J. CSS, Vol. 43 (1991), Seiten 425–440.

[Pe07] O. Perron: Grundlagen für eine Theorie des Jacobischen Kettenbruchalgorithmus. Math. Ann. 64 (1907), Seiten 1–76.

[Po1884] H. Poincaré: Sur une généralisation des fractions continues. Comptes Rendus Acad. Sci., Paris 99 (1884), Seiten 1014–1016.

[RoS62] J. B. Rosser and L. Schoenfeld: Approximate Formulas for some Functions of Prime Numbers. Illinois Journal of Mathematics 6 (1962), Seiten 64–94.

[RSc95] C. Rössner and C.P. Schnorr: Computation of Highly Regular Nearby Points. Proceedings of the 3rd Israel Symposium on Theory of Computing and Systems, Tel Aviv (1995).

[RSe96a] C. Rössner and J.-P. Seifert: On the Hardness of Approximating Shortest Integer Relations among Rational Numbers. Proceedings of 2nd Computing: the Australasian Theory Symposium CATS'96, Melbourne (1996).

[RSe96b] C. Rössner and J.-P. Seifert: Approximating Good Simultaneous Diophantine Approximations is almost **NP**–hard. Proceedings of the 21st International Symposium on Mathematical Foundations of Computer Science, Krakau (1996), Springer Lecture Notes of Computer Science.

[Sc87] C.P. Schnorr: A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms. Theoretical Computer Science 53 (1987), Seiten 201–224.

[Sc88] C.P. Schnorr: A more Efficient Algorithm for Lattice Basis Reduction. Journal of Algorithms (1988), Seiten 47–62.

[SE94] C.P. Schnorr and M. Euchner: Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems. Math. Programming 66 (1994), Seiten 181–199.

[Sz70] G. Szekeres: Multidimensional Continued Fractions. Ann. Univ. Sci. Budapest, Eőtvős Sect. Math. 13 (1970), Seiten 113–140.

[Va86] B. Vallée: Une approche géometrique de la réduction des résaux en petite dimension. Thèse de Doctorat, Université de Caen (1986).

[Wi63] J.H. Wilkinson: Rounding Errors in Algebraic Processes. Prentice–Hall, Englewood Cliffs (1963).

[WS77] F. J. MacWilliams, N. J. A. Sloane: The Theory of Error–correcting Codes. North Holland, Amsterdam (1977).

Index

Lebenslauf

Name:	Carsten Nikolaus Rössner
Adresse:	Am Sandberg 57, 60599 Frankfurt am Main
Geburtstag und –ort:	19. Dezember 1964, Frankfurt am Main
Familienstand:	ledig
Schulbildung:	1971 – 1975 Grundschule an der Anne Frank–Schule in Offenbach 1975 – 1984 Gymnasium an der Albert Schweitzer–Schule in Offenbach mit Abschluß Abitur (Juni 1984)
Grundwehrdienst:	Juli 1984 – September 1985
Hochschulbildung:	Oktober 1985 – April 1991 Studium der Mathematik und Informatik an der Johann Wolfgang Goethe–Universität in Frankfurt am Main Oktober 1987: Vordiplom in Mathematik Dezember 1987: Vordiplom in Informatik April 1991: Diplom in Mathematik (Titel der Diplomarbeit: *Algorithmen zur Gitterreduktion und ganzzahligen Programmierung*)
Graduiertenstudium:	seit Mai 1991 Doktorand in der Arbeitsgruppe Mathematische Informatik der Johann Wolfgang Goethe–Universität;
Berufserfahrung:	von März 1988 bis März 1991 freiberufliche Tätigkeit bei der Johann W. Schimmel GmbH & Co. KG, Offenbach: Software–Beratung und –Installation für ein Midrange–System von Abteilungsrechnern seit Mai 1991 wissenschaftliche Tätigkeit an der Johann Wolfgang Goethe–Universität: Betreuung von Seminaren, Praktika, Workshops und Verwaltung eines lokalen Netzes von Workstations
Sprachen:	Englisch, Spanisch
Hobbys:	Lesen, Musik, Tennis, Fußball, Skifahren