

Diplomarbeit  
**Approximative Berechnung  
der Sterndiskrepanz**

vorgelegt von  
cand.-Dipl. Math. Carola Winzen

Institut für Informatik  
Christian-Albrechts-Universität zu Kiel

Kiel, im Juli 2007

1. Gutachter : Prof. Dr. Anand Srivastav
2. Gutachter : Prof. Dr. Dieter Betten

# Danksagung

Die vorliegende Arbeit entstand in der Zeit vom Januar bis Juli 2007 am Lehrstuhl für Diskrete Optimierung der Christian-Albrechts-Universität zu Kiel .

Herrn Professor Dr. Anand Srivastav möchte ich für die Übernahme der Betreuung und die Unterstützung dieser Arbeit danken.

Mein herzlicher Dank gilt auch Herrn Dr. Michael Gnewuch für die Initiierung des Themas sowie seine wertvollen Anregungen, die maßgeblich zum Gelingen der Arbeit beigetragen haben.

Kiel, im Juli 2007



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Einführung in die Theorie der Sterndiskrepanz</b>	<b>5</b>
2.1	Definition und Bedeutung der Sterndiskrepanz . . . . .	5
2.2	Diskretisierung der Diskrepanz . . . . .	7
<b>3</b>	<b>Komplexität</b>	<b>13</b>
3.1	Grundlagen der Komplexitätstheorie . . . . .	14
3.2	$\mathcal{NP}$ -Vollständigkeit der $(k, \varepsilon)$ -Boxen-Probleme . . . . .	23
3.3	Komplexitätstheoretische Betrachtung der Sterndiskrepanz . . . . .	33
<b>4</b>	<b>Berechnung der Sterndiskrepanz über <math>k</math>-Boxen</b>	<b>41</b>
4.1	Die $k$ -Boxen-Probleme . . . . .	42
4.2	Formulierung der LPs . . . . .	47
4.3	Berechnung der Sterndiskrepanz über $k$ -Boxen . . . . .	55
<b>5</b>	<b>Berechnung der Sterndiskrepanz mithilfe von Threshold Accepting</b>	<b>59</b>
5.1	Threshold Accepting . . . . .	59
5.2	Der Algorithmus von Winker und Fang . . . . .	62
5.3	Eine Alternative zum Algorithmus von Winker und Fang . . . . .	64
5.4	Vergleich der vorgestellten Algorithmen . . . . .	71
<b>6</b>	<b>Ausblick</b>	<b>77</b>

Literaturverzeichnis

79

# Kapitel 1

## Einleitung

Der Begriff „Diskrepanz“ (v. lat.: *discrepare* = uneinig, verschieden sein) dient der Beschreibung der Abweichung eines vorliegenden Zustandes zu einer als ideal anzusehenden Situation und ist ein in allen wissenschaftlichen Bereichen gebräuchlicher Begriff.

In der vorliegenden Arbeit werden wir uns in erster Linie mit der Theorie der geometrischen Diskrepanzen beschäftigen, welche auch Theorie der Irregularität von Verteilungen genannt wird. Sie findet ihre Anwendungen unter anderem in der numerischen Integration, in der Computergraphik sowie in der Finanzmathematik und ist eine vergleichsweise junge Disziplin der Mathematik.

Erste Arbeiten auf diesem Gebiet sind 1916 von Weyl [Wey16] veröffentlicht worden. Weyl beschäftigte sich vorrangig mit der Gleichverteilung von Folgen im eindimensionalen Fall. Höherdimensionale Situationen wurden erst einige Zeit später betrachtet.

Wir werden im Folgenden vor allem die noch zu definierende Sterndiskrepanz untersuchen, welche die Abweichung eines von endlich vielen Punkten erzeugten Maßes vom Lebesgue-Maß misst.

Wer noch andere Resultate aus der Theorie der Diskrepanzen erfahren möchte, sei auf das Buch „The Discrepancy Method - Randomness and Complexity“ von Bernard Chazelle [Cha01] verwiesen.

Eine gute Einführung in die Theorie der geometrischen Diskrepanzen findet man im Buch „Geometric Discrepancy“ von Jiří Matoušek [Mat99]. Lesenswert sind dort unter anderem die Konstruktionen von Punktemengen mit geringer Diskrepanz sowie die Herleitung unterer Schranken für die Diskrepanz einer Menge von gegebener Mächtigkeit. Auf diese durchaus interessanten Verfahren können wir an dieser Stelle jedoch nicht weiter eingehen und legen dem interessierten Leser die Lektüre dieses Buches ans Herz.

Wir werden uns in der vorliegenden Arbeit in erster Linie mit der Komplexität der

Berechnung der Sterndiskrepanz sowie mit verschiedenen Algorithmen zur Approximation dieses Wertes auseinandersetzen.

Dazu werden wir zunächst in Kapitel 2 eine kurze Einführung in die Grundlagen der Theorie geometrischer Diskrepanzen geben und eine Formel zur Berechnung der Sterndiskrepanz einer gegebenen Punktmenge im eindimensionalen Fall herleiten.

Im dritten Kapitel wollen wir dann zunächst zeigen, dass die effiziente Berechnung der Diskrepanz einer gegebenen Folge im Allgemeinen nicht möglich ist.

Dass dieses Resultat von Interesse ist, zeigen Bemerkungen in [WF97], [DEM96] und [Mat99]. Dort wird die Frage nach der Komplexität der Berechnung von  $D_n^*(X)$  bzw. nach effizienten Algorithmen zur Berechnung der Diskrepanz gestellt.

In dieser Arbeit gelingt es uns nun erstmalig, diese bislang offene Frage zu beantworten. Zudem werden wir zeigen können, dass die Möglichkeit einer beliebig guten Approximation der Sterndiskrepanz in einer von  $d$  und  $n$  polynomiell abhängigen Zeit ebenfalls nicht zu erwarten ist.

Darüber hinaus werden wir die von Thiémarc aufgeworfene Frage nach der Komplexität der Berechnung des Volumens kleinster bzw. größter von in  $0$  verankerten Boxen, welche eine gegebene Anzahl von Punkten enthalten, beantworten können. Er vermutet in [Thi01b] zwar bereits, dass dieses Problem  $\mathcal{NP}$ -vollständig ist, gibt jedoch an, dass er dieses nicht zu beweisen vermochte.

Auf die zum Beweis dieser Resultate benötigten komplexitätstheoretischen Grundlagen werden wir ebenfalls in Kapitel 3 eingehen.

In einem weiteren Schritt werden wir zwei Verfahren zur Berechnung der Diskrepanz vorstellen und dazu zunächst etwas näher auf die Arbeit [Thi01b] von Eric Thiémarc eingehen. Die von ihm vorgestellte Arbeit ermöglicht die exakte Berechnung der Diskrepanz einer Folge, kann aber auch zur Approximation dieses Wertes herangezogen werden. Es sei jedoch darauf hingewiesen, dass für uns in diesem Zusammenhang vor allem die theoretische Betrachtung der Sterndiskrepanz von Bedeutung sein wird.

In diesem Abschnitt werden wir auch eine Alternative zu den von Thiémarc eingeführten Begriffen vorstellen und zeigen, warum sie sich für die Berechnung der Diskrepanz besser anbietet.

Mithilfe dieser neuen Begriffe werden wir die Resultate von Thiémarc etwas verallgemeinern können. Zudem werden wir durch Angabe von Gegenbeispielen zwei von Thiémarc aufgestellte Behauptungen widerlegen.



In Kapitel 5 werden wir kurz eine von Winker und Fang in [WF97] beschriebene Heuristik zur Berechnung von  $D_n^*(X)$  vorstellen und einige Veränderungen an dieser vornehmen, um schließlich die so erhaltenen Heuristiken miteinander zu vergleichen. Diese Algorithmen wurden im Zuge der vorliegenden Arbeit in C implementiert; hier steht im Gegensatz zu den vorhergehenden Kapiteln die praktische Berechnung bzw. Approximation der Sterndiskrepanz im Vordergrund.

Schließlich wollen wir in Kapitel 6 einige offene Fragen, die sich an die hier vorgestellten Resultate anschließen, aufwerfen und einen kleinen Einblick in interessante Anknüpfungspunkte geben.



# Kapitel 2

## Einführung in die Theorie der Sterndiskrepanz

In diesem Kapitel werden wir nun zunächst die grundlegenden Definitionen sowie erste Eigenschaften der Sterndiskrepanz vorstellen. Wir werden zudem eine Formel zur Berechnung der Sterndiskrepanz in der eindimensionalen Situation herleiten und eine Übersicht über bisher veröffentlichte Ergebnisse zur Berechnung der Sterndiskrepanz einer gegebenen Punktemenge geben.

Zudem werden wir die Koksma-Hlawka-Ungleichung kennenlernen. Sie verdeutlicht die Notwendigkeit geeigneter Algorithmen zur Berechnung bzw. zur Approximation der Diskrepanz gegebener Punktemengen und zeigt die praktische Relevanz dieser Größe.

Wir beginnen nun mit der für die gesamte Arbeit gültigen

**Generalvoraussetzung:** Im Folgenden seien  $n, d$  natürliche Zahlen,  $X = (x^i)_{i=1}^n$  eine endliche Folge in  $[0, 1]^d$  und mit  $\lambda_d$  sei das  $d$ -dimensionale Lebesgue-Maß bezeichnet.

### 2.1 Definition und Bedeutung der Sterndiskrepanz

#### 2.1.1 Definition (Box, $A(y, X)$ )

Zu  $y = (y_j)_{j=1}^d \in [0, 1]^d$  definieren wir die von  $y$  erzeugte abgeschlossene Box

$$[0, y] = \prod_{j=1}^d [0, y_j]$$

und entsprechend die von  $y$  erzeugte halboffene Box

$$[0, y[ = \prod_{j=1}^d [0, y_j[.$$

Ferner setzen wir

$$V_y = \lambda_d([0, y]) = \lambda_d([0, y[)$$

und

$$A(y, X) = \sum_{i=1}^n 1_{[0, y[}(x^i) \quad \text{sowie} \quad \bar{A}(y, X) = \sum_{i=1}^n 1_{[0, y]}(x^i).$$

Es liegen damit genau  $A(y, X)$  Punkte von  $X$  in der von  $y$  erzeugten halboffenen und  $\bar{A}(y, X)$  Punkte von  $X$  in der von  $y$  erzeugten abgeschlossenen Box. Schon mit diesen wenigen Bezeichnungen können wir nun die von uns in dieser Arbeit näher zu untersuchende Größe definieren:

### 2.1.2 Definition (Sterndiskrepanz)

Der Wert

$$D_n^*(X) = \sup_{y \in [0, 1]^d} \left| V_y - \frac{1}{n} A(y, X) \right| \quad (2.1)$$

heißt Sterndiskrepanz von  $X$ . Die Sterndiskrepanz ist somit eine Größe für die Abweichung des Lebesgue-Maßes von dem diskreten Maß  $\frac{1}{n} A(\cdot, X)$ . Da wir uns im Folgenden fast ausschließlich mit der Sterndiskrepanz beschäftigen werden, sprechen wir mitunter einfach von Diskrepanzen, wenn aus dem Zusammenhang klar ist, welche Diskrepanz wir meinen.

Ein Ziel der Diskrepanztheorie ist die Berechnung von Folgen  $X$  mit möglichst geringer Diskrepanz. Man möchte also für jede in  $0$  verankerte Box das Volumen durch die anteilige Anzahl der Punkte innerhalb dieser (halboffenen) Box möglichst gut approximieren können. Der nun folgende Satz macht deutlich, warum solche Folgen für die Mathematik von Interesse sind:

### 2.1.3 Satz (Koksma-Hlawka-Ungleichung)

Die Grundidee numerischer Integration ist die Approximation eines Integrals durch Bildung eines gewichteten Mittelwertes über eine endliche Anzahl von Funktionswerten. Eine besonders einfache und daher in der Praxis häufig angewandte Methode der Approximation erhält man über sogenannte Quasi-Monte-Carlo-Methoden: hier werden

alle Funktionswerte gleich gewichtet, so dass eine umständliche a-priori Berechnung der Gewichte entfällt.

Eine Abschätzung der Approximationsgüte von Quasi-Monte-Carlo-Methoden liefert die folgende (scharfe) Ungleichung:

Ist  $f : [0, 1]^d \rightarrow \mathbb{R}$  eine zu integrierende Funktion, so gilt für jede endliche Menge  $M \subseteq [0, 1]^d$  die Koksma-Hlawka-Ungleichung

$$\left| \int_{[0,1]^d} f(s) ds - \frac{1}{|M|} \sum_{m \in M} f(m) \right| \leq V(f) D_n^*(M). \quad (2.2)$$

Dabei bezeichnet  $V(f)$  die Variation von  $f$  im Sinne von Hardy und Krause, welche nur von der Funktion  $f$  selber sowie der Dimension  $d$  abhängt. Für unsere Zwecke genügt die Einsicht, dass der Approximationsfehler in (2.2) proportional zur Diskrepanz der Menge  $M$  ist. Wir verzichten daher an dieser Stelle auf eine exakte Definition der Hardy-Krause-Variation.

Auch auf den Beweis wollen wir nicht näher eingehen, er kann jedoch in [KN74] nachgelesen werden. Die Originalarbeiten sind [Kok43] und [Hla61].

Eine Verallgemeinerung der Koksma-Hlawka-Ungleichung, welche die recht technische Hardy-Krause-Variation vermeidet und sich stattdessen der Theorie von Sobolevräumen bedient, findet man in [HSW04].

## 2.2 Diskretisierung der Diskrepanz

Die oben angegebene Definition (2.1) suggeriert, dass wir zur Berechnung der Diskrepanz unendlich viele Punkte heranziehen müssen. Glücklicherweise genügt jedoch die Auswertung endlich vieler Punkte:

### 2.2.1 Definition (Gitter $\Gamma(X)$ )

Für alle  $j \in \mathbb{N}_{\leq d}$  sei

$$\Gamma_j(X) = \{x_j^i ; i \in \mathbb{N}_{\leq n}\} \quad \text{und} \quad \bar{\Gamma}_j(X) = \Gamma_j(X) \cup \{1\}.$$

Wir setzen weiter

$$\Gamma(X) = \prod_{j=1}^d \Gamma_j(X) \quad \text{sowie} \quad \bar{\Gamma}(X) := \prod_{j=1}^d \bar{\Gamma}_j(X)$$

und nennen  $\Gamma(X)$  sowie  $\bar{\Gamma}(X)$  die von  $X$  induzierten Gitter.

### 2.2.2 Lemma (Berechnung von $D_n^*(X)$ )

Es gilt

$$D_n^*(X) = \max \left\{ \max_{y \in \bar{\Gamma}(X)} V_y - \frac{1}{n} A(y, X), \max_{y \in \Gamma(X)} \frac{1}{n} \bar{A}(y, X) - V_y \right\}. \quad (2.3)$$

*Beweis.* Eine einfache Überlegung zeigt

$$D_n^*(X) = \sup_{y \in [0,1]^d} \left\{ V_y - \frac{1}{n} A(y, X), \frac{1}{n} \bar{A}(y, X) - V_y \right\}.$$

Sei nun  $y \in [0, 1]^d$ . Dann gilt im Fall  $y_j \geq \min \bar{\Gamma}_j(X)$ ,  $1 \leq j \leq d$  für

$$y' = (\max \{x_j \in \Gamma_j(X) ; x_j \leq y_j\})_{j=1}^d$$

schon  $\bar{A}(y, X) = \bar{A}(y', X)$  und  $V_{y'} \leq V_y$ .

Im Fall der Existenz eines  $j \in \mathbb{N}_{\leq d}$  mit  $y_j < \min \bar{\Gamma}_j(X)$  gilt  $\bar{A}(y, X) = 0$ , weswegen dieser Fall nicht weiter zu beachten ist.

Analog erhalten wir für

$$y'' = (\min \{x_j \in \bar{\Gamma}_j(X) ; x_j \geq y_j\})_{j=1}^d$$

bereits  $A(y, X) = A(y'', X)$  sowie  $V_{y''} \geq V_y$ .

□

Nun haben wir das kontinuierliche Problem (2.1) auf das diskrete Problem (2.3) zurückführen können. Nach obiger Formel müssen wir jedoch im Allgemeinen noch immer  $(n+1)^d$  Punkte auswerten, was vor allem für große  $d$  in der Praxis zu zeitaufwendig sein wird.

Man kann sich zwar überlegen, dass die Anzahl der zu berechnenden Größen weiter gesenkt werden kann, werden jedoch in Kapitel 3 auch ein Resultat kennenlernen, welches uns zeigt, dass unter der Annahme  $\mathcal{P} \neq \mathcal{NP}$  eine Berechnung von  $D_n^*(X)$  in einer von  $n$  und  $d$  polynomiell abhängigen Anzahl von Schritten im Allgemeinen nicht möglich ist (die Begriffe  $\mathcal{P}$  und  $\mathcal{NP}$  werden ebenfalls in Kapitel 3 erläutert).

Wir benötigen daher Algorithmen, die uns auch in hohen Dimensionen zumindest eine Approximation von  $D_n^*(X)$  erlauben.

Bevor wir uns jedoch den höheren Dimensionen zuwenden, wollen wir noch ein Korollar aus 2.2.2 gewinnen:

### 2.2.3 Korollar (Berechnung von $D_n^*(X)$ im Fall $d = 1$ )

Es gelte  $d = 1$  und die Folgenglieder seien so sortiert, dass  $x^1 \leq \dots \leq x^n$  gilt. Dann erhalten wir die auf Niederreiter (vgl. auch [KN74]) zurückgehende Formel

$$D_n^*(X) = \frac{1}{2n} + \max_{1 \leq i \leq n} \left| x^i - \frac{2i-1}{2n} \right|.$$

*Beweis.* Nach 2.2.2 gilt mit  $x^{n+1} = 1$  schon

$$\begin{aligned} D_n^*(X) &= \max \left\{ \max_{y \in \bar{\Gamma}(X)} V_y - \frac{1}{n} A(y, X), \max_{y \in \Gamma(X)} \frac{1}{n} \bar{A}(y, X) - V_y \right\} \\ &= \max \left\{ \max_{1 \leq i \leq n+1} x^i - \frac{i-1}{n}, \max_{1 \leq i \leq n} \frac{i}{n} - x^i \right\} \\ &= \max_{1 \leq i \leq n} \left\{ \max \left\{ x^i - \frac{i-1}{n}, \frac{i}{n} - x^i \right\} \right\} \\ &= \max_{1 \leq i \leq n} \left\{ \frac{x^i - \frac{i-1}{n} + \frac{i}{n} - x^i}{2} + \frac{\left| x^i - \frac{i-1}{n} - \left( \frac{i}{n} - x^i \right) \right|}{2} \right\} \\ &= \frac{1}{2n} + \max_{1 \leq i \leq n} \left| x^i - \frac{2i-1}{2n} \right| \end{aligned}$$

Dabei haben wir im vorletzten Schritt die für alle  $r, s \in \mathbb{R}$  gültige Gleichung  $\max\{r, s\} = \frac{r+s}{2} + \frac{|r-s|}{2}$  ausgenutzt.  $\square$

Insbesondere erhalten wir aus Korollar 2.2.3 die Ungleichung  $D_n^*(X) \geq \frac{1}{2n}$ . Die Gleichheit gilt, wie wir ebenfalls der Ungleichung entnehmen können, genau dann, wenn  $x^i = \frac{2i-1}{n}$  für alle  $i \in \mathbb{N}_{\leq n}$  gilt.

Auch für andere niedrigdimensionale Situationen existieren Formeln zur exakten Berechnung der Sterndiskrepanz:

De Clerk hat in [Cle86] eine Formel für Folgen in  $[0, 1]^2$  mit paarweise disjunkten Koordinaten angegeben. Eine Verallgemeinerung auf nicht notwendigerweise disjunkte Koordinaten findet man in [BZ93]. Die dort angegebene Formel kann auch zur Berechnung der Sterndiskrepanz einer Folge in anderen niedrigdimensionalen Einheitswürfeln  $[0, 1]^d$  herangezogen werden. Für die Berechnung der Diskrepanz im höherdimensionalen Fall ist diese Formel aufgrund ihrer Laufzeit von  $O\left(\frac{n^d}{d!}\right)$  jedoch nicht anwendbar.

Eine Alternative findet man in [DEM96], wo ein Algorithmus mit Laufzeit  $O(n^{1+\frac{d}{2}})$  angegeben ist, was jedoch für große  $d$  immer noch zu viel ist. Zudem wurde dieser Algorithmus bisher unserer Kenntnis nach nur für Dimensionen  $d \leq 2$  implementiert.

Zwar keine Möglichkeit zur exakten Berechnung, aber immerhin zur Approximation der Sterndiskrepanz einer gegebenen Folge wollen wir hier kurz vorstellen, da sie sich ebenfalls

dirket aus 2.2.2 ableiten lässt.

### 2.2.4 Approximation von $D_n^*(X)$ mithilfe von $\delta$ -Covern

Die nun folgende Methode erlaubt die Berechnung oberer und unterer Schranken von  $D_n^*(X)$ .

Wir zerlegen dazu zunächst den Einheitswürfel  $[0, 1]^d$  in paarweise disjunkte halboffene achsenparallele Boxen  $[x, y[$  und setze für jede Box  $B$

$$B^- := [0, x[ \quad \text{sowie} \quad B^+ := [0, y[.$$

Die Partition bezeichnen wir mit  $\mathcal{B}$ . Schließlich setzen wir für jede Folge  $X$

$$U(\mathcal{B}, X) := \max_{B \in \mathcal{B}} \max \left\{ \frac{A(B^+, X)}{n} - \lambda_d(B^-), \lambda_d(B^+) - \frac{A(B^-, X)}{n} \right\}$$

sowie

$$L(\mathcal{B}, X) := \max_{B \in \mathcal{B}} \max \left\{ \left| \frac{A(B^-, X)}{n} - \lambda_d(B^-) \right|, \left| \frac{A(B^+, X)}{n} - \lambda_d(B^+) \right| \right\}$$

und erhalten so durch einfache Umformungen die Ungleichungen

$$L(\mathcal{B}, X) \leq D_n^*(X) \leq U(\mathcal{B}, X).$$

Zudem kann man ebenfalls schnell nachrechnen, dass das Volumen des Intervalls  $[L(\mathcal{B}, X), U(\mathcal{B}, X)]$  nicht größer ist als die maximale Differenz der Volumina  $\lambda_d(B^+) - \lambda_d(B^-)$  der Boxen in  $\mathcal{B}$ , i.e.

$$U(\mathcal{B}, X) - L(\mathcal{B}, X) \leq \max\{\lambda_d(B^+) - \lambda_d(B^-) ; B \in \mathcal{B}\}.$$

Diese Idee wirft die Frage auf, wie klein das Gewicht

$$w(\mathcal{B}) := \max\{\lambda_d(B^+) - \lambda_d(B^-) ; B \in \mathcal{B}\}$$

für gegebene Kardinalität der Partition werden kann.

Das folgende Programm etwa kann zur Konstruktion eines äquidistanten Gitters minimalen Gewichtes und Kardinalität  $\leq k^d$  verwendet werden:

$$\min_{y_1, \dots, y_k} \max_{i \in \{1, \dots, k\}} \prod_{j=1}^s y_{i_j} - \prod_{j=1}^s y_{i_{j-1}}$$

$$\text{s.t. } 0 = y_0 \leq y_1 \leq \dots \leq y_{k-1} \leq y_k$$



Man überlegt sich jedoch schnell, dass geeignetere Zerlegungen  $\mathcal{B}$  von  $[0, 1]^d$  (d.h. Zerlegungen kleineren Gewichtes) mit  $|\mathcal{B}| \leq k^d$  existieren.

Andersherum kann man sich auch fragen, wie groß für gegebenes  $\varepsilon > 0$  die Kardinalität der kleinsten Zerlegung  $\mathcal{B}_\varepsilon$  mit Gewicht  $w(\mathcal{B}_\varepsilon) \leq \varepsilon$  ist. Als Antwort auf diese Frage findet man mögliche Konstruktionen in [Thi01a] und [DGS05].

Thiémard gibt in seinem Artikel eine Konstruktion an, die zu gegebenem  $\varepsilon$  eine Partition  $\mathcal{B}_\varepsilon$  liefert, für die

$$|\mathcal{B}_\varepsilon| \leq \binom{d+h}{d} \quad \text{mit} \quad h = \left\lceil \frac{d \cdot \log(\varepsilon)}{\log(1-\varepsilon)} \right\rceil$$

gilt und zeigt zudem, dass die Werte  $A(B^+, X)$  sowie  $A(B^-, X)$  mithilfe von Datenstrukturen, die auf sogenannten „range trees“ aufbauen, in  $O((\log n)^d)$  berechnet werden können, während ein naives Abzählen der Werte  $O(n \cdot d)$  Operationen erfordert.

Man kann somit in

$$O((d + (\log n)^d) \cdot |\mathcal{B}_\varepsilon| + C^d \cdot n \cdot (\log n)^d)$$

Rechenschritten ein Intervall  $[L(\mathcal{B}_\varepsilon, X), U(\mathcal{B}_\varepsilon, X)]$  mit  $D_n^*(X) \in [L(\mathcal{B}_\varepsilon, X), U(\mathcal{B}_\varepsilon, X)]$  und Länge  $w(\mathcal{B}_\varepsilon) \leq \varepsilon$  berechnen. Für die Konstruktion des „range trees“ benötigen wir dabei  $O(C^d \cdot n \cdot (\log n)^d)$  Schritte, wobei  $C \in \mathbb{R}_{>1}$ , für die Konstruktion der Partition  $O(d \cdot |\mathcal{B}_\varepsilon|)$  Rechenschritte. Wir wollen auf diese eher technischen Konstruktionen hier nicht weiter eingehen, werden aber in 4.3.3 noch einmal auf die hier vorgestellte Idee zurückkommen.



# Kapitel 3

## Komplexität

In diesem Kapitel widmen wir uns der Frage nach der Komplexität des Diskrepanzproblems sowie der verwandten  $k$ -Boxen-Probleme.

Dazu werden wir im ersten Unterkapitel eine kurze Einführung in die Komplexitätstheorie geben. Hier werden wir zunächst kennenlernen, wie man Entscheidungs- und Optimierungsprobleme formal definieren kann, was unter den Komplexitätsklassen  $\mathcal{P}$  und  $\mathcal{NP}$  zu verstehen ist und einige Grundlagen aus der Theorie der Approximierbarkeit vorstellen. Dabei verzichten wir auf eine vollständige Darstellung der Komplexitätsklassen und beschränken uns im Wesentlichen auf die in der vorliegenden Arbeit benötigten Begriffe.

In den darauffolgenden Unterkapiteln werden wir dann die  $k$ -Boxen-Probleme sowie das Sterndiskrepanzproblem näher betrachten und deren  $\mathcal{NP}$ -Vollständigkeit beweisen. Für das Problem der Sterndiskrepanz werden wir schließlich ein negatives Resultat über die Approximierbarkeit formulieren können.

Die Kenntnis einiger weniger graphentheoretischer Grundlagen werden im Folgenden vorausgesetzt, können aber in jedem in die Theorie der endlichen Graphen einführenden Buch nachgelesen werden.

Bevor wir jedoch mit unserer Exkursion in die Theorie der Komplexitätsklassen beginnen, treffen wir noch folgende

**Konvention:** Alle in diesem Kapitel betrachteten Graphen seien endlich, ungerichtet und schlingenlos.

## 3.1 Grundlagen der Komplexitätstheorie

### 3.1.1 Definition (Entscheidungs- und Optimierungsprobleme)

Ist  $\Sigma$  eine endliche Menge, genannt Alphabet, so ist ein Problem über  $\Sigma$  ein Paar  $\Pi = (I, \mathcal{L})$  mit  $I \subseteq \Sigma^*$  und  $\mathcal{L} \in \mathcal{F}(I, 2^{\Sigma^*})$ . Dabei bezeichnet  $\Sigma^*$  die Menge aller endlichen Folgen über  $\Sigma$  und  $\mathcal{F}(I, 2^{\Sigma^*})$  die Menge aller Funktionen mit Definitionsbereich  $I$  und Werten in der Potenzmenge von  $\Sigma^*$ .

$I$  verstehen wir als die Menge aller Instanzen des Problems  $\Pi$  und für alle Instanzen  $i \in I$  die Menge  $\mathcal{L}(i)$  als Menge der zulässigen Lösungen der Instanz  $i$ .

Man unterscheidet weiter in Entscheidungs-, Optimierungs- und Suchprobleme. Da wir uns im Folgenden ausschließlich mit den beiden erstgenannten beschäftigen werden, verzichten wir auf eine ausführliche Darstellung der Suchprobleme.

Das Paar  $E = (\Pi, S)$  heißt Entscheidungsproblem, wenn  $\Pi = (I, \mathcal{L})$  ein Problem mit  $\mathcal{L}(i) \in \{\{0\}, \{1\}\}$  für alle  $i \in I$  ist und  $S = \{i \in I ; \mathcal{L}(i) = \{1\}\}$  gilt.  $S$  wird häufig auch als Sprache bezeichnet.

Ein Optimierungsproblem hingegen ist ein Tripel  $O = (\Pi, c, Ziel)$ , wobei wieder  $\Pi = (I, \mathcal{L})$  ein Problem ist,  $c$  eine Kostenfunktion darstellt, d.h. für jede Instanz  $i \in I$  und jede zulässige Lösung  $l \in \mathcal{L}(i)$  ist  $c_i(l)$  ein reeller Wert (häufig als Kosten bezeichnet), und  $Ziel \in \{\min, \max\}$  gilt, je nachdem ob ein Maximierungs- oder aber ein Minimierungsproblem vorliegt.

Bei gegebener Instanz  $i \in I(O)$  ist im Rahmen eines Optimierungsproblems unsere Aufgabe somit die Berechnung des Wertes

$$Ziel \{c_i(l) ; l \in \mathcal{L}(i)\}.$$

Bei Vorliegen eines Suchproblems fragen wir uns nach einer zulässigen Lösung  $l^* \in \mathcal{L}(i)$  mit

$$c_i(l^*) = Ziel \{c_i(l) ; l \in \mathcal{L}(i)\}.$$

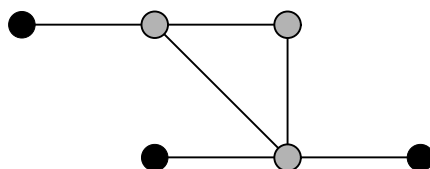
Wir wollen an dieser Stelle zur Verdeutlichung der Begriffe ein erstes Beispiel angeben und definieren dazu zunächst:

### 3.1.2 Definition (Clique)

Ist  $G = (V, E)$  ein Graph und  $C \subseteq V$ , so heißt  $C$  Clique von  $G$ , falls für alle  $c_1, c_2 \in C$  schon  $\{c_1, c_2\} \in E$  gilt.

### 3.1.3 Beispiel einer Clique

Die im unten angegebenen Graphen grau unterlegten Knoten bilden eine Clique der Größe drei.



Eines der bekanntesten Entscheidungsprobleme der Graphentheorie ist das folgende

### 3.1.4 Entscheidungsproblem (CLIQUE)

In der Notation von oben gilt  $\text{CLIQUE} = (I, \mathcal{L}, S)$  mit

$$I = \{(G, k) ; G = (V, E) \text{ Graph}, k \in \mathbb{N}_{\leq |V|}\},$$

$$\mathcal{L}((G, k)) = \begin{cases} \{1\} & , \text{ falls in } G \text{ eine Clique } C \text{ mit } |C| = k \text{ existiert} \\ \{0\} & , \text{ sonst} \end{cases}$$

und

$$S = \{(G, k) \in I ; \text{ in } G \text{ existiert eine Clique } C \text{ mit } |C| = k\}.$$

Wir wollen Entscheidungsprobleme im weiteren Verlauf jedoch wie folgt darstellen:

$$\begin{array}{ll} \text{Eingabe} & : \text{ Graph } G=(V,E), k \in \mathbb{N}_{\leq |V|} \\ \text{Frage} & : \text{ Existiert eine Clique } C \text{ von } G \text{ mit } |C| = k? \end{array}$$

Diese Darstellung erleichtert das Verständnis des vorliegenden Problems. Die formal korrekte Formulierung des entsprechenden Problems gemäß 3.1.1 ist jedoch stets tautologisch.

Das dazugehörige Optimierungsproblem lautet

### 3.1.5 Optimierungsproblem (CLIQUE)

Wir stellen das Problem zunächst wieder gemäß 3.1.1 dar und erhalten

$$\text{CLIQUE} = ((\{G = (V, E) ; G \text{ Graph}\}, \mathcal{L}), |\cdot|, \max),$$

wobei für alle Graphen  $G = (V, E)$  und alle Teilmengen  $C \subseteq V$

$$\mathcal{L}(G) := 2^V \quad \text{und} \quad |C|_G := |C|$$

gilt.

In einer etwas übersichtlicheren Darstellung sieht das Problem wie folgt aus:

Eingabe : Graph  $G=(V,E)$   
 Frage : Wie groß ist die größte Clique von  $G$ , d.h welchen Wert nimmt  
 $\max \{ |C| ; C \text{ ist eine Clique von } G \}$  an?

Wir werden später auf diese Probleme zurückkommen, fahren aber zunächst mit unserer Einführung in die Komplexitätstheorie fort und kommen nun zu dem Begriff des Algorithmus:

### 3.1.6 Definition (Algorithmus)

Ein Algorithmus  $\mathcal{A}$  ist ein Tripel  $(Q \cup \{q_0, \text{nein}, \text{ja}, \text{Halt}\}, \Sigma \cup \{\_ \}, \delta)$  bestehend aus

- einer endlichen Zustandsmenge  $Q \cup \{q_0, \text{nein}, \text{ja}, \text{Halt}\}$  mit Anfangszustand  $q_0$  und den drei Endzuständen halt, ja, nein
- einem endlichen Alphabet  $\Sigma$  und dem Symbol  $\_$  (Leerzeichen)
- einer Zustandsüberföhrungsfunktion

$$\delta : (Q \cup \{q_0\}) \times (\Sigma \cup \{\_ \}) \longrightarrow (Q \cup \{q_0, \text{nein}, \text{ja}, \text{Halt}\}) \times (\Sigma \cup \{\_ \}) \times \{\leftarrow, \rightarrow, -\}$$

Das dieser Definition zugrunde liegende Modell ist die Turingmaschine, ein im Jahr 1936 vom britischen Mathematiker Alan Turing in [Tur37] entwickeltes Modell zur Definition einer Klasse berechenbarer Funktionen.

Wir stellen uns unter der Turingmaschine ein unendlich langes (oder aber ein endliches, ausreichend langes), mit Elementen aus  $\Sigma$  beschriebenes Band vor, auf dem ein Lese- und Schreibkopf vom Anfangszustand  $q_0$  ausgehend verschiedene Operationen durchföhrt. Eine Besonderheit der Turingmaschine ist dabei, dass wir mit den drei Operationen „Lesen“, „Schreiben“ und „Kopf bewegen“ auskommen, um die von einem Computer berechenbaren Funktionen darstellen zu können.

Die auszuföhrenden Operationen werden von der Zustandsüberföhrungsfunktion  $\delta$  vorgegeben. Wie wir an der Definition von  $\delta$  erkennen können, hängt dabei die jeweils nächste

durchzuführende Operation vom aktuellen Zustand sowie dem Zeichen, welches sich in der Position des Bandes, über der sich der Lese- und Schreibkopf derzeit befindet, ab. Die Ausgabe ist ein neuer Zustand  $q \in Q \cup \{q_0, \text{nein}, \text{ja}, \text{Halt}\}$ , ein Zeichen, mit dem der Inhalt der Position, über der sich der Lese- und Schreibkopf befindet, überschrieben wird sowie einer Richtung  $\leftarrow, \rightarrow, -$ , in die sich der Lese- und Schreibkopf bewegen soll. Dabei steht die Richtung „-“ für ein Verharren auf der aktuellen Position.

Dieses Vorgehen wird solange wiederholt, bis einer der Endzustände ausgegeben wird.

Jedes  $x \in \Sigma^*$  heißt Eingabe für  $\mathcal{A}$ .  $\mathcal{A}$  ist ein Algorithmus für das Problem  $\Pi = (I, \mathcal{L})$ , wenn er für jede Eingabe  $i \in I$  terminiert, d.h. einen Endzustand erreicht.

Gilt für das Entscheidungsproblem  $E = ((I, \mathcal{L}), S)$  für alle  $i \in S$  bereits  $\mathcal{A}(i) = \text{ja}$  und  $\mathcal{A}(i) = \text{nein}$  für alle  $i \in I \setminus S$ , so sagen wir, dass der Algorithmus  $\mathcal{A}$  die Sprache  $S$  entscheidet und nennen  $\mathcal{A}$  einen Lösungsalgorithmus.

Ist  $O = ((I, \mathcal{L}), c, \text{Ziel})$  ein Optimierungsproblem und gilt für alle  $i \in I$

$$\mathcal{A}(i) \in \{l^* \in \mathcal{L}(i) ; c_i(l^*) = \text{Ziel} \{c_i(l) ; l \in \mathcal{L}(i)\}\},$$

so sprechen wir wiederum von einem Lösungsalgorithmus und sagen, dass  $\mathcal{A}$  das Problem  $O$  löst.

Wir belassen es an dieser Stelle bei diesen Erläuterungen, da zum Verständnis der formalen Begriffe jedes in die Komplexität einführende Buch herangezogen werden kann. Für die vorliegende Arbeit genügt ein intuitives Verständnis der genannten Begriffe. Bevor wir nun die Komplexitätsklassen  $\mathcal{P}$  und  $\mathcal{NP}$  definieren können, benötigen wir noch die folgenden Definitionen:

### 3.1.7 Definition (Laufzeit eines Algorithmus)

Ist  $\mathcal{A}$  ein Algorithmus für das Problem  $\Pi = (I, \mathcal{L})$ , so verstehen wir unter der zeitlichen Komplexität von  $\mathcal{A}$  eine Abbildung  $t_{\mathcal{A}} : \mathbb{N} \rightarrow \mathbb{N}$  mit

$$t_{\mathcal{A}}(n) = \max \{s \in \mathbb{N} ; \exists i \in I \cap \Sigma^n : \mathcal{A} \text{ terminiert bei Eingabe von } i \text{ nach } s \text{ Schritten}\}.$$

Die Länge der Folge  $i \in I$  heißt Eingabelänge und wird häufig mit  $|i|$  bezeichnet. Gilt nun  $t_{\mathcal{A}} = O(p)$  für eine Polynomfunktion  $p$ , so heißt  $\mathcal{A}$  polynomiell.

Auch hier genügt im Folgenden ein eher intuitives Verständnis. Als Eingabelänge kann man sich etwa bei einem graphentheoretischen Problem das Produkt der Anzahl der Kanten mit der Anzahl der Knoten vorstellen. Dazu berücksichtige man die bis auf Isomorphie eindeutige Darstellung eines Graphen durch seine Inzidenzmatrix.

Für eine endliche Folge  $X = (x^i)_{i=1}^n$  im  $d$ -dimensionalen Einheitswürfel  $[0, 1]^d$  ist die

Eingabelänge gerade

$$|X| = \sum_{i=1}^n \sum_{j=1}^d |x_j^i|,$$

wobei mit  $|x_j^i|$  die Länge der Binärkodierung der Zahl  $x_j^i$  bezeichnet sei.

Wir können an dieser Stelle noch festhalten und man kann sich schnell überlegen, dass ein polynomieller Lösungsalgorithmus für das Optimierungsproblem nur dann existieren kann, wenn ein solcher für das zugehörige Entscheidungsproblem zu finden ist. Manchmal gilt auch die Umkehrung, etwa bei dem Problem CLIQUE: Liegt für das Entscheidungsproblem CLIQUE ein polynomieller Lösungsalgorithmus vor, so löst folgender Algorithmus das Optimierungsproblem in einer von  $|V|$  polynomiell abhängigen Anzahl von Schritten:

1. Setze  $k=1$
2. solange  $\mathcal{A}(k) = \text{Ja}$ , setze  $k=k+1$
3. Gib  $k - 1$  aus

Da wir nun die zeitliche Komplexität kennengelernt haben, bietet es sich an, an dieser Stelle noch kurz auf den folgenden Begriff einzugehen:

### 3.1.8 Definition ( $\mathcal{NP}$ -Optimierungsproblem)

Ein Optimierungsproblem  $O = (\Pi, c, \text{Ziel})$  über der Menge  $\Sigma$  mit  $\Pi = (I, \mathcal{L})$  heißt  $\mathcal{NP}$ -Optimierungsproblem, falls

1. für alle  $i \in \Sigma^*$  in Polynomzeit entschieden werden kann, ob  $i \in I$  ist,
2. ein Polynom  $p$  existiert so, dass für alle  $i \in I$  und  $l \in \mathcal{L}(i)$  die Ungleichung  $|l| \leq p(|i|)$  gilt, die Kosten  $c_i(l)$  von  $l$  bezüglich  $i$  in Polynomzeit berechnet werden können und
3. für alle  $i \in I$  und  $l \in \Sigma^*$  mit  $|l| \leq p(|i|)$  in Polynomzeit entschieden werden kann, ob  $l \in \mathcal{L}(i)$  gilt.

### 3.1.9 Definition (Verifikationsalgorithmus)

Ist  $E = ((I, \mathcal{L}), S)$  ein Entscheidungsproblem und  $\mathcal{A}$  ein Algorithmus, so heißt  $\mathcal{A}$  Verifikationsalgorithmus für  $E$ , falls es eine Teilmenge  $Z$  von  $\Sigma^*$  gibt so, dass die folgenden Bedingungen erfüllt sind:



1.  $\mathcal{A}$  terminiert für alle  $(i, z) \in I \times Z$
2. Zu jedem  $i \in S$  existiert ein  $z \in Z$  mit  $\mathcal{A}(i, z) = \text{Ja}$
3. Für alle  $i \in I \setminus S$  und  $z \in Z$  gilt  $\mathcal{A}(i, z) = \text{Nein}$

Ist  $i \in I$  und  $z \in Z$  mit  $\mathcal{A}(i, z) = \text{Ja}$ , so heißt  $z$  auch Zertifikat für  $i$ . Wir können  $z$  als Beweis für die „Gültigkeit“ von  $i$  ansehen. Ist beispielsweise für das Entscheidungsproblem CLIQUE eine Instanz  $(G = (V, E), k \in \mathbb{N}_{\leq |V|})$  gegeben, so ist  $C \subseteq V$  ein Zertifikat für  $(g, k)$ , falls  $C$  eine Clique von  $G$  mit  $|C| \geq k$  ist.

### 3.1.10 Definition (polynomielle Reduzierbarkeit)

Sind  $E_1 = ((I_1, \mathcal{L}_1), S_1)$  und  $E_2 = ((I_2, \mathcal{L}_2), S_2)$  zwei Entscheidungsprobleme, so nennen wir  $E_1$  polynomiell auf  $E_2$  reduzierbar, in Zeichen  $E_1 \leq_p E_2$ , wenn ein polynomieller Algorithmus  $\mathcal{A}$  für  $E_1$  existiert so, dass

$$\mathcal{A}(i) \in I_2 \text{ und } i \in S_1 \Leftrightarrow \mathcal{A}(i) \in S_2 \quad \text{für alle } i \in I_1$$

gilt.  $\mathcal{A}$  heißt in dieser Situation auch polynomielle Transformation.

Der Algorithmus  $\mathcal{A}$  transformiert sozusagen eine Instanz  $i_1$  des Entscheidungsproblems  $E_1$  in eine solche von  $E_2$ . Solche Transformationen sind unter anderem dann sinnvoll, wenn man die Situationen des Entscheidungsproblems  $E_2$  bereits genauer untersucht hat und sich durch die Transformation eine Aussage über  $E_1$  erhofft. Es gibt aber auch andere Anwendungen der polynomiellen Transformationen, von denen wir im Folgenden noch häufig Gebrauch machen werden und die wir im Folgenden noch näher erläutern werden.

Nun haben wir alle Voraussetzungen beisammen, um die drei grundlegenden Komplexitätsklassen für Entscheidungsprobleme definieren zu können:

### 3.1.11 Definition (Die Klassen $\mathcal{P}$ , $\mathcal{NP}$ und $\mathcal{NPC}$ )

- Die Komplexitätsklasse  $\mathcal{P}$  ist die Klasse aller Entscheidungsprobleme  $E$ , für die ein polynomieller Lösungsalgorithmus existiert.
- Die Komplexitätsklasse  $\mathcal{NP}$  umfasst diejenigen Entscheidungsprobleme, für welche ein polynomieller Verifikationsalgorithmus existiert.
- Wir sagen, dass ein Entscheidungsproblem  $E$   $\mathcal{NP}$ -schwer ist, wenn für alle  $\tilde{E} \in \mathcal{NP}$  schon  $\tilde{E} \leq_p E$  gilt. Mit  $\mathcal{NPC}$  bezeichnen wir die Klasse aller  $\mathcal{NP}$ -vollständigen Entscheidungsprobleme, d.h. die Klasse aller  $\mathcal{NP}$ -schweren Probleme, welche selber in  $\mathcal{NP}$  liegen.

Offensichtlich gilt  $\mathcal{P} \subseteq \mathcal{NP}$  (ist  $\mathcal{A}$  ein polynomieller Lösungsalgorithmus für  $E = ((I, \mathcal{L}), S)$ , so erhalte vermöge  $\mathcal{A}' : I \times \Sigma^* \rightarrow \{ \text{Ja, Nein} \}, (i, z) \mapsto \mathcal{A}(i)$  einen polynomiellen Verifikationsalgorithmus), ungeklärt ist jedoch bislang die Frage, ob  $\mathcal{P} = \mathcal{NP}$  oder  $\mathcal{P} \subset \mathcal{NP}$  gilt.

Der im Jahre 1966 mit der Fields-Medaille ausgezeichnete US-amerikanische Mathematiker Stephen Smale bezeichnet in einer seiner Arbeiten (siehe [Sma98]) diese Frage neben der Riemann-Hypothese und der Poincaré-Vermutung (die im Jahr 2002 vom russischen Mathematiker Grigori Perelman bewiesen wurde) als eines der drei größten ungelösten Probleme der Mathematik.

Definiert man auf der Klasse aller Entscheidungsprobleme die Relation  $\sim_p$  durch

$$E_1 \sim_p E_2 \Leftrightarrow (E_1 \leq_p E_2 \wedge E_2 \leq_p E_1),$$

so ist  $\sim_p$  offensichtlich eine Äquivalenzrelation. Sowohl  $\mathcal{P}$  als auch  $\mathcal{NPC}$  sind Äquivalenzklassen von  $\sim_p$ . Um die Frage „ $\mathcal{P} = \mathcal{NP}$ ?“ zu beantworten, würde also genügen, ein Problem  $E \in \mathcal{NPC}$  mit polynomielltem Lösungsalgorithmus anzugeben. Da dieses bisher nicht gelungen ist, gehen viele Strukturwissenschaftler davon aus, dass  $\mathcal{P} \subset \mathcal{NP}$  gilt.

Wir werden uns im Folgenden in erster Linie mit  $\mathcal{NP}$ -vollständigen Problemen beschäftigen und wollen daher nun zunächst ein paar Beispiele solcher Entscheidungsprobleme kennenlernen. Die hier angegebenen Beispiele werden wir auch in späteren Kapiteln noch einmal betrachten.

Ein erstes klassisches Beispiel  $\mathcal{NP}$ -vollständiger Probleme ist das schon oben eingeführte CLIQUEN-Problem. In der Tat gilt der folgende

### 3.1.12 Satz (CLIQUE $\in \mathcal{NPC}$ )

Das Problem CLIQUE ist  $\mathcal{NP}$ -vollständig.

*Beweis.* • Um CLIQUE  $\in \mathcal{NP}$  zu zeigen, geben wir einen polynomiellen Verifikationsalgorithmus an:

Eingabe : Graph  $G = (V, E)$ , natürliche Zahl  $k \in \mathbb{N}$ , Lösungsvorschlag  $C \subseteq V$

Rechenschritte:

1. Ist  $|C| < k$ , so gib „Nein“ aus und halte.
2. Für alle  $v_1, v_2 \in C$  prüfe, ob  $\{v_1, v_2\} \in E$  gilt. Ist  $\{v_1, v_2\} \notin E$ , so gib „Nein“ aus und halte.

3. Gib „Ja“ aus und halte.

Man überzeugt sich schnell davon, dass der angegebene Algorithmus nach spätestens  $O\left(\binom{|C|}{2}\right)$  Schritten terminiert.

Da  $|C| \leq |G|$  gilt, hat der Algorithmus eine Laufzeit von  $O\left(\binom{|G|}{2}\right)$ .

- Da es sich bei diesem um das Standardbeispiel für  $\mathcal{NP}$ -vollständige Probleme handelt, führen wir den Beweis der  $\mathcal{NP}$ -Schwere von CLIQUE hier nicht durch, er kann jedoch unter anderem im Buch „Theoretische Informatik“ von Ingo Wegener ([Weg05]) nachgelesen werden.

□

Um einen ersten Einblick in typische Beweisverfahren zur Herleitung der  $\mathcal{NP}$ -Vollständigkeit eines gegebenen Problems zu bekommen, definieren wir noch das folgende Entscheidungsproblem, welches wir in 3.2 noch einmal zur Herleitung eines Komplexitätsergebnisses heranziehen werden.

### 3.1.13 Entscheidungsproblem (HALFCLIQUE)

Eingabe : Graph  $G=(V,E)$  mit gerader Knotenzahl  $|V| = 2k$   
 Frage : Existiert eine Clique  $H \subseteq V$  mit  $|H| = k$ ?

Der einzige Unterschied zum CLIQUEN-Problem besteht also darin, dass wir uns hier nunmehr nur fragen, ob es eine Clique der Mächtigkeit  $\frac{|V|}{2}$  gibt. Auch wenn das Problem auf den ersten Blick deutlich einfacher als das allgemeine CLIQUEN-Problem erscheinen mag, so können wir dennoch folgenden Satz festhalten:

### 3.1.14 Satz (HALFLIQUE $\in \mathcal{NPC}$ )

Auch das Problem HALFLIQUE ist  $\mathcal{NP}$ -vollständig.

*Beweis.* • HALFCLIQUE  $\in \mathcal{NP}$ : Folgt aus Resultat 3.1.12, da HALFCLIQUE einen Spezialfall von CLIQUE darstellt.

- HALFCLIQUE ist  $\mathcal{NP}$ -schwer: da gemäß Satz 3.1.12 das Problem CLIQUE  $\mathcal{NP}$ -schwer ist und  $\leq_p$  offensichtlich eine transitive Relation ist, genügt es CLIQUE auf HALFCLIQUE zu reduzieren:

Sei dazu  $G = (V, E)$ ,  $k \in \mathbb{N}_{\leq n}$  eine Instanz des Entscheidungsproblems CLIQUE. Wir unterscheiden die beiden Fälle:

1.  $k \geq \frac{|V|}{2}$ : Setze  $G' = (V \cup W, E)$ , wobei  $W$  eine zu  $V$  disjunkte Menge der Mächtigkeit  $|W| = 2k - |V|$  sei.  
Dann hat  $G'$  gerade  $|W| + |V| = 2k$  Knoten und besitzt genau dann eine Clique der Größe  $k$ , wenn eine solche in  $G$  existiert.
2.  $k < \frac{|V|}{2}$ : Wähle eine zu  $V$  disjunkte Menge  $W$  mit  $|W| = |V| - 2k$  und setze  $E' = E \cup \{\{w, v\} ; w \in W, v \in (W \setminus \{w\}) \cup V\}$  sowie  $G' = (V \cup W, E')$ .  
Offenbar besitzt nun  $G$  genau dann eine Clique der Größe  $k$ , wenn in  $G'$  eine der Größe  $|W| + k = |V| - k = \frac{1}{2}(|V \cup W|)$  existiert.

□

Nachdem wir uns nun etwas ausführlicher mit Entscheidungsproblemen beschäftigt haben, wollen wir im Folgenden noch kurz auf Optimierungsprobleme eingehen.

Weiß man von einem Optimierungsproblem, dass das zugehörige Entscheidungsproblem  $\mathcal{NP}$ -schwer ist, so weiß man auch für das Optimierungsproblem selbst, dass kein Lösungsalgorithmus mit polynomieller Laufzeit existiert und es ist natürlich, sich zu fragen, wie gut man das Problem in polynomieller Zeit approximieren kann. Man gibt somit die Hoffnung auf die exakte Berechnung einer zu untersuchenden Größe auf und lässt nunmehr kleine Abweichungen von einer Optimallösung zu. Wir haben oben etwa gelernt, dass das Entscheidungsproblem CLIQUE ein  $\mathcal{NP}$ -schweres Problem ist. Wir können uns nun fragen, ob wir wenigstens einen Algorithmus angeben können, der zu gegebenem Graphen  $G = (V, E)$  eine Clique  $T$  liefert, deren Mächtigkeit nicht sehr viel kleiner als die der Clique größter Kardinalität ist.

Dieses wollen wir nun formalisieren.

### 3.1.15 Definition ( $\varepsilon$ -Algorithmus)

Sei  $O = ((I, \mathcal{L}), c, \text{Ziel})$  ein Optimierungsproblem und  $\mathcal{A}$  ein Algorithmus mit  $\mathcal{A}(i) \in \mathcal{L}(i)$  für alle  $i \in I$ . Ist  $\varepsilon > 1$ , so nennen wir  $\mathcal{A}$  einen  $\varepsilon$ -Algorithmus von  $O$ , wenn für alle  $i \in I$

$$R(i, \mathcal{A}(i)) := \max \left\{ \frac{c_i(\mathcal{A}(i))}{c_i^*}, \frac{c_i^*}{c_i(\mathcal{A}(i))} \right\} \leq \varepsilon$$

gilt. Dabei sei  $c_i^* = \text{Ziel} \{c_i(l) ; l \in \mathcal{L}(i)\}$ .

Bei Vorliegen eines Maximierungsproblems gilt somit für einen  $\varepsilon$ -Algorithmus  $\mathcal{A}$

$$c_i(\mathcal{A}(i)) \geq \frac{c_i^*}{\varepsilon},$$

im Falle eines Minimierungsproblems erhalten wir

$$c_i(\mathcal{A}(i)) \leq \varepsilon c_i^*.$$

Der Wert  $R(i, \mathcal{A}(i))$  heißt auch Approximationsgüte von  $\mathcal{A}(i)$  für  $i$ .

### 3.1.16 Definition ((volles) Polynomzeit-Aproximationsschema)

Wir sagen, das Optimierungsproblem  $O = ((I, \mathcal{L}), c, Ziel)$  habe ein Polynomzeit-Approximationsschema (kurz: PTAS, engl. polynomial-time approximation scheme), wenn ein Algorithmus  $\mathcal{A}$  existiert so, dass  $\mathcal{A}$  zu jedem  $\varepsilon > 1$  ein  $\varepsilon$ -Algorithmus von  $O$  ist und die Laufzeit von  $\mathcal{A}$  polynomiell in der Eingabelänge  $|i|$  abhängt. Dabei kann die Laufzeit bei gegebenem Wert  $\varepsilon$  von der Länge seiner Binärcodierung abhängen, auch exponentiell.

Hängt  $\mathcal{A}$  in obiger Situation nicht nur von  $|i|$ , sondern auch von  $\frac{1}{\varepsilon-1}$  polynomiell ab, so heißt das Approximationsschema auch volles Polynomzeit-Aproximationsschema, kurz FPTAS (engl. fully polynomial-time approximation scheme).

## 3.2 $\mathcal{NP}$ -Vollständigkeit der $(k, \varepsilon)$ -Boxen-Probleme

In diesem Unterkapitel wollen wir die Komplexität der in Kapitel 4 näher betrachteten Probleme der  $k$ -Boxen untersuchen. Dazu werden wir die Probleme zunächst genauer definieren und dann zeigen, dass sie  $\mathcal{NP}$ -vollständig sind.

Wir werden in 4.1.8 zeigen, dass die Sterndiskrepanz mithilfe von  $k$ -Boxen berechnet werden kann. In diesem Abschnitt wollen wir aber zunächst der Übersichtlichkeit wegen nur die benötigten Definitionen angeben.

Diese Resultate deuten zwar bereits darauf hin, dass auch das von uns in erster Linie betrachtete Problem der Berechnung der Sterndiskrepanz nicht in einer von  $d$  und  $n$  polynomiell abhängigen Anzahl von Schritten lösbar sein wird (sofern  $\mathcal{P} \neq \mathcal{NP}$ ), beweisen diese Tatsache jedoch nicht. Dennoch werden wir für die Herleitung dieses Satzes die hier vorgestellten Methoden verwenden können.

### 3.2.1 $\mathcal{NP}$ -Vollständigkeit des $(k^-, \varepsilon^+)$ -Boxen-Problems

#### 3.2.1.1 Definition $((k^-, \varepsilon^+)$ -Box)

Sind  $d, n \in \mathbb{N}, k \in \mathbb{N}_0, \varepsilon > 0, X = (x^i)_{i=1}^n$  eine Folge in  $[0, 1]^d$  und  $y \in \bar{\Gamma}(X)$ , so heißt die von  $y$  erzeugte Box  $(k^-, \varepsilon^+)$ -Box, wenn  $A(y, X) \leq k$  und  $V_y \geq \varepsilon$  gilt.

In einer  $(k^-, \varepsilon^+)$ -Box liegen also höchstens  $k$  Punkte und das Volumen der Box ist  $\geq \varepsilon$ .

### 3.2.1.2 Entscheidungsproblem ( $(k^-, \varepsilon^+)$ -Box)

Zu  $k \in \mathbb{N}_0$  definieren wir das  $(k^-, \varepsilon^+)$ -BOXEN-Problem wie folgt:

Eingabe :  $d \in \mathbb{N}, n \in \mathbb{N}$ , endliche Folge  $(x^i)_{i=1}^n$  in  $[0, 1]^d, \varepsilon > 0$   
 Frage : Existiert eine halboffene  $(k^-, \varepsilon^+)$ -Box?

Wie bereits in der Einleitung angemerkt stellt Eric Thiémond in seinem Artikel [Thi01b] zwar bereits die Vermutung auf, dass die Berechnung des Volumens der größten Box, welche eine gegebene Anzahl von Punkten enthält,  $\mathcal{NP}$ -vollständig ist, erwähnt jedoch auch, dass er dieses nicht zu beweisen vermochte. Diese Lücke können wir an dieser Stelle schließen. Erstaunlicherweise können wir im Fall der  $(k^-, \varepsilon^+)$ -Boxen bereits für fest vorgegebenes  $k \in \mathbb{N}_0$  zeigen, dass die Frage nach dem Volumen der größten (halboffenen) Box, welche höchstens  $k$  Punkte enthält,  $\mathcal{NP}$ -vollständig ist.

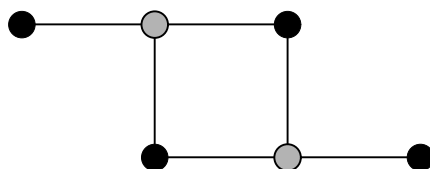
Wir werden im Folgenden zunächst beweisen, dass das  $(0^-, \varepsilon^+)$ -BOXEN-Problem in  $\mathcal{NPC}$  liegt und daraus folgern, dass dieses schon für alle  $k \in \mathbb{N}_0$  gelten muss. Dazu definieren wir zunächst einen weiteren Begriff, der in der Graphentheorie von großer Bedeutung ist:

### 3.2.1.3 Definition (dominierende Menge)

Ist  $G = (V, E)$  ein Graph und  $T \subseteq V$ , so heißt  $T$  dominierende Menge (dominating set) von  $G$ , falls für alle  $v \in V \setminus T$  ein  $t \in T$  mit  $\{v, t\} \in E$  existiert.

### 3.2.1.4 Beispiel einer dominierenden Menge

Im folgenden Beispiel ist die Menge der grau unterlegten Knoten eine dominierende Menge des dargestellten Graphen.



### 3.2.1.5 Optimierungsproblem (DOMINATING SET)

Eingabe : Graph  $G = (V, E)$   
 Frage : Welche Mächtigkeit hat die kleinste dominierende Menge von  $G$ ?

**3.2.1.6 Entscheidungsproblem (DOMINATING SET)**

Eingabe : Graph  $G = (V, E), m \in \mathbb{N}_{\leq |V|}$

Frage : Gibt es eine dominierende Menge  $T$  von  $G$  mit  $|T| \leq m$ ?

**3.2.1.7 Satz (DOMINATING SET  $\in \mathcal{NPC}$ )**

Da dieses ein wohlbekanntes Resultat der Komplexitätstheorie ist, verzichten wir an dieser Stelle auf einen Beweis und verweisen stattdessen auf [GJ79].

**3.2.1.8 Satz  $((0^-, \varepsilon^+) - \text{BOX} \in \mathcal{NPC})$** 

Das  $(0^-, \varepsilon^+)$ -BOXEN-Problem ist  $\mathcal{NP}$ -vollständig.

*Beweis.* • Die Existenz eines Verifikationsalgorithmus für  $(0^-, \varepsilon^+)$ -BOX mit polynomialer Laufzeit ist offensichtlich.

- Wir zeigen  $\text{DOMINATING SET} \leq_p (0^-, \varepsilon^+)$ -BOX:

Sei  $G = (V, E)$ ,  $k \in \mathbb{N}_{\leq |V|}$  eine Instanz von DOMINATING SET. Wir können ohne Beschränkung der Allgemeinheit annehmen, dass  $V = \mathbb{N}_{\leq n}$  für  $n = |V|$  gilt.

Nun setzen wir für alle  $i, j \in \mathbb{N}_{\leq n}$

$$x_j^i = \begin{cases} 1 - \frac{1}{2^{n+1}} & , \text{ falls } \{i, j\} \in E \text{ oder } i = j \\ \frac{1}{2^{n+1}} & , \text{ sonst} \end{cases}$$

und

$$x^i = (x_j^i)_{j=1}^n \quad \text{sowie} \quad X = (x^i)_{i=1}^n$$

und zeigen, dass in  $G$  genau dann eine dominierende Menge der Mächtigkeit  $\leq k$  existiert, wenn wir in  $[0, 1]^n$  eine von  $y \in \bar{\Gamma}(X)$  erzeugte, keinen Punkt der Folge  $X$  enthaltende Box  $[0, y[$  mit Volumen  $V_y \geq \left(1 - \frac{1}{2^{n+1}}\right)^k$  finden.

Zunächst halten wir jedoch fest, dass für die Längen  $|x_j^i|$  und  $|\varepsilon|$  der Binärkodierungen der Folgenglieder sowie des Wertes  $\varepsilon$  aufgrund der Gleichung

$$1 - \frac{1}{2^{n+1}} = \sum_{j=1}^{n+1} \frac{1}{2^j}$$

die Beziehungen  $|x_j^i| = O(n)$  und  $|\varepsilon| = \left| \left(1 - \frac{1}{2^{n+1}}\right)^k \right| = O(k \cdot n)$  gelten.

Insbesondere ist damit die angegebene Transformation von DOMINATING SET auf

$(0^-, \varepsilon^+)$ -BOX polynomiell.

Kommen wir nun zum Beweis der angegebenen Äquivalenz:

„ $\Rightarrow$ “: Sei  $T \subseteq V$  mit  $|T| \leq k$  so, dass für alle  $v \in V \setminus T$  ein  $t \in T$  mit  $\{v, t\} \in E$  existiert.

Setze nun

$$y_j = \left(1 - \frac{1}{2^{n+1}}\right) \cdot 1_T(j) + 1_{\mathbb{N}_{\leq n} \setminus T}(j), \quad j = 1, \dots, n$$

und  $y = (y_j)_{j=1}^n$ . Wir zeigen im Folgenden, dass  $[0, y[$  eine  $\left(0^-, \left[1 - \frac{1}{2^{n+1}}\right]^k\right)^+$ -Box ist und halten dazu zunächst fest, dass  $y \in \bar{\Gamma}(X)$  und

$$V_y = \prod_{j=1}^n y_j = \prod_{j \in T} \left(1 - \frac{1}{2^{n+1}}\right) = \left(1 - \frac{1}{2^{n+1}}\right)^{|T|} \geq \left(1 - \frac{1}{2^{n+1}}\right)^k$$

gilt.

Für alle  $i \in T$  ist  $x_i^i = 1 - \frac{1}{2^{n+1}} = y_i$  und zu jedem  $i \in \mathbb{N} \setminus T$  existiert ein  $t \in T$  so, dass  $\{i, t\} \in E$ , also  $x_i^i = 1 - \frac{1}{2^{n+1}} = y_t$  gilt. In beiden Fällen erhalten wir  $x^i \notin [0, y[$  und somit  $A(y, X) = 0$ .

„ $\Leftarrow$ “: Sei nun die Existenz einer  $\left(0^-, \left[1 - \frac{1}{2^{n+1}}\right]^k\right)^+$ -Box  $[0, y[$  angenommen.

Dann gilt  $|\{j \in \mathbb{N}_{\leq n} ; y_j \geq 1 - \frac{1}{2^{n+1}}\}| = n$ , da andernfalls aufgrund der Voraussetzung  $y \in \bar{\Gamma}(X)$  bereits

$$V_y = \prod_{j=1}^n y_j \leq \frac{1}{2^{n+1}} < 1 - \frac{n}{2^{n+1}} \leq \left(1 - \frac{1}{2^{n+1}}\right)^n \leq \left(1 - \frac{1}{2^{n+1}}\right)^k$$

gelten würde und analog erhalten wir für die Menge  $T$  aller Indizes  $j$  mit  $y_j = 1 - \frac{1}{2^{n+1}}$  die Ungleichung  $|T| \leq k$ .

Dabei haben wir im mittleren Schritt die für alle reellen Zahlen  $z \geq -1$  und alle  $m \in \mathbb{N}$  gültige Bernoullische Ungleichung  $1 + mz \leq (1 + z)^m$  ausgenutzt.

Wegen  $A(y, X) = 0$  gilt  $|T| \geq 1$ , und damit gilt für alle  $i \in \mathbb{N}_{\leq n}$  schon  $i \in T$  oder aber es existiert ein  $t \in T$  mit  $x_t^i = y_t = 1 - \frac{1}{2^{n+1}}$  und damit  $\{i, t\} \in E$ .

Also ist  $T$  eine dominierende Menge für  $G$  mit  $|T| \leq k$ .

□



### 3.2.1.9 Bemerkung

Obiger Beweis lässt sich ebenso mit der Setzung

$$x_j^i = \begin{cases} \alpha & , \text{ falls } \{i, j\} \in E \text{ oder } i = j \\ \beta & , \text{ sonst} \end{cases}$$

durchführen, solange  $\beta < \alpha^n$  gilt. Insbesondere kann  $\beta = 0$  gewählt werden. Es ist daher denkbar, dass sich auf diese Weise die Komplexität verwandter Probleme untersuchen lässt. Wir wenden uns nun jedoch dem allgemeinen  $(k^-, \varepsilon^+)$ -Boxen-Problem zu:

### 3.2.1.10 Korollar $((k^-, \varepsilon^+)$ -Box $\in \mathcal{NPC}$ )

Für alle  $k \in \mathbb{N}_0$  ist das  $(k^-, \varepsilon^+)$ -BOXEN-Problem  $\mathcal{NP}$ -vollständig und dieses gilt bereits für die Instanzen mit  $n = d$ .

*Beweis.* Sei  $k \in \mathbb{N}$ . Da wir im vorhergehenden Satz bereits eingesehen haben, dass das  $(0^-, \varepsilon^+)$ -BOXEN-Problem  $\mathcal{NP}$ -schwer ist, genügt es nun, dieses auf das allgemeine  $(k^-, \varepsilon^+)$ -BOXEN-Problem zu reduzieren:

Seien dazu  $n, d \in \mathbb{N}$ ,  $X = (x^i)_{i=1}^n$  eine Folge in  $[0, 1]^d$  und  $\varepsilon > 0$ .

Wir setzen  $\tilde{X} = (x^i)_{i=1}^{n+k}$ , wobei  $x^{n+i} = (0)_{j=1}^d$  für alle  $i \in \mathbb{N}_{\leq k}$  gelte.

Dann existiert offensichtlich genau dann eine  $(0^-, \varepsilon^+)$ -Box bezüglich  $X$ , wenn wir bezüglich  $\tilde{X}$  eine  $(k^-, \varepsilon^+)$ -Box finden.  $\square$

## 3.2.2 $\mathcal{NP}$ -Vollständigkeit des $(k^+, \varepsilon^-)$ -Boxen-Problems

### 3.2.2.1 Definition $((k^+, \varepsilon^-)$ -Box)

Seien  $d, n, k \in \mathbb{N}$ ,  $\varepsilon > 0$  und  $X = (x^i)_{i=1}^n$  in  $[0, 1]^d$ . Die von  $y \in \Gamma(X)$  erzeugte abgeschlossene Box  $[0, y]$  heißt  $(k^+, \varepsilon^-)$ -Box, wenn  $\bar{A}(y, X) \geq k$  und  $V_y \leq \varepsilon$  gilt.

Bevor wir nun das  $(k^+, \varepsilon^-)$ -BOXEN-Problem definieren werden, halten wir zunächst einmal fest, dass wir bei gegebener Folge  $X = (x^i)_{i=1}^n$  in  $[0, 1]^d$  zu festem  $k$  alle abgeschlossenen Boxen mit mindestens  $k$  Punkten bestimmen können, indem wir für jede der  $\binom{n}{k}$  Teilmengen  $T$  von  $\mathbb{N}_{\leq n}$  mit  $|T| = k$  die kleinste Box

$$B_T = \prod_{j=1}^d [0, \max\{x_j^t ; t \in T\}]$$

betrachten, die jeden der Punkte  $x^t$ ,  $t \in T$  enthält. Nun erhalten wir

$$V_{min}^k = \min \{ \lambda_d(B_T) ; T \subseteq \mathbb{N}_{\leq n}, |T| = k \}$$

und haben somit das Volumen  $V_{min}^k$  der kleinsten mindestens  $k$  Punkte enthaltenden Box mit polynomiellm Aufwand berechnen können.

Wir müssen daher das  $(k^+, \varepsilon^-)$ -BOXEN-Problem etwas anders definieren und den Wert  $k$  als Eingabewert vorgeben.

### 3.2.2.2 Entscheidungsproblem $((k^+, \varepsilon^-)$ -Box)

Eingabe :  $d, n \in \mathbb{N}$ , endliche Folge  $(x^i)_{i=1}^n \in [0, 1]^d$ ,  $\varepsilon > 0$ ,  $k \in \mathbb{N}$   
 Frage : Existiert eine  $(k^+, \varepsilon^-)$ -Box?

Wie zuvor werden wir auch hier die  $\mathcal{NP}$ -Vollständigkeit schon für die Instanzen mit  $d = n$  erhalten und werden für den Beweis eine Reduktion des CLIQUEN-Problems auf das  $(k^+, \varepsilon^-)$ -BOXEN-Problem angeben. Dazu benötigen wir zunächst noch einige Definitionen und Lemmata.

Der Übersichtlichkeit halber sei erwähnt, dass wir die Reduktion in mehreren Schritten vollziehen und insgesamt

$$\begin{aligned} \text{CLIQUE} &\leq_p \text{HALFCLIQUE} \\ &\leq_p \text{BALANCED COMPLETE BIPARTITE SUBGRAPH} \\ &\leq_p \text{BALANCED SUBGRAPH} \\ &\leq_p (k^+, \varepsilon^-) - \text{BOX} \end{aligned}$$

zeigen werden.

Die erste Ungleichung haben wir bereits in 3.1.14 eingesehen.

### 3.2.2.3 Definition (ausbalanzierter Subgraph)

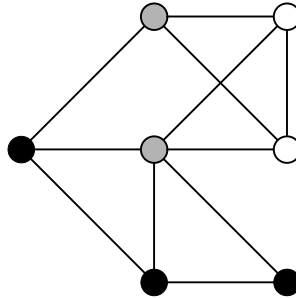
Ist  $G = (V, E)$  ein Graph,  $k \in \mathbb{N}_{\leq |V|}$  und sind  $I, J \subseteq V$  mit  $I \cap J = \emptyset$  sowie  $|I| = |J| = k$  so, dass für alle  $i \in I$  und  $j \in J$  schon  $\{i, j\} \in E$  gilt, so nennen wir das Paar  $(I, J)$  einen ausbalanzierten Subgraphen (engl.: balanced subgraph) der Größe  $k$ .

### 3.2.2.4 Bemerkung (vollständig bipartiter Graph)

Ist  $G = (V, E)$  ein Graph mit  $|V| = 2k$  Knoten, so ist  $G$  bei Existenz eines ausbalanzierten Subgraphen der Größe  $k$  gerade der vollständige bipartite Graph  $K_{k,k}$ .

### 3.2.2.5 Beispiel eines ausbalanzierten Subgraphen

Im unten stehenden Beispiel ist mit der Menge  $I$  der grauen Knoten und  $J$  der weißen Knoten das Paar  $(I, J)$  ein ausbalanzierter Subgraph der Größe 2.



### 3.2.2.6 Entscheidungsproblem (BALANCED SUBGRAPH)

Eingabe : Graph  $G = (V, E)$ ,  $k \in \mathbb{N}_{\leq |V|}$

Frage : Existiert in  $G$  ein ausbalanzierter Subgraph der Größe  $k$ ?

Wir werden zeigen, dass nicht nur das BALANCED SUBGRAPH-Problem, sondern auch sein bipartites Analogon  $\mathcal{NP}$ -vollständig ist:

### 3.2.2.7 Entscheidungsproblem (BALANCED COMPLETE BIPARTITE SUBGRAPH)

Eingabe : Biparter Graph  $G = (V_1 \dot{\cup} V_2, E)$ ,  $k \in \mathbb{N}_{\leq |V|}$

Frage : Existiert in  $G$  ein ausbalanzierter Subgraph der Größe  $k$ ?

Dieses Problem ist in [GJ79] formuliert worden und als Hinweis für einen Beweis wird das CLIQUEN-Problem angegeben. Den unten angegebenen Beweis hat David Johnson im Jahr 1987 in [Joh87] veröffentlicht, mit der Bemerkung, dass sich der Beweis der  $\mathcal{NP}$ -Vollständigkeit nicht wenigen, die sie zeigen wollten, entzogen hat. Den raffinierten Beweis wollen wir hier daher noch einmal angeben.

### 3.2.2.8 Satz (BALANCED COMPLETE BIPARTITE SUBGRAPH $\in \mathcal{NPC}$ )

Das Problem BALANCED COMPLETE BIPARTITE SUBGRAPH ist  $\mathcal{NP}$ -vollständig.

*Beweis.* • Dass BALANCED COMPLETE BIPARTITE SUBGRAPH  $\in \mathcal{NP}$  gilt, ist offensichtlich.

- BALANCED COMPLETE BIPARTITE SUBGRAPH ist  $\mathcal{NP}$ -schwer:

Sei  $G = (V, E)$ ,  $k = \frac{|V|}{2}$  eine Instanz von HALFCLIQUE. Definiere den Graphen  $G' = (V', E')$  durch

$$\begin{aligned} V' &= V \cup W \cup E \\ E' &= \{\{e, w\} ; e \in E, w \in W\} \cup \{\{e, v\} ; e \in E, v \in V, v \notin e\} \end{aligned}$$

und setze

$$k' = \binom{k}{2}$$

wobei  $W$  eine von  $V$  disjunkte Menge der Mächtigkeit  $\binom{k}{2} - k$  sei. Die Konstruktion von  $G'$  kann in polynomieller Zeit erfolgen. Offenbar ist  $G'$  ein bipartiter Graph mit den Partitionen  $E$  und  $V \cup W$ .

Es verbleibt zu zeigen, dass  $G$  genau dann eine Clique der Größe  $k$  enthält, wenn es in  $G'$  einen ausbalanzierten Subgraphen der Größe  $k'$  gibt.

„ $\Leftarrow$ “: Seien  $I, J \subseteq V'$  mit  $|I| = |J| = k'$  und  $\{i, j\} \in E'$  für alle  $i \in I, j \in J$ . Wir können ohne Beschränkung der Allgemeinheit annehmen, dass  $I \subseteq E$  und  $J \subseteq V \cup W$  gilt.

Da nun  $|W| \leq k' - k$  ist, folgt  $|V \cap J| \geq k = \frac{|V|}{2}$  und damit  $|V \setminus J| = |V \setminus (V \cap J)| \leq k$ . Da ferner  $|J| = k' = \binom{k}{2}$  gilt und nach Konstruktion von  $E'$  für alle  $e \in I$  bereits beide Endpunkte von  $e$  in  $V \setminus J$  liegen, erhalten wir damit, dass  $(V \setminus J, I)$  ein Subgraph von  $G$  mit höchstens  $k$  Knoten und  $\binom{k}{2}$  Kanten ist, was  $(V \setminus J, I) = K_k$  impliziert.

„ $\Rightarrow$ “: Sei  $(J, I)$  eine Clique der Größe  $k$  von  $G$ . Dann gilt für alle  $v \in W \cup (V \setminus J)$  und alle  $i \in I$  schon  $\{v, i\} \in E'$ . Da ferner

$$\begin{aligned} |W \cup (V \setminus J)| &= |W| + |V \setminus J| \\ &= \left( \binom{k}{2} - k \right) + |V| - \frac{|V|}{2} \\ &= \binom{k}{2} \\ &= |I| \\ &= k' \end{aligned}$$

gilt, bildet  $(W \cup (V \setminus J), I)$  einen ausbalanzierten Subgraphen der Größe  $k'$ .

□

Damit gilt selbstverständlich auch  $\text{BALANCED SUBGRAPH} \in \mathcal{NP}$  und wir können schließlich die  $\mathcal{NP}$ -Vollständigkeit des  $(k^+, \varepsilon^-)$ -BOXEN-Problems beweisen:

### 3.2.2.9 Satz $((k^+, \varepsilon^-)\text{-Box} \in \mathcal{NP})$

Das Problem  $(k^+, \varepsilon^-)$ -BOX ist  $\mathcal{NP}$ -vollständig.

*Beweis.* • Dass das  $(k^+, \varepsilon^-)$ -BOXEN-Problem in  $\mathcal{NP}$  liegt, ist wiederum leicht einzusehen.

- $(k^+, \varepsilon^-)$ -BOX ist  $\mathcal{NP}$ -schwer:

Sei  $G = (V, E)$ ,  $k \in \mathbb{N}$  eine Instanz von  $\text{BALANCED SUBGRAPH}$ . Wir können ohne Einschränkung annehmen, dass  $V = \mathbb{N}_{\leq n}$  gilt und setzen

$$x_j^i = \begin{cases} \alpha & , \text{ falls } \{i, j\} \in E \\ \beta & , \text{ sonst} \end{cases} ,$$

sowie

$$x^i = (x_j^i)_{j=1}^n, \quad i = 1, \dots, n.$$

Dabei seien  $\alpha, \beta \in ]0, 1[$  mit  $\alpha < \beta$ .

Wir zeigen nun, dass in  $G$  genau dann ein ausbalanzierter Subgraph der Größe  $k$  existiert, wenn es zu  $X := (x^i)_{i=1}^n$  eine  $(k^+, [\alpha^k \beta^{n-k}]^-)$ -Box gibt.

„ $\Rightarrow$ “: Seien  $I, J \subseteq V$  mit  $I \cap J = \emptyset$ ,  $|I| = |J| = k$  und  $\{i, j\} \in E$  für alle  $i \in I, j \in J$ . Setze

$$y_j := \alpha \cdot 1_J(j) + \beta \cdot 1_{V \setminus J}(j).$$

Dann gilt

$$\lambda_n([0, y]) = \alpha^k \beta^{n-k}$$

und

$$\bar{A}(y, X) = \sum_{i=1}^n 1_{[0, y]}(x^i) \geq \sum_{i \in I} 1_{[0, y]}(x^i) = k.$$

Damit ist  $[0, y[$  eine  $(k^+, [\alpha^k \beta^{n-k}]^-)$ -Box.

„ $\Leftarrow$ “: Sei  $y \in \Gamma(X)$  mit  $\lambda_n([0, y]) \leq \alpha^k \beta^{n-k}$  und  $\bar{A}(y, X) \geq k$ .

Dann gilt  $y_j \geq \alpha$  für alle  $j \in \mathbb{N}_{\leq n}$ , da andernfalls für alle  $i$  schon  $1_{[0, y]}(x^i) = 0$  gälte.

Zudem gilt aufgrund der Bedingung  $\lambda_n([0, y]) \leq \alpha^k \beta^{n-k}$  auch  $y_j < \beta$  für mindestens  $k$  Koordinaten  $j_1, \dots, j_k$ .

Nach Voraussetzung existieren paarweise verschiedene Indizes  $i_1, \dots, i_k$  mit  $x^{i_1}, \dots, x^{i_k} \in [0, y]$ . Dann gilt aber nach Konstruktion von  $X$  bereits  $\{i, j\} \in E$  für alle  $i \in \{i_1, \dots, i_k\}$  und  $j \in \{j_1, \dots, j_k\}$ . Zudem ist  $\{i_1, \dots, i_k\} \cap \{j_1, \dots, j_k\} = \emptyset$ , da aufgrund der Schlingenlosigkeit von  $G$  für alle  $j \in \{j_1, \dots, j_k\}$  schon  $x_j^j = \beta > \alpha$  gilt.

Aufgrund der Beliebigkeit der Wahl von  $\alpha$  und  $\beta$ , kann die hier vorgenommene Transformation polynomiell gewählt werden.  $\square$

Das in obigem Beweis konstruierte Gitter  $\bar{\Gamma}(X)$  hat eine Mächtigkeit von  $3^n$ . Im Allgemeinen ist die Mächtigkeit des Gitters  $\bar{\Gamma}(\hat{X})$  einer  $n$ -punktigen Folge  $\hat{X} = (\hat{x}^i)_{i=1}^n$  in  $[0, 1]^d$  gerade  $(n+1)^d$ , was im obigen Fall einer Mächtigkeit von  $(n+1)^n$  entsprechen würde. Auch wenn dieser Wert sichtlich größer ist als  $3^n$ , bleibt strenggenommen die Frage, ob das  $(k^+, \varepsilon^-)$ -BOXEN-Problem auch dann noch  $\mathcal{NP}$ -vollständig ist, wenn wir bei der Eingabe zusätzlich fordern, dass die Koordinaten paarweise verschieden sind. Dass wir diese Frage positiv beantworten können, zeigt der folgende

### 3.2.2.10 Satz ( $\mathcal{NP}$ -Vollständigkeit von $(k^+, \varepsilon^-)$ -Box bei Vorliegen paarweise verschiedener Koordinaten)

Da der Beweis der  $\mathcal{NP}$ -Vollständigkeit des  $(k^+, \varepsilon^-)$ -BOXEN-Problems genau wie in 3.2.2.9 durchgeführt werden kann, geben wir hier nur eine kurze Beweisskizze an:

*Beweis.* Sei zunächst  $G = (\mathbb{N}_{\leq n}, E)$ ,  $k \in \mathbb{N}$  eine Instanz von BALANCED SUBGRAPH und  $\varphi : \mathbb{N}_{\leq n} \times \mathbb{N}_{\leq n} \rightarrow \mathbb{N}_{\leq n^2}$  eine bijektive Abbildung. Wir können ohne Beschränkung der Allgemeinheit annehmen, dass  $n \geq 2$  gilt und setzen

$$x_j^i = \begin{cases} \frac{\varphi((i,j))}{2^{2nk}} & , \text{ falls } \{i, j\} \in E \\ 1 - \frac{\varphi((i,j))}{2^{2n}} & , \text{ sonst} \end{cases}$$

In  $G$  existiert genau dann ein ausbalanzierter Subgraph der Größe  $k$ , wenn wir bezüglich  $X = ((x_j^i)_{j=1}^d)_{i=1}^n$  eine  $(k^+, [(\frac{n^2}{2^{2nk}})^k (1 - \frac{1}{2^{2n}})^{n-k}]^-)$ -Box finden:

„ $\Rightarrow$ “ Mit den Bezeichnungen von 3.2.2.9 setze

$$y_j := \frac{n^2}{2^{2nk}} \cdot 1_J(j) + \left(1 - \frac{1}{2^{2n}}\right) \cdot 1_{V \setminus J}(j)$$

und zeige  $\bar{A}(y, X) \geq k$ .

„ $\Leftarrow$ “ Ist  $y \in \Gamma(X)$  mit

$$\lambda_n([0, y]) \leq \left(\frac{n^2}{2^{2nk}}\right)^k \left(1 - \frac{1}{2^{2n}}\right)^{n-k}$$

und  $\bar{A}(y, X) \geq k$ , so zeige, dass für mindestens  $k$  Koordinaten  $j_1, \dots, j_k$  schon

$$y_j \leq \frac{n^2}{2^{2nk}}$$

gelten muss, da andernfalls

$$\begin{aligned} \lambda_n([0, y]) &> \left(\frac{1}{2^{2nk}}\right)^{k-1} \left(1 - \frac{n^2}{2^{2n}}\right)^{n-k+1} \\ &> \left(\frac{n^2}{2^{2nk}}\right)^k \left(1 - \frac{1}{2^{2n}}\right)^{n-k} \end{aligned}$$

gälte, wobei sich die zweite Ungleichung aus

$$\begin{aligned} \left(1 - \frac{n^2}{2^{2n}}\right)^{n-k+1} &= \left(1 - \frac{n^2}{2^n \cdot 2^n}\right)^{n-k+1} \\ &\geq \left(1 - \frac{1}{2^n}\right)^{n-k+1} \\ &\geq \left(1 - \frac{1}{2^n}\right)^n \\ &\geq 1 - \frac{n}{2^n} \\ &\geq \frac{n}{2^n} \\ &> \left(\frac{n}{2^n}\right)^{2k} \\ &= \frac{1}{2^{2nk}} \cdot n^{2k} \end{aligned}$$

und der daraus folgenden Ungleichung

$$\left(\frac{1}{2^{2nk}}\right)^{k-1} \left(1 - \frac{n^2}{2^{2n}}\right)^{n-k+1} > \left(\frac{1}{2^{2nk}}\right)^{k-1} \cdot \frac{1}{2^{2nk}} \cdot n^{2k} = \left(\frac{1}{2^{2nk}}\right)^k \cdot n^{2k} = \left(\frac{n^2}{2^{2nk}}\right)^k$$

ergibt.

Nun kann man den Beweis wie in 3.2.2.9 zuende führen. □

### 3.3 Komplexitätstheoretische Betrachtung der Sterndiskrepanz

In diesem Abschnitt untersuchen wir die Komplexität der Berechnung der Sterndiskrepanz, die ja im Zentrum unserer Betrachtungen steht.

### 3.3.1 Definition der Probleme STERNDISKREPANZ

#### 3.3.1.1 Entscheidungsproblem (STERNDISKREPANZ)

Eingabe :  $d \in \mathbb{N}, n \in \mathbb{N}$ , endliche Folge  $X = (x^i)_{i=1}^n$  in  $[0, 1]^d, \varepsilon > 0$   
 Frage : Gilt  $D_n^*(X) \geq \varepsilon$  ?

Da wir in 2.2.2 bereits eingesehen haben dass

$$D_n^*(X) = \max \left\{ \max_{y \in \bar{\Gamma}(X)} V_y - \frac{1}{n} A(y, X), \max_{y \in \Gamma(X)} \frac{1}{n} \bar{A}(y, X) - V_y \right\}$$

gilt, können wir das Problem umformulieren zu:

#### 3.3.1.2 Entscheidungsproblem (STERNDISKREPANZ)

Eingabe :  $d \in \mathbb{N}, n \in \mathbb{N}$ , endliche Folge  $X = (x^i)_{i=1}^n$  in  $[0, 1]^d, \varepsilon > 0$   
 Frage : Existiert ein  $y \in \bar{\Gamma}(X)$  mit  $V_y - \frac{1}{n} A(y, X) \geq \varepsilon$  oder  $\frac{1}{n} \bar{A}(y, X) - V_y \geq \varepsilon$  ?

Das dazugehörige Optimierungsproblem ist das folgende

#### 3.3.1.3 Optimierungsproblem (STERNDISKREPANZ)

Eingabe :  $d \in \mathbb{N}, n \in \mathbb{N}$ , endliche Folge  $X = (x^i)_{i=1}^n$  in  $[0, 1]^d$   
 Frage : Wie groß ist  $D_n^*(X)$  ?

Beschränken wir uns im Fall des Optimierungsproblems STERNDISKREPANZ die Menge der zulässigen Lösungen auf die Gitterpunkte  $y \in \bar{\Gamma}(X)$  (was nach 2.2.2 eine sinnvolle Einschränkung darstellt), so können wir in Definition 3.1.8  $p = \text{Id}_{\mathbb{N}}$  setzen.

Auch können wir in Polynomzeit erkennen, ob zu einer gegebenen Folge  $X$  ein  $d \in \mathbb{N}$  existiert so, dass  $X$  in  $[0, 1]^d$  liegt und ebenfalls in Polynomzeit lassen sich die Kosten  $\max \left\{ \frac{1}{n} \bar{A}(y, X) - V_y, V_y - \frac{1}{n} A(y, X) \right\}$  von  $y$  bezüglich  $X$  berechnen, so dass das Problem STERNDISKREPANZ ein  $\mathcal{NP}$ -Optimierungsproblem darstellt.

### 3.3.2 $\mathcal{NP}$ -Vollständigkeit der Berechnung von $D_n^*(X)$

Im vorhergehenden Unterkapitel haben wir die  $\mathcal{NP}$ -Vollständigkeit der  $(k^-, \varepsilon^+)$ -BOXEN- sowie der  $(k^+, \varepsilon^-)$ -BOXEN-Probleme eingesehen. Auf ähnliche Art und Weise gelingt uns



nun auch die Analyse der Komplexität des Problems der Berechnung von  $D_n^*(X)$ .

Wir beweisen zunächst den folgenden

### 3.3.2.1 Satz (STERNDISKREPANZ $\in \mathcal{N}\mathcal{P}\mathcal{C}$ )

Besonders interessant macht diesen Satz, dass man für die Berechnung der  $L_p$ -Diskrepanz bezüglich der im Ursprung verankerten Boxen, welche für eine Folge  $X = (x^i)_{i=1}^n$  in  $[0, 1]^d$  durch

$$D_n^p(X) = \left( \int_{[0,1]^d} \left| V_x - \frac{1}{n} A(x, X) \right|^p dx \right)^{\frac{1}{p}}$$

definiert ist, weiß, dass sie für den Wert  $p = 2$  mithilfe der Formel von Warnock (vgl. [Mat99]) in  $O(dn^2)$  Schritten berechnet werden kann. Ein alternativer Algorithmus wurde 1996 von K. Frank und S. Heinrich in [FH96] veröffentlicht. Dieser Algorithmus hat eine Laufzeit von  $O(n(\log(n))^{d-1})$ .

Jiří Matoušek merkt in seinem Buch [Mat99] an, dass für Werte  $p \neq 2$  keine Algorithmen polynomieller Laufzeit zur Berechnung der  $L_p$ -Diskrepanz bekannt sind. Da die Sterndiskrepanz als  $L_\infty$ -Diskrepanz aufgefasst werden kann (aus historischen Gründen schreibt man  $D_n^*(\cdot)$  anstelle von  $D_n^\infty(\cdot)$ ), werden wir gleich einsehen, dass die Suche nach einem effizienten Algorithmus zur Berechnung von  $D_n^p(\cdot)$  zumindest für  $p = \infty$  erfolglos verlaufen muss, falls  $\mathcal{P} \neq \mathcal{N}\mathcal{P}$ .

Auch Winker und Fang stellen in ihrem Artikel [WF97] die Frage nach der Komplexität der Berechnung der Sterndiskrepanz, ohne jedoch eine Vermutung zu äußern.

Zur technischen Vereinfachung setzen wir:

$$V_{min}^k := \inf \{ V_y ; y \in [0, 1]^d, \bar{A}(y, X) \geq k \}$$

und

$$V_{max}^k := \sup \{ V_y ; y \in [0, 1]^d, A(y, X) \leq k \}.$$

Beide Größen werden wir in Kapitel 4 noch ausführlicher untersuchen. Wir benötigen an dieser Stelle neben den beiden abkürzenden Schreibweisen noch die folgende (leicht einzusehende oder in 4.1.8 nachzulesende) Gleichheit

$$D_n^*(X) = \max \left\{ \max_{k=0, \dots, n-1} V_{max}^k - \frac{k}{n}, \max_{k=1, \dots, n} \frac{k}{n} - V_{min}^k \right\}.$$

Nun aber kommen wir zunächst zum Beweis des mehrfach angekündigten Satzes der  $\mathcal{N}\mathcal{P}$ -Vollständigkeit von STERNDISKREPANZ:

*Beweis (Satz 3.3.2.1):*

- Ist  $y \in \bar{\Gamma}(X)$ , so können wir die beiden Werte  $V_y - \frac{1}{n}A(y, X)$  sowie  $\frac{1}{n}\bar{A}(y, X) - V_y$  in  $O(nd)$  berechnen und erhalten somit  $\text{STERNDISKREPANZ} \in \mathcal{NP}$ .
- Um zu zeigen, dass  $\text{STERNDISKREPANZ}$  ein  $\mathcal{NP}$ -schweres Problem ist, führen wir wie in 3.2.1.8 eine Reduktion von  $\text{DOMINATING SET}$  auf  $\text{STERNDISKREPANZ}$  durch. Sei dazu  $G = (\mathbb{N}_{\leq n}, E)$ ,  $k \in \mathbb{N}_{\leq n}$  eine Instanz des Entscheidungsproblems  $\text{DOMINATING SET}$ . Wir können ohne Beschränkung der Allgemeinheit annehmen, dass  $n \geq 2$  und  $k < n$  gilt und setzen

$$x_j^i = \begin{cases} 1 - \frac{1}{2^{n+1}} & , \text{ falls } \{i, j\} \in E \text{ oder } i = j \\ 0 & , \text{ sonst} \end{cases}$$

Wir werden zeigen, dass bezüglich  $X = (x^i)_{i=1}^n$  genau dann ein  $y \in \bar{\Gamma}(X)$  mit

$$V_y - \frac{1}{n}A(y, X) \geq \left(1 - \frac{1}{2^{n+1}}\right)^k \quad \text{oder} \quad \frac{1}{n}\bar{A}(y, X) - V_y \geq \left(1 - \frac{1}{2^{n+1}}\right)^k$$

existiert, wenn wir in  $G$  eine dominierende Menge  $T$  mit  $|T| \leq k$  finden.

Wie wir bereits in 3.2.1.8 festgestellt haben, gilt für die Länge  $|x_j^i|$  der Binärkodierung der Folgenglieder  $|x_j^i| = O(n)$  und analog erhalten wir für die Eingabelänge von  $\varepsilon = \left(1 - \frac{1}{2^{n+1}}\right)^k$  die Beziehung  $|\varepsilon| = O(k \cdot n)$ , so dass die angegebene Transformation von  $\text{DOMINATING SET}$  auf  $\text{STERNDISKREPANZ}$  tatsächlich polynomiell ist.

Kommen wir nun zum Beweis der Äquivalenz:

„ $\Leftarrow$ “ : Es sei  $T$  eine dominierende Menge von  $G$  mit  $|T| \leq k$ . Wie in 3.2.1.8 zeige

$$V_{max}^0 \geq \left(1 - \frac{1}{2^{n+1}}\right)^{|T|} \geq \left(1 - \frac{1}{2^{n+1}}\right)^k$$

und erhalte

$$\begin{aligned} D_n^*(X) &\geq V_{max}^0 - \frac{0}{n} \\ &\geq \left(1 - \frac{1}{2^{n+1}}\right)^k. \end{aligned}$$

„ $\Rightarrow$ “ : Nun gelte  $D_n^*(X) \geq \left(1 - \frac{1}{2^{n+1}}\right)^k$ . Wir zeigen im Folgenden zunächst, dass in diesem Fall schon  $D_n^*(X) = V_{max}^0$  gelten muss:

1. Für alle  $l \in \mathbb{N}_{\leq n}$  gilt  $\frac{l}{n} - V_{min}^l < \left(1 - \frac{1}{2^{n+1}}\right)^k$ :

Ist  $l \in \mathbb{N}_{\leq n-1}$ , so gilt

$$\frac{l}{n} - V_{min}^l \leq \frac{n-1}{n} = 1 - \frac{1}{n} < 1 - \frac{k}{2^{n+1}} \leq \left(1 - \frac{1}{2^{n+1}}\right)^k \leq D_n^*(X).$$

Zudem gilt für das Volumen der kleinsten alle  $n$  Folgenglieder umfassende Box

$$V_{min}^n = \left(1 - \frac{1}{2^{n+1}}\right)^n$$

und damit

$$1 - V_{min}^n = 1 - \left(1 - \frac{1}{2^{n+1}}\right)^n < \left(1 - \frac{1}{2^{n+1}}\right)^k \leq D_n^*(X).$$

Hier geht neben der Ungleichung  $\frac{2n}{2^{n+1}} < 1$  wieder die Bernoullische Ungleichung ein:

$$\begin{aligned} 1 &< 2 - \frac{2n}{2^{n+1}} \\ &\leq \left(1 - \frac{n}{2^{n+1}}\right) + \left(1 - \frac{k}{2^{n+1}}\right) \\ &\leq \left(1 - \frac{1}{2^{n+1}}\right)^n + \left(1 - \frac{1}{2^{n+1}}\right)^k \end{aligned}$$

2. Für alle  $l \in \mathbb{N}_{\leq n-1}$  gilt  $V_{max}^l - \frac{l}{n} < \left(1 - \frac{1}{2^{n+1}}\right)^k$ :

Da wir  $n \geq 2$  annehmen konnten, gilt

$$V_{max}^l \leq 1 - \frac{1}{2^{n+1}}$$

für alle  $l \in \mathbb{N}_{\leq n-1}$ , da für jede halboffene  $V_{max}^l$ -Box  $[0, y[$  wenigstens ein Punkt im Komplement  $[0, 1]^n \setminus [0, y[$  liegen muss.

Damit gilt jedoch für jedes  $l \in \mathbb{N}_{\leq n-1}$  schon

$$V_{max}^l - \frac{l}{n} \leq \left(1 - \frac{1}{2^{n+1}}\right) - \frac{1}{n}$$

und wir erhalten aufgrund von

$$\begin{aligned} \left(1 - \frac{1}{2^{n+1}}\right) - \left(1 - \frac{1}{2^{n+1}}\right)^k &\leq \left(1 - \frac{1}{2^{n+1}}\right) - \left(1 - \frac{1}{2^{n+1}}\right)^n \\ &\leq \left(1 - \frac{1}{2^{n+1}}\right) - \left(1 - \frac{n}{2^{n+1}}\right) \\ &= \frac{n-1}{2^{n+1}} \\ &< \frac{1}{n} \end{aligned}$$

die Ungleichung

$$V_{max}^l - \frac{l}{n} < \left(1 - \frac{1}{2^{n+1}}\right)^k \leq D_n^*(X).$$

Damit haben wir  $D_n^*(X) = V_{max}^0$  eingesehen und zeigen nun wie in 3.2.1.8, dass in  $G$  eine dominierende Menge  $T$  der Mächtigkeit  $|T| \leq k$  existieren muss:

Sei dazu  $y \in \bar{\Gamma}(X)$  so, dass  $[0, y[$  eine  $V_{max}^0$ -Box ist, die Box  $[0, y[$  also keinen Punkt der Folge  $X$  enthält und ein Volumen  $\lambda_n([0, y]) = V_{max}^0 = D_n^*(X) \geq \left(1 - \frac{1}{2^{n+1}}\right)^k$  aufweist.

Dann gilt

$$\left| \left\{ j \in \mathbb{N}_{\leq n} ; y_j \geq 1 - \frac{1}{2^{n+1}} \right\} \right| = n,$$

da andernfalls bereits

$$V_y = 0 < \left(1 - \frac{1}{2^{n+1}}\right)^k$$

gälte. Außerdem erhalten wir

$$\left| \left\{ j \in \mathbb{N}_{\leq n} ; y_j = 1 - \frac{1}{2^{n+1}} \right\} \right| \leq k$$

und wir können den Beweis wie in 3.2.1.8 zuende führen.

□

### 3.3.3 Nichtapproximierbarkeit von STERNDISKREPANZ

Wie bereits oben angesprochen stellt sich nun die Frage nach der Approximierbarkeit von STERNDISKREPANZ. Wir haben dazu den Begriff des  $\varepsilon$ -Algorithmus kennengelernt und werden nun zeigen, dass wir die Größe  $D_n^*(X)$  in polynomieller Zeit nicht bis auf einen vorgegebenen Fehler  $\varepsilon > 1$  approximieren können.

#### 3.3.3.1 Satz (Nichtapproximierbarkeit von STERNDISKREPANZ in polynomieller Zeit)

Gilt  $\mathcal{P} \neq \mathcal{NP}$ , so existiert kein Algorithmus  $\mathcal{A}$ , der für alle  $\varepsilon > 1$  und alle Instanzen  $X$  von STERNDISKREPANZ in einer in der Eingabelänge der Instanz  $|X|$  polynomiell abhängigen Zeit einen Vektor  $\mathcal{A}_\varepsilon(I)$  mit Approximationsgüte  $\leq \varepsilon$  liefert.

Dieser Satz macht die Hoffnung auf eine Möglichkeit zur effizienten Berechnung der Sterndiskrepanz einer gegebenen Punktfolge zunichte.

*Beweis (Satz 3.3.3.1):* Um einen Widerspruchsbeweis zu führen treffen wir die Annahme der Existenz eines Polynoms  $p$  und eines Algorithmus  $\mathcal{A}$  für das Optimierungsproblem STERNDISKREPANZ, welcher uns zu jedem  $\varepsilon > 1$  und jeder Instanz  $X$  in  $p(|X|)$  Schritten einen Vektor  $\mathcal{A}_\varepsilon(X)$  mit Approximationsgüte  $\leq \varepsilon$  liefert und zeigen, dass dann schon  $\mathcal{P} = \mathcal{NP}$  gelten muss.

Sei dazu  $G = (V, E)$  eine Instanz des Optimierungsproblems DOMINATING SET. Wir können wieder ohne Beschränkung der Allgemeinheit annehmen, dass  $V = \mathbb{N}_{\leq n}$ ,  $n \geq 2$  sowie  $E \neq \emptyset$  gilt. Es sei  $T$  eine dominierende Menge kleinster Mächtigkeit von  $G$  und  $t := |T|$ . Dann gilt wegen  $|E| \geq 1$  schon  $t < n$ .

Wie in 3.3.2.1 konstruiere die Folge  $X = (x^i)_{i=1}^n$  in  $[0, 1]^n$ . Dann gilt, wie wir oben eingesehen haben,  $D_n^*(X) = \left(1 - \frac{1}{2^{n+1}}\right)^t$ .

Nun wähle  $\varepsilon \in \mathbb{R}_{>1}$  mit  $\varepsilon < \left(1 - \frac{1}{2^{n+1}}\right)^{-1}$ .

Da  $\mathcal{A}$  ein  $\varepsilon$ -Algorithmus und STERNDISKREPANZ ein Maximierungsproblem ist, erhalten wir für die Lösung  $\mathcal{A}(X, \varepsilon)$  die Ungleichung

$$\frac{D_n^*(X)}{\delta^*(\mathcal{A}(X, \varepsilon))} \leq \varepsilon,$$

wobei  $\delta^*$  wie folgt definiert sei:

$$\delta^* : [0, 1]^n \longrightarrow \mathbb{R}_{\geq 0}, y \longmapsto \max \left\{ V_y - \frac{1}{n}A(y, X), \frac{1}{n}\bar{A}(y, X) - V_y \right\}.$$

Dann muss insbesondere

$$\delta^*(\mathcal{A}(X, \varepsilon)) > \left(1 - \frac{1}{2^{n+1}}\right)^n$$

gelten, da andernfalls

$$\begin{aligned} \frac{D_n^*(X)}{\delta^*(\mathcal{A}(X, \varepsilon))} &\geq \left(1 - \frac{1}{2^{n+1}}\right)^{t-n} \\ &\geq \left(1 - \frac{1}{2^{n+1}}\right)^{-1} \\ &> \varepsilon \end{aligned}$$

gälte.

Also gilt

$$\delta^*(\mathcal{A}(X, \varepsilon)) = V_{\mathcal{A}(X, \varepsilon)} - \frac{1}{n}A(\mathcal{A}(X, \varepsilon), X),$$

da wir sonst

$$\begin{aligned} \delta^*(\mathcal{A}(X, \varepsilon)) &= \frac{1}{n}\bar{A}(\mathcal{A}(X, \varepsilon), X) - V_{\mathcal{A}(X, \varepsilon)} \\ &\leq \frac{1}{n}\bar{A}(\mathcal{A}(X, \varepsilon), X) - V_{\min}^{\bar{A}(\mathcal{A}(X, \varepsilon), X)} \\ &\leq \frac{n-1}{n} \\ &< \left(1 - \frac{1}{2^{n+1}}\right)^n \end{aligned}$$

erhielten. Die Gültigkeit der vorletzten Ungleichung haben wir in 3.3.2.1 eingesehen.

Zudem muss wegen

$$\delta^*(\mathcal{A}(X, \varepsilon)) \leq V_{\max}^{A(\mathcal{A}(X, \varepsilon), X)} - \frac{1}{n}A(\mathcal{A}(X, \varepsilon), X)$$

auch  $A(\mathcal{A}(X, \varepsilon), X) = 0$  gelten, denn für alle  $l \in \mathbb{N}_{\leq n-1}$  zeigt

$$\begin{aligned} \left(1 - \frac{1}{2^{n+1}}\right)^n &\geq 1 - \frac{n}{2^{n+1}} \\ &> 1 - \frac{1}{2^{n+1}} - \frac{1}{n} \end{aligned}$$

schon

$$V_{max}^l - \frac{l}{n} \leq 1 - \frac{1}{2^{n+1}} - \frac{1}{n} < \left(1 - \frac{1}{2^{n+1}}\right)^n < \delta^*(\mathcal{A}(X, \varepsilon))$$

(vgl. für die Gültigkeit der ersten Ungleichung auch 3.3.2.1).

Wir erhalten damit

$$\delta^*(\mathcal{A}(X, \varepsilon)) = V_{\mathcal{A}(X, \varepsilon)}.$$

Setze nun

$$y_j := (\min \{x_j \in \bar{\Gamma}(X)_j ; x_j \geq (\mathcal{A}(X, \varepsilon))_j\})_{j=1}^n$$

und erhalte wie in 3.3.2.1 die Gleichheit

$$\left| \left\{ j \in \mathbb{N}_{\leq n} ; y_j \geq 1 - \frac{1}{2^{n+1}} \right\} \right| = n.$$

Die Berechnung von  $y$  kann in  $O(n)$  erfolgen und es gilt

$$A(y, X) = A(\mathcal{A}(X, \varepsilon), X) = 0 \quad \text{sowie} \quad V_y \geq V_{\mathcal{A}(X, \varepsilon)}.$$

Nun erhalten wir schließlich für  $s = \left| \left\{ j \in \mathbb{N}_{\leq n} ; y_j = 1 - \frac{1}{2^{n+1}} \right\} \right|$  folgende Ungleichungen

$$\begin{aligned} \left(1 - \frac{1}{2^{n+1}}\right)^{-1} &> \varepsilon \\ &\geq \frac{D_n^*(X)}{V_{\mathcal{A}(X, \varepsilon)}} \\ &\geq \frac{\left(1 - \frac{1}{2^{n+1}}\right)^t}{V_y} \\ &= \left(1 - \frac{1}{2^{n+1}}\right)^{t-s}, \end{aligned}$$

was  $t = s$  zeigt. Damit ist  $\left\{ j \in \mathbb{N}_{\leq n} ; y_j = 1 - \frac{1}{2^{n+1}} \right\}$  eine dominierende Menge kleinster Mächtigkeit von  $G$ .

Mit der Wahl von  $\varepsilon = 1 + \frac{1}{2^{n+2}}$  sehen wir, dass die Transformation von DOMINATING SET in STERNDISKREPANZ polynomiell in  $n$  gewählt werden kann. Da auch  $\mathcal{A}$  sowie der oben angewendete Rundungsalgorithmus polynomielle Laufzeit haben, haben wir somit einen polynomiellen Lösungsalgorithmus für DOMINATING SET gefunden. Ein solcher kann aber wegen  $\text{DOMINATING SET} \in \mathcal{N}\mathcal{P}\mathcal{C}$  nicht existieren, sofern  $\mathcal{P} \neq \mathcal{N}\mathcal{P}$  gilt.  $\square$

# Kapitel 4

## Berechnung der Sterndiskrepanz über $k$ -Boxen

Wie bereits in der Einleitung erwähnt, wollen wir in diesem Kapitel im Wesentlichen auf die von Eric Thiémond in [Thi01b] vorgestellte Arbeit eingehen.

Die Grundidee seines Ansatzes ist die Berechnung des Volumens der kleinsten abgeschlossenen sowie der größten halboffenen im Ursprung verankerten Boxen, welche eine vorgegebene Anzahl von Punkten enthalten.

Indem man die Volumenfunktion logarithmiert, lassen sich diese zu den schon in 3.2 beschriebenen Entscheidungsproblemen gehörigen Optimierungsprobleme als ganzzahlige lineare Programme formulieren.

Wir werden im ersten Teil mit der Vorstellung der von Thiémond definierten Begriffe beginnen und auf Probleme hinweisen, die diese Definitionen mit sich führen. Um diese Probleme zu umgehen, werden wir dann die neuen Begriffe der  $k^+$ - sowie der  $k^-$ -Boxen einführen und deren Nützlichkeit für die Berechnung der Sterndiskrepanz darstellen.

In einem weiteren Unterkapitel werden wir wie oben angegeben die ganzzahligen linearen Programme formulieren und einige Hinweise zu ihrer Lösung geben.

Dabei wird es uns gelingen auf einige Ganzzahligkeitsbedingungen zu verzichten, was die Komplexität der Programme ein wenig verringert. Zudem werden wir die von Thiémond geforderte Bedingung  $x_j^i \neq 0$ ,  $1 \leq i \leq n$ ,  $1 \leq j \leq d$ , vermeiden können.

In einem dritten Schritt wollen wir dann darstellen, wie man mittels der vorgestellten Methoden die Sterndiskrepanz einer gegebenen Folge berechnen kann.

## 4.1 Die $k$ -Boxen-Probleme

### 4.1.1 Definition ( $k$ -Box)

Ist  $k \in \mathbb{N}$  und  $y \in [0, 1]^d$ , so nennen wir  $[0, y]$  eine abgeschlossene  $k$ -Box, falls  $\bar{A}(y, X) = k$  gilt. Im Fall  $A(y, X) = k$  heißt  $[0, y[$  halboffene  $k$ -Box.

### 4.1.2 Lemma (Berechnung von $D_n^*(X)$ über $k$ -Boxen)

Setzen wir

$$\tilde{V}_{min}^k = \inf \{V_y ; y \in [0, 1]^d, \bar{A}(y, X) = k\}$$

und

$$\tilde{V}_{max}^k = \sup \{V_y ; y \in [0, 1]^d, A(y, X) = k\},$$

so gilt

$$D_n^*(X) = \max \left\{ \max_{k=0, \dots, n-1} \tilde{V}_{max}^k - \frac{k}{n}, \max_{k=1, \dots, n} \frac{k}{n} - \tilde{V}_{min}^k \right\}.$$

*Beweis.* Beachte zunächst, dass  $\tilde{V}_{min}^0 = 0$  und  $\tilde{V}_{max}^n = 1$  gilt. Diese beiden Volumina brauchen daher wegen  $0 - \tilde{V}_{min}^0 = 0 = \tilde{V}_{max}^n - \frac{n}{n}$  nicht weiter beachtet zu werden.

Die Behauptung folgt nun aus Lemma 2.2.2, in welchem wir

$$D_n^*(X) = \max \left\{ \max_{y \in \bar{\Gamma}(X)} V_y - \frac{1}{n} A(y, X), \max_{y \in \Gamma(X)} \frac{1}{n} \bar{A}(y, X) - V_y \right\}$$

eingesehen haben. □

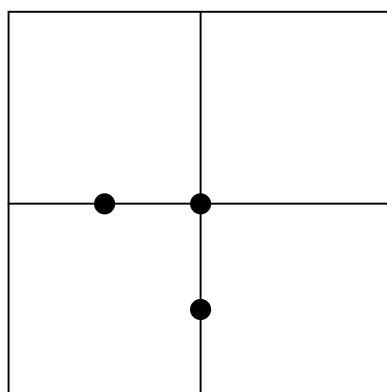
Kennen wir also für alle  $k \in \mathbb{N}_{\leq n}$  das Volumen der größten halboffenen  $k$ -Box sowie der kleinsten abgeschlossenen Box, die genau  $k$  Punkte enthält, so können wir die Sterndiskrepanz von  $X$  schnell über die oben angegebene Formel berechnen. Auch wenn sich manche der  $\tilde{V}_{min}^k$ - und  $\tilde{V}_{max}^k$ -Probleme schnell lösen lassen (einige Beispiele sind in 4.1.9 angegeben), haben wir in 3.2 bereits gesehen, dass die  $\tilde{V}_{min}^k$ - und  $\tilde{V}_{max}^k$ -Probleme  $\mathcal{NP}$ -schwere Probleme sind.

Dennoch ist Lemma 4.1.2 für die approximative Berechnung von  $D_n^*(X)$  nützlich. Es sei an dieser Stelle bereits auf 4.2 verwiesen, wo wir eine Methode zur Berechnung von oberen Schranken für  $D_n^*(X)$  kennenlernen werden. Wir halten jedoch zunächst einmal Folgendes fest:



### 4.1.3 Bemerkungen

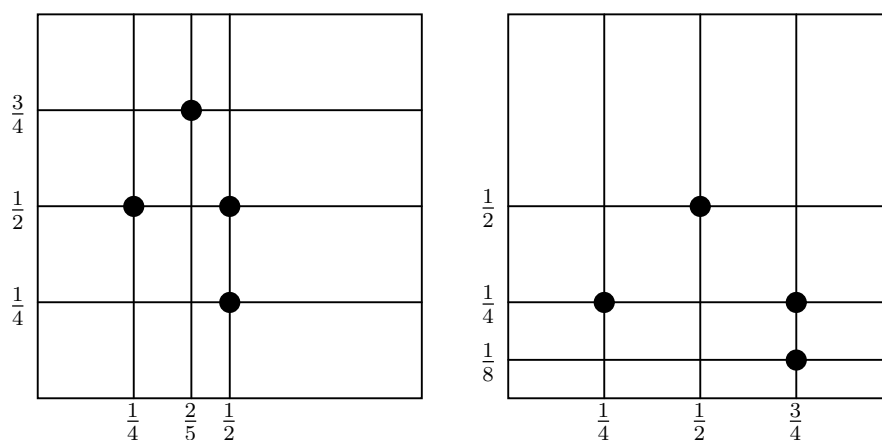
1. Im Allgemeinen existiert nicht für alle  $k \in \mathbb{N}$  eine Box, die genau  $k$  Punkte umfasst. Dieses gilt auch, anders als von Thiémond behauptet, für solche Folgen, in denen die Folgenglieder paarweise verschieden sind. Im unten angegebenen Beispiel etwa kann weder eine halboffene noch eine abgeschlossene Box mit genau 2 Punkten gefunden werden.



2. Wie die folgenden Beispiele zeigen, gelten die beiden von Thiémond fälschlicherweise als offensichtlich titulierten Ungleichungen

$$\tilde{V}_{min}^k \leq \tilde{V}_{min}^{k+1} \quad \text{und} \quad \tilde{V}_{max}^k \leq \tilde{V}_{max}^{k+1}$$

im Allgemeinen nicht:



Im linken Beispiel etwa gilt

$$\tilde{V}_{min}^3 = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4} < \frac{3}{10} = \frac{2}{5} \cdot \frac{3}{4} = \tilde{V}_{min}^2,$$

in der rechten, leicht verschobenen Situation erhalten wir

$$\tilde{V}_{max}^3 = \frac{1}{2} < \frac{3}{4} = \tilde{V}_{max}^2.$$

Wir sehen, dass in keinem der drei Fälle paarweise verschiedene Koordinaten vorliegen. Für solche Situationen können wir die folgenden positiven Resultate festhalten:

#### 4.1.4 Lemma (Situation bei Vorliegen paarweise verschiedener Koordinaten)

1. Ist  $|\Gamma_j(X)| = n$  für ein  $j \in \mathbb{N}_{\leq d}$ , so existiert für alle  $k \in \mathbb{N}_{\leq n}$  eine halboffene sowie eine abgeschlossene  $k$ -Box.
2. Gilt die Gleichung  $|\Gamma_j(X)| = n$  sogar für alle  $j \in \mathbb{N}_{\leq d}$ , so gelten die Ungleichungen

$$\tilde{V}_{min}^k < \tilde{V}_{min}^{k+1} \quad \text{und} \quad \tilde{V}_{max}^k < \tilde{V}_{max}^{k+1}$$

für alle  $k \in \mathbb{N}_{\leq n-1}$

*Beweis.* zu (1): Es sei  $X = (x_j^i)_{j=1}^n$  eine Folge in  $[0, 1]^d$  und  $k \in \mathbb{N}_{\leq n}$ . Ohne Beschränkung der Allgemeinheit gelte  $|\Gamma_1(X)| = n$  und  $x_1^1 < x_1^2 < \dots < x_1^n$ . Dann enthält die abgeschlossene Box

$$[0, x_1^k] \times \prod_{j=2}^n [0, \max\{x_j^i ; i \in \mathbb{N}_{\leq k}\}]$$

genau die  $k$  Punkte  $x^1, \dots, x^k$ , ebenso die halboffene Box

$$[0, x_1^{k+1}] \times \prod_{j=2}^n [0, \max\{\max\{x_j^i ; i \in \mathbb{N}_{\leq k}\}, \min\{x_j^i ; \forall l \in \mathbb{N}_{\leq k} : x_j^i > x_j^l\}\}].$$

zu (2): Für alle  $j \in \mathbb{N}_{\leq d}$  gelte  $|\Gamma_j(X)| = n$  und es sei  $k \in \mathbb{N}_{\leq n-1}$ . Ferner erzeuge  $y \in [0, 1]^d$  eine abgeschlossene  $(k+1)$ -Box mit  $V_y = \tilde{V}_{min}^{k+1}$ . Setzen wir

$$y'_j = \begin{cases} \max\{x_1^i \in \Gamma_1(X) ; x_1^i < y_1\} & , \text{ falls } j = 1 \\ y_j & , \text{ sonst} \end{cases}$$

und  $y' = (y'_j)_{j=1}^d$ , so gilt  $\bar{A}(y', X) = k$  und  $\tilde{V}_{min}^k \leq V_{y'} < V_y = \tilde{V}_{min}^{k+1}$ .

Die Aussage  $\tilde{V}_{max}^k < \tilde{V}_{max}^{k+1}$  lässt sich analog beweisen. □

Da wir das Vorliegen von Situationen wie in den oben angegebenen Beispielen 4.1.3 jedoch nicht von vorneherein ausschließen wollen, definieren wir nun:

#### 4.1.5 Definition ( $k^+$ -Box, $k^-$ -Box)

Die abgeschlossene Box  $[0, y]$  heißt  $k^+$ -Box, falls  $\bar{A}(y, X) \geq k$  gilt. In einer solchen Box liegen also mindestens  $k$  Punkte.

Liegen in der von  $y$  erzeugten halboffenen Box höchstens  $k$  Folgenglieder, d.h. gilt  $A(y, X) \leq k$ , so nennen wir  $[0, y[$  eine  $k^-$ -Box.

#### 4.1.6 Definition ( $V_{min}^k, V_{max}^k$ )

Wir definieren weiter

$$V_{min}^k = \inf\{V_y ; y \in [0, 1]^d, \bar{A}(y, X) \geq k\}$$

und

$$V_{max}^k = \sup\{V_y ; y \in [0, 1]^d, A(y, X) \leq k\}.$$

Zur technischen Vereinfachung definieren wir noch:

#### 4.1.7 Definition ( $V_{min}^k$ -, $V_{max}^k$ -Boxen)

Gilt  $V_{min}^k = \lambda_d([0, y])$ , so nennen wir  $[0, y]$  eine  $V_{min}^k$ -Box, wenn  $\bar{A}(y, X) \geq k$  ist. Im Fall  $V_{max}^k = \lambda_d([0, y[)$  und  $A(y, X) \leq k$  heißt die von  $y$  erzeugte halboffene Box  $V_{max}^k$ -Box.

Man beachte, dass mehrere  $V_{min}^k$ - und  $V_{max}^k$ -Boxen existieren können. Wiederum wie in 2.2.2 zeigt man, dass für eine  $V_{min}^k$ -Box  $[0, y]$  bereits  $y \in \Gamma(X)$  und für eine  $V_{max}^k$ -Box  $[0, y[$  schon  $y \in \bar{\Gamma}(X)$  gilt.

Dass diese Definitionen sinnvoll sind, zeigt das folgende Lemma:

#### 4.1.8 Lemma (Berechnung von $D_n^*(X)$ über $k^+$ - und $k^-$ -Boxen)

Es gilt

$$D_n^*(X) = \max \left\{ \max_{k=0, \dots, n-1} V_{max}^k - \frac{k}{n}, \max_{k=1, \dots, n} \frac{k}{n} - V_{min}^k \right\}.$$

*Beweis.* Da wir die Gleichheit für  $\tilde{V}_{min}^k$  und  $\tilde{V}_{max}^k$  bereits in 4.1.2 eingesehen haben, müssen wir nur die folgenden beiden Fälle betrachten:

1. In einer  $V_{min}^k$ -Box liegen mehr als  $k$  Punkte:

Dann gilt  $V_{min}^{k+1} = V_{min}^k$  und somit

$$D_n^*(X) \geq \frac{k+1}{n} - V_{min}^{k+1} > \frac{k}{n} - V_{min}^{k+1} = \frac{k}{n} - V_{min}^k.$$

Das  $V_{min}^k$ -Problem muss daher nicht weiter beachtet werden.

2. In einer  $V_{max}^k$ -Box liegen weniger als  $k$  Punkte:

Dann gilt  $V_{max}^{k-1} = V_{max}^k$  und wir erhalten

$$D_n^*(X) \geq V_{max}^{k-1} - \frac{k-1}{n} > V_{max}^{k-1} - \frac{k}{n} = V_{max}^k - \frac{k}{n}.$$

In diesem Fall muss das  $V_{max}^k$ -Problem nicht weiter beachtet werden.

□

#### 4.1.9 Beispiele einfach zu lösender $V_{min}^k$ - und $V_{max}^k$ -Probleme

1.  $V_{min}^1$ : Es gilt

$$V_{min}^1 = \min\{V_{x^i} ; 1 \leq i \leq n\}.$$

2.  $V_{min}^n$ : Da in dieser Box alle Punkte liegen müssen, gilt

$$V_{min}^n = \prod_{j=1}^d \max\{x_j^i ; 1 \leq i \leq n\}.$$

3.  $V_{min}^{n-1}$ : Zu jeder Koordinate  $j \in \mathbb{N}_{\leq d}$  betrachte, falls existent, das Volumen  $V_j$  der kleinsten Box, welche alle Punkte außer den mit der größten  $j$ -ten Koordinate umfasst und erhalte

$$V_{min}^{n-1} = \min\{ \min\{V_j ; 1 \leq j \leq d\} , V_{min}^n \}.$$

4.  $V_{max}^{n-1}$ : Zu jeder Koordinate  $j \in \mathbb{N}_{\leq d}$  berechne  $y'_j = \max\{x_j^i ; 1 \leq i \leq n\}$  und  $l$  so, dass  $y'_l = \max\{y'_j ; 1 \leq j \leq d\}$ . Nun setze

$$y_j = \begin{cases} y'_l & , \text{ falls } j = l \\ 1 & , \text{ sonst} \end{cases}$$

und erhalte

$$V_{max}^{n-1} = V_y.$$

## 4.2 Formulierung der LPs

In diesem Unterkapitel werden wir nun die Grundlage für ein Verfahren zur Berechnung von oberen Schranken von  $D_n^*(X)$  kennenlernen und dazu zunächst die Minimierungs- und Maximierungsprobleme aus 4.1.8 als ganzzahlige lineare Programme (LPs) darstellen und nach und nach einige Bemerkungen zur ihrer Lösung machen. Dazu benötigen wir zunächst ein paar Definitionen:

### 4.2.1 Koordinatenordnende Funktion

Für alle  $j \in \mathbb{N}_{\leq d}$  sei  $\sigma_j \in S_{n+1}$  so, dass

$$x_j^{\sigma_j(1)} \leq \dots \leq x_j^{\sigma_j(n)} \leq x_j^{\sigma_j(n+1)} = 1.$$

Zu jedem Index  $\delta = (\delta_1, \dots, \delta_d) \in (\mathbb{N}_{\leq n+1})^d$  definieren wir die von  $\delta$  erzeugten Boxen

$$[0, \delta] = \prod_{j=1}^d [0, x_j^{\sigma_j(\delta_j)}] \quad \text{sowie} \quad [0, \delta[ = \prod_{j=1}^d [0, x_j^{\sigma_j(\delta_j)}[.$$

### 4.2.2 Bemerkung und Definition ( $z_j^i(\delta)$ )

Offenbar gilt genau dann  $x^i \in [0, \delta]$  (bzw.  $x^i \in [0, \delta[$ ), wenn für alle  $j \in \mathbb{N}_{\leq d}$  die Ungleichung  $\sigma_j^{-1}(i) \leq \delta_j$  (bzw.  $\sigma_j^{-1}(i) \leq \delta_j - 1$ ) gilt. Da wir diese Beziehung im Folgenden häufig heranziehen werden, setzen wir

$$z_j^i(\delta) = \begin{cases} 1 & , \quad \text{falls } \sigma_j^{-1}(i) \leq \delta_j \\ 0 & , \quad \text{sonst} \end{cases}$$

Zu jedem  $\delta$  existiert genau eine Folge  $z = ((z_j^i(\delta)))_{j=1}^d_{i=1}^n$  in  $\{0, 1\}^{nd}$  und zu jeder solchen Folge  $z$  finden wir genau ein  $\delta(z) \in \mathbb{N}_{\leq n+1}^d$  mit  $z = z(\delta(z))$ . Ist aus dem Zusammenhang klar, welches  $z$  gemeint ist, schreiben wir der Übersichtlichkeit wegen auch  $\delta$  anstelle von  $\delta(z)$ .

### 4.2.3 Konvention

Um das k-Boxen-Problem als LP auszudrücken, werden wir die Volumina der betrachteten Boxen logarithmieren und setzen daher bis auf Weiteres voraus, dass für alle  $i \in \mathbb{N}_{\leq n}, j \in \mathbb{N}_{\leq d}$  die Bedingung  $x_j^i \neq 0$  gilt.

Nun sind wir in der Lage, das ganzzahlige LP zu formulieren. Es sei schon jetzt darauf hingewiesen, dass im folgenden LP genau dann  $y^i = 1$  gilt, wenn der Punkt  $x^i$  innerhalb der (abgeschlossenen)  $k^+$ -Box liegt, welche wir durch Lösen des LPs erhalten.

#### 4.2.4 Das $\ln(V_{min}^k)$ -Problem als ganzzahliges LP

$$\ln(V_{min}^k) = \min \sum_{j=1}^d [\ln(x_j^{\sigma_j(1)}) + \sum_{i=2}^n z_j^{\sigma_j(i)} (\ln(x_j^{\sigma_j(i)}) - \ln(x_j^{\sigma_j(i-1)}))] ]$$

s.t.

- (i)  $1 = z_j^{\sigma_j(1)} = \dots = z_j^{\sigma_j(k)} \geq \dots \geq z_j^{\sigma_j(n)} \quad \forall j \in \mathbb{N}_{\leq d}$
- (ii)  $z_j^{\sigma_j(i)} = z_j^{\sigma_j(i+1)} \quad \forall j \in \mathbb{N}_{\leq d} \forall i \in \mathbb{N}_{\leq n} : x_j^{\sigma_j(i)} = x_j^{\sigma_j(i+1)}$
- (iii)  $y^i \leq z_j^i \quad \forall j \in \mathbb{N}_{\leq d} \forall i \in \mathbb{N}_{\leq n}$
- (iv)  $y^i \geq 1 - d + \sum_{j=1}^d z_j^i \quad \forall i \in \mathbb{N}_{\leq n}$
- (v)  $\sum_{i=1}^n y^i \geq k$
- (vi)  $y^i \in \{0, 1\} \quad \forall i \in \mathbb{N}_{\leq n}$
- (vii)  $z_j^i \in \{0, 1\} \quad \forall i \in \mathbb{N}_{\leq n} \forall j \in \mathbb{N}_{\leq d}$

#### 4.2.5 Erläuterungen zur Formulierung 4.2.4

- Da in der zu findenden Box  $[0, \delta]$  mindestens  $k$  Punkte liegen, muss für alle  $j \in \mathbb{N}_{\leq d}$  die Ungleichung  $x_j^{\sigma_j(\delta_j)} \geq x_j^{\sigma_j(k)}$  gelten und so können wir die Werte  $1 = z_j^{\sigma_j(1)} = \dots = z_j^{\sigma_j(k)}$  fixieren.
- Bedingung (ii) besagt nur, dass im Fall zweier Punkte mit derselben  $j$ -ten Koordinate  $x_j^{\sigma_j(i)} = x_j^{\sigma_j(i+1)}$  auch  $z_j^{\sigma_j(i)} = z_j^{\sigma_j(i+1)}$  gelten muss. Denn in diesem Fall gilt entweder

$$x_j^{\sigma_j(\delta_j)} \leq x_j^{\sigma_j(i)} \quad \text{und} \quad x_j^{\sigma_j(\delta_j)} \leq x_j^{\sigma_j(i+1)}$$

oder aber

$$x_j^{\sigma_j(\delta_j)} > x_j^{\sigma_j(i)} \quad \text{und} \quad x_j^{\sigma_j(\delta_j)} > x_j^{\sigma_j(i+1)}.$$

- Bedingungen (iii) und (iv) drücken aus, dass genau dann  $y^i = 1$  gilt, wenn  $x^i \in [0, \delta]$  ist:  
Gilt nämlich  $x^i \in [0, \delta]$ , so ist  $\sigma_j^{-1}(i) \leq \delta_j$  und damit  $z_j^i = 1$  für alle  $j \in \mathbb{N}_{\leq d}$ . Nach Bedingung (iv) gilt dann bereits  $y^i \geq 1 - d + \sum_{j=1}^d z_j^i = 1$ , also  $y^i = 1$ .  
Gilt jedoch  $x^i \notin [0, \delta]$ , so existiert ein  $j \in \mathbb{N}_{\leq d}$  mit  $x_j^i > \delta_j$ . Dann ist aber  $z_j^i = 0$  und Bedingung (iii) liefert  $y^i \leq z_j^i = 0$ .

- Die Gleichung (v) sichert die Existenz von mindestens  $k$  Punkten innerhalb der von  $\delta$  erzeugten abgeschlossenen Box.
- Schließlich wird in (vi) und (vii) sichergestellt, dass sowohl die Variablen  $y^i$ ,  $i = 1, \dots, n$  als auch die Variablen  $z_j^i$ ,  $i = 1, \dots, n, j = 1, \dots, d$  nur binäre Werte annehmen: Ein Punkt  $x^i$  liegt entweder in  $[0, \delta]$  oder aber es gilt  $x^i \notin [0, \delta]$ .

Da die Lösung von binären LPs nicht einfach ist, kommt uns folgender Satz zugute, der uns die Relaxierung von  $nd$  Variablen erlaubt:

#### 4.2.6 Satz (Relaxierung der Bedingung $z_j^i \in \{0, 1\}$ )

Relaxieren wir in 4.2.4 die Bedingung (vii) zu

$$(vii') \quad z_j^i \in [0, 1] \quad \forall i \in \mathbb{N}_{\leq n} \forall j \in \mathbb{N}_{\leq d},$$

so ändert sich der Zielwert nicht, denn auch im so relaxierten LP gilt schon  $z_j^i \in \{0, 1\}$  für alle  $i = 1, \dots, n$ ,  $j = 1, \dots, d$ .

*Beweis.* Es sei  $z \in [0, 1]^{nd}$  eine zulässige Lösung des relaxierten  $\ln(V_{min}^k)$ -LPs, welche den Zielfunktionswert minimiert.

- Wir betrachten zunächst den Fall, dass für alle  $j \in \mathbb{N}_{\leq d}$  die  $j$ -ten Koordinaten der Punkte  $x^i$ ,  $i = 1, \dots, n$  paarweise verschieden sind und nehmen die Existenz eines Wertes  $z_j^{\sigma_j(i)} \in ]0, 1[$  an. Aufgrund von Bedingung (i) können wir ohne Beschränkung der Allgemeinheit  $z_j^{\sigma_j(i+1)} = 0$  annehmen.  
Nun liefert uns Bedingung (iii) die Ungleichung  $y^{\sigma_j(i)} \leq z_j^{\sigma_j(i)} < 1$ . Da  $y^{\sigma_j(i)}$  ganzzahlig ist, erhalten wir  $y^{\sigma_j(i)} = 0$  und wir ändern somit an der Zulässigkeit der Lösung  $z$  nichts, wenn wir  $z_j^{\sigma_j(i)} = 0$  setzen. Der Zielfunktionswert verringert sich dadurch (denn es gilt  $\ln(x_j^{\sigma_j(i)}) - \ln(x_j^{\sigma_j(i-1)}) > 0$ , dieser Summand fällt jedoch in der modifizierten Lösung weg) und wir erhalten somit ein Widerspruch zur Optimalität der Ursprungslösung.
- Im Fall nicht paarweise verschiedener  $j$ -ten Koordinaten ändert sich an obiger Argumentation nur etwas im Fall  $x_j^{\sigma_j(i)} = x_j^{\sigma_j(i-1)}$ . Dann gilt jedoch aufgrund von Bedingung (ii) mit  $z_j^{\sigma_j(i)} \in ]0, 1[$  auch  $z_j^{\sigma_j(i-1)} = z_j^{\sigma_j(i)} \in ]0, 1[$  und wie eben erhalten wir  $y^{\sigma_j(i-1)} = y^{\sigma_j(i)} = 0$  und wir können nicht nur  $z_j^{\sigma_j(i)} = 0$ , sondern auch  $z_j^{\sigma_j(i-1)} = 0$  sowie (mit derselben Begründung)  $z_j^{\sigma_j(l)} = 0$  für alle  $l \in \mathbb{N}_{\leq i}$  mit  $x_j^{\sigma_j(l)} = x_j^{\sigma_j(i)}$  setzen ohne die Zulässigkeit der Lösung zu verändern. Wieder ist jedoch der Ziel-

funktionswert der neu erhaltenen Lösung kleiner als zuvor und wir erhalten somit die Ganzzahligkeit aller  $z_j^i$ ,  $1 \leq i \leq n, 1 \leq j \leq d$ .

□

Es verbleiben noch die  $n$  Ganzzahligkeitsbedingungen  $y^i \in \{0, 1\}$ ,  $i = 1, \dots, n$ . Relaxiert man auch diese, so erhält man eine untere Schranke für den Wert  $V_{min}^k$ . Wie wir diese untere Schranken für die Berechnung der Sterndiskrepanz einer gegebenen Folge nutzen können, werden wir in 4.3 erläutern.

Bevor wir uns jedoch der Berechnung von  $D_n^*(X)$  zuwenden, wollen wir zunächst noch das ganzzahlige LP für die Maximierungsprobleme  $V_{max}^k$  vorstellen. Die erforderlichen Veränderungen an den Nebenbedingungen ergeben sich in erster Linie daraus, dass wir nunmehr keine abgeschlossenen, sondern halboffene  $k^-$ -Boxen  $[0, y[$  suchen. Dabei gilt im folgenden linearen Programm ähnlich wie in 4.2.4 genau dann  $y^i = 1$ , wenn der Punkt  $x^i$  in der von  $z$  erzeugten halboffenen Box  $[0, \delta(z)[$  liegt.

#### 4.2.7 Das $\ln(V_{max}^k)$ -Problem als ganzzahliges LP

$$\ln(V_{max}^k) = \max \sum_{j=1}^d [\ln(x_j^{\sigma_j(1)}) + \sum_{i=2}^{n+1} z_j^{\sigma_j(i)} (\ln(x_j^{\sigma_j(i)}) - \ln(x_j^{\sigma_j(i-1)}))] ]$$

s.t.

- (i)  $1 = z_j^{\sigma_j(1)} = \dots = z_j^{\sigma_j(k+1)} \geq \dots \geq z_j^{\sigma_j(n+1)} \quad \forall j \in \mathbb{N}_{\leq d}$
- (ii)  $z_j^{\sigma_j(i)} = z_j^{\sigma_j(i+1)} \quad \forall j \in \mathbb{N}_{\leq d} \forall i \in \mathbb{N}_{\leq n} : x_j^{\sigma_j(i)} = x_j^{\sigma_j(i+1)}$
- (iii)  $y^i \leq z_j^{\sigma_j(1+\sigma_j^{-1}(i))} \quad \forall j \in \mathbb{N}_{\leq d} \forall i \in \mathbb{N}_{\leq n}$
- (iv)  $y^i \geq 1 - d + \sum_{j=1}^d z_j^{\sigma_j(1+\sigma_j^{-1}(i))} \quad \forall i \in \mathbb{N}_{\leq n}$
- (v)  $\sum_{i=1}^n y^i \leq k$
- (vi)  $y^i \in \{0, 1\} \quad \forall i \in \mathbb{N}_{\leq n}$
- (vii)  $z_j^i \in \{0, 1\} \quad \forall i \in \mathbb{N}_{\leq n+1} \forall j \in \mathbb{N}_{\leq d}$

#### 4.2.8 Erläuterungen zur Formulierung 4.2.7

- Da für alle  $j \in \mathbb{N}_{\leq d}$  in der Box

$$\prod_{l=1, \dots, j-1} [0, 1[ \times [0, x_j^{\sigma_j(k+1)}[ \times \prod_{l=j+1, \dots, d} [0, 1[$$

höchstens  $k$  Punkte liegen, können wir die Werte  $1 = z_j^{\sigma_j(1)} = \dots = z_j^{\sigma_j(k+1)}$  fixieren.



- Wie oben bereits angekündigt, besagen Bedingungen (iii) und (iv) in Analogie zu den Bedingungen (iii) und (iv) des Minimierungsproblems 4.2.4, dass ein Punkt  $x^i$  genau dann in der Box  $[0, \delta[$  liegt, wenn  $y^i = 1$  gilt.  
Gilt  $x^i \in [0, \delta[$ , so ist  $\sigma_j^{-1}(i) < \delta_j$  und damit  $z_j^{\sigma_j(1+\sigma_j^{-1}(i))} = 1$  für alle  $j \in \mathbb{N}_{\leq d}$ . Nach Bedingung (iv) gilt somit  $y^i \geq 1 - d + \sum_{j=1}^d z_j^{\sigma_j(1+\sigma_j^{-1}(i))} = 1$ .  
Ist andererseits  $x^i \notin [0, \delta[$ , so existiert ein  $j \in \mathbb{N}_{\leq d}$  mit  $x_j^i \geq \delta_j$ . In diesem Fall erhalte  $z_j^{\sigma_j(1+\sigma_j^{-1}(i))} = 0$  und Bedingung (iii) liefert  $y^i \leq z_j^{\sigma_j(1+\sigma_j^{-1}(i))} = 0$ .
- Gleichung (v) gewährleistet, dass innerhalb der von  $\delta$  erzeugten Box höchstens  $k$  Punkte liegen.

### 4.2.9 Bemerkung

Leider gilt das Analogon von Satz 4.2.6 für das LP 4.2.7 nicht:

In einem solchen Satz für das LP 4.2.7 würde man im Fall von  $z_i^i \in ]0, 1[$  diesen Wert auf 1 aufrunden wollen, was wegen der restringierenden Bedingungen

$$z_j^{\sigma(i)} \geq z_j^{\sigma(i+1)}$$

und

$$0 = y^i \geq 1 - d + \sum_{j=1}^d z_j^{\sigma_j(1+\sigma_j^{-1}(i))}$$

nicht immer möglich sein wird.

Da die Lösung des relaxierten  $V_{max}^k$ -Problems jedoch sehr häufig ganzzahlig sind, vermeiden wir an dieser Stelle die Formulierung eines technisch aufwendigeren Analogons zu Satz 4.2.6.

Immerhin können wir ein paar zusätzliche Nebenbedingungen einführen:

1. Gilt  $\bar{A}(x^i, X) > k$ , so setze  $y^i = 0$ .
2. Falls  $\bar{A}(x^i, X) = k$  und  $x^p \notin [0, x^i]$ , kann dem LP die Restriktion  $y^i + y^p \leq 1$  hinzugefügt werden

Nachdem wir nun die Formulierung der LPs kennengelernt haben, können wir nun noch einmal auf Konvention 4.2.3 zurückkommen und die beiden LPs so modifizieren, dass auch das Vorliegen von  $x_j^i = 0$  zugelassen werden kann:

#### 4.2.10 Lemma (Modifikation der LPs bei Vorliegen von $x_j^i = 0$ )

1. Zu jedem  $j \in \mathbb{N}_{\leq d}$  definieren wir  $m_j = |\{i \in \mathbb{N}_{\leq n} ; x_j^i = 0\}|$  und erhalten für

$$k > \max\{m_j ; j \in \mathbb{N}_{\leq d}\}$$

die Zielwertfunktion

$$\begin{aligned} \ln(V_{min}^k) &= \min \sum_{j=1}^d [1_{\{m_j=0\}}(j) \ln(x_j^{\sigma_j(1)}) + 1_{\{m_j \neq 0\}}(j) \ln(x_j^{\sigma_j(m_j+1)})] \\ &\quad + \sum_{i=m_j+2}^n z_j^{\sigma_j(i)} (\ln(x_j^{\sigma_j(i)}) - \ln(x_j^{\sigma_j(i-1)})) \end{aligned}$$

An den Nebenbedingungen muss nichts geändert werden.

Man beachte, dass  $D_n^*(X) \geq \frac{\max\{m_j ; j \in \mathbb{N}_{\leq d}\}}{n}$  gilt, denn für

$$l \in \mathbb{N}_{\leq d} \quad \text{mit} \quad m_l = \max\{m_j ; j \in \mathbb{N}_{\leq d}\}$$

enthält die Box

$$\prod_{j=1}^{l-1} [0, 1] \times \{0\} \times \prod_{j=l+1}^d [0, 1]$$

schon  $m_l$  Punkte, hat aber ein Volumen von

$$\lambda_d \left( \prod_{j=1}^{l-1} [0, 1] \times \{0\} \times \prod_{j=l+1}^d [0, 1] \right) = 0.$$

Wir können daher für  $k \leq \max\{m_j ; j \in \mathbb{N}_{\leq d}\}$  auf die Formulierung eines  $\ln(V_{min}^k)$ -LPs verzichten.

2. Für das Maximierungsproblem betrachten wir zunächst den Fall, dass der Ursprung  $(0, \dots, 0)$  von jedem der Folgenglieder  $x^i$  verschieden ist. Dann gilt mit der Setzung von oben  $A(x^{\sigma_j(m_j+1)}, X) = 0$  sowie  $V_{x^{\sigma_j(m_j+1)}} > 0$  und wir können für alle  $k \in \{0, \dots, n-1\}$  die Zielwertfunktion umformulieren zu:

$$\begin{aligned} \ln(V_{max}^k) &= \max \sum_{j=1}^d [1_{\{m_j=0\}}(j) \ln(x_j^{\sigma_j(1)}) + 1_{\{m_j \neq 0\}}(j) \ln(x_j^{\sigma_j(m_j+1)})] \\ &\quad + \sum_{i=m_j+2}^n z_j^{\sigma_j(i)} (\ln(x_j^{\sigma_j(i)}) - \ln(x_j^{\sigma_j(i-1)})) \end{aligned}$$

3. Wird schließlich der Ursprung von wenigstens einem der Folgenglieder angenommen, so gilt mit  $a = |\{i \in \mathbb{N}_{\leq n} ; x^i = (0, \dots, 0)\}|$  für alle  $k \in \mathbb{N}_{\leq a}$  schon  $V_{max}^k(X) = 0$ . Für  $k = a+1, \dots, n$  erhalten wir  $V_{max}^k(X) = V_{max}^{k-a}(\tilde{X})$ , wobei  $\tilde{X}$  die Folge sei, die entsteht, wenn wir diejenigen Folgenglieder von  $X$  auslassen, welche den Wert des Ursprungs annehmen.

### 4.2.11 Bemerkung (Randomisiertes Runden)

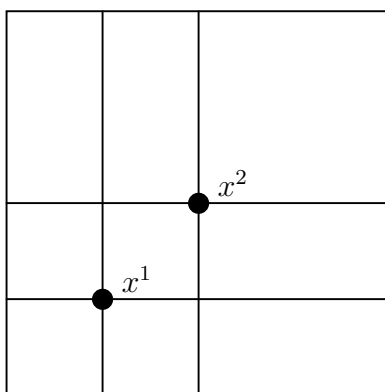
Ein zur näherungsweisen Lösung von ganzzahligen LPs gerne angewendetes Verfahren ist das randomisierte Runden, dem folgende Idee zugrunde liegt:

1. Löse das relaxierte LP und erhalte einen Lösungsvektor  $(y^1, \dots, y^n)$ .
2. Runde  $y^i$  mit Wahrscheinlichkeit  $y^i$  zum Wert 1 auf und mit Wahrscheinlichkeit  $1 - y^i$  zum Wert 0 ab und erhalte so einen ganzzahligen Vektor  $(\tilde{y}^1, \dots, \tilde{y}^n)$ . Dann ist der Vektor  $(\tilde{y}^1, \dots, \tilde{y}^n)$  das Ergebnis eines Zufallsexperiments mit  $n$  voneinander unabhängigen Zufallsgrößen  $Y^1, \dots, Y^n$ , für die  $P(Y^i = 1) = y^i$  und  $P(Y^i = 0) = 1 - y^i$  gilt.

Verschiedene Varianten dieses Verfahrens sichern in günstigen Situationen mit einer nach unten abschätzbaren Wahrscheinlichkeit ab, dass der auf diese Weise erhaltene Vektor  $(\tilde{y}^1, \dots, \tilde{y}^n)$  eine zulässige Lösung des ganzzahligen LPs ist und sein Zielfunktionswert nicht weit vom Optimum entfernt liegt.

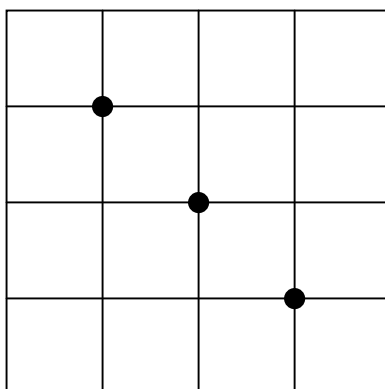
Es wäre wünschenswert, dieses Verfahren zur Lösung unserer LPs 4.2.4 und 4.2.7 zu verwenden, können es aber nicht effizient einsetzen, da wir die Werte der  $y^i$ 's nicht unabhängig voneinander runden können:

1. Das folgende einfache Beispiel zeigt, dass in diesem Fall mit  $y^2 = 1$  auch  $y^1 = 1$  gelten muss, unabhängig davon, ob ein  $V_{min}^k$ - oder ein  $V_{max}^k$ -Problem vorliegt:



Wir erhalten allgemein  $P(Y^p = 1 | Y^i = 1) = 1$  für alle  $p, i$  mit  $x^p \in [0, x^i]$ .

2. Das nun folgende Beispiel macht deutlich, dass die Abhängigkeiten noch erheblich komplexer sind:



Allgemein muss

$$P(Y^p = 1 | Y^{i_1} = 1, \dots, Y^{i_l} = 1) = 1$$

für alle  $p, i_1, \dots, i_l$  mit

$$x^p \in \prod_{j=1}^d [0, \max \{x_j^i ; i \in \{i_1, \dots, i_l\}\}]$$

gelten.

Wir müssen also einsehen, dass der Aufwand zur Berechnung dieser Abhängigkeiten zu hoch ist. Eine geeignete Alternative, welche auf derselben Idee fußt, bildet das von Thiémard in [Thi01b] erwähnte Variable Forcing:

#### 4.2.12 Variable Forcing

1. Löse das relaxierte LP und erhalte einen Lösungsvektor  $(y^1, \dots, y^n)$ .
2. Ist die Vermutung, dass eine bestimmte Variable  $y^i$  in der optimalen Lösung des ganzzahligen LPs einen bestimmten Wert  $v$  annimmt, so füge dem LP die Bedingung

$$y^i = 1 - v$$

zu.

3. Löse das modifizierte (und wiederum relaxierte) LP. Ist der so erhaltene Zielfunktionswert im größer [im Falle eines Maximierungsproblems kleiner] als eine heuristisch bestimmte obere [untere] Schranke des Problems, so kann  $y^i = v$  fixiert werden.

Zur Anwendung dieses Verfahrens benötigt man gute Heuristiken zur Berechnung von oberen Schranken für die  $V_{min}^k$ -, sowie zur Berechnung von unteren Schranken für die

$V_{max}^k$ -Probleme. Auf diese wollen wir im Folgenden jedoch nicht weiter eingehen und verweisen stattdessen erneut auf die Arbeit von Thiémarc. Zum Schluss dieses Kapitels wollen wir noch kurz erläutern, wie mit den oben beschriebenen Methoden die Sterndiskrepanz einer gegebenen Menge berechnet werden kann:

### 4.3 Berechnung der Sterndiskrepanz über $k$ -Boxen

In diesem Unterkapitel interessieren wir uns dafür, wie die vorgestellten Methoden bei gegebener Folge  $X$  zur Berechnung von  $D_n^*(X)$  herangezogen werden können. Da wir dazu angeben wollen, wie man vorgehen könnte, um die Diskrepanz von  $X$  zu berechnen, führen wir folgende, aus mathematischer Sicht sicherlich nicht korrekte, aber dennoch sehr hilfreiche Notationen ein:

#### 4.3.1 Notationen

Für alle  $k \in \mathbb{N}_{\leq n}$  bezeichne

$\bar{V}_{min}^k$  die kleinste bisher bestimmte obere Schranke von  $V_{min}^k$ ,

$\underline{V}_{min}^k$  die größte bisher bestimmte untere Schranke von  $V_{min}^k$ ,

$\bar{V}_{max}^k$  die kleinste bisher bestimmte obere Schranke von  $V_{max}^k$

und

$\underline{V}_{max}^k$  die größte bisher bestimmte untere Schranke von  $V_{max}^k$ .

Wir setzen weiter

$$\bar{D}_{min}^k(X) = \frac{k}{n} - \underline{V}_{min}^k,$$

$$\underline{D}_{min}^k(X) = \frac{k}{n} - \bar{V}_{min}^k,$$

$$\bar{D}_{max}^k(X) = \bar{V}_{max}^k - \frac{k}{n},$$

$$\underline{D}_{max}^k(X) = \underline{V}_{max}^k - \frac{k}{n}$$

und schließlich noch

$$\bar{D}_n^*(X) = \max \left\{ \max_{1 \leq k \leq n} \bar{D}_{min}^k(X), \max_{0 \leq k \leq n-1} \bar{D}_{max}^k(X) \right\}$$

sowie

$$\underline{D}_n^*(X) = \max \left\{ \max_{1 \leq k \leq n} \underline{D}_{min}^k(X), \max_{0 \leq k \leq n-1} \underline{D}_{max}^k(X) \right\}$$

und erhalten auf diese Art und Weise

$$D_n^*(X) \in [\underline{D}_n^*(X), \bar{D}_n^*(X)]$$

### 4.3.2 Berechnung von $D_n^*(X)$

#### 1. Schritt: Initialisierung

- (a) Mithilfe einer Kombination aus Greedy- und 2-opt-Algorithmen (vgl. [Thi01b]) bestimme für alle  $k \in \{0, \dots, n\}$  erste  $k^+$ -Boxen  $[0, y]$  sowie  $k^-$ -Boxen  $[0, y[$  und erhalte so eine obere Schranke  $\bar{V}_{min}^k$  für das  $V_{min}^k$ -Problem sowie eine untere Schranke  $\underline{V}_{max}^k$  für das  $V_{max}^k$ -Problem.

Beachte dabei, dass die Probleme  $V_{min}^1$ ,  $V_{min}^{n-1}$ ,  $V_{min}^n$  und  $V_{max}^{n-1}$ , wie in 4.1.9 beschrieben, bereits in kurzer Zeit gelöst werden können und wir nach der Durchführung der oben erwähnten Algorithmen noch

- $\bar{V}_{min}^k = \min\{V_{min}^{n-1}, \bar{V}_{min}^k\}$  für alle  $k \in \{2, \dots, n-2\}$
- $\underline{V}_{min}^k = \max\{V_{min}^1, \underline{V}_{min}^k\}$  für alle  $k \in \{2, \dots, n-2\}$
- $\bar{V}_{max}^k = \min\{V_{max}^{n-1}, \bar{V}_{max}^k\}$  für alle  $k \in \{0, \dots, n-2\}$

setzen können.

- (b) Gemäß 4.2.5 und 4.2.8 können wir weiter

$$\underline{V}_{min}^k = \max\left\{\prod_{j=1}^d x_j^{\sigma_j^{(k)}}, \underline{V}_{min}^k\right\}, \quad k = 2, \dots, n-2$$

sowie

$$\underline{V}_{max}^k = \max\left\{\prod_{j=1}^d x_j^{\sigma_j^{(k+1)}}, \underline{V}_{max}^k\right\}, \quad k = 0, \dots, n-2$$

setzen.

Die so erhaltenen Schranken für  $\underline{D}_{min}^k(X)$  und  $\underline{D}_{max}^k(X)$  sind laut Thiémond bereits sehr gut. Er belegt diese Behauptung mit Zahlen der von ihm durchgeführten Experimente.

#### 2. Schritt: (Approximative) Berechnung von $D_n^*(X)$

Da die soeben berechnete untere Schranke  $\underline{D}_n^*(X)$  für die Sterndiskrepanz von  $X$  sehr gut zu sein scheinen, konzentrieren wir uns im Folgenden auf die Verbesserung der oberen Schranke  $\bar{D}_n^*(X)$ .

Wähle dazu nacheinander dasjenige Problem, in welchem bei aktuellem Stand der Berechnungen der Wert  $\bar{D}_n^*(X)$  sein Maximum annimmt und löse das zugehörige relaxierte LP, falls das Problem zum ersten Mal näher betrachtet wird.

Wir erhalten so eventuell eine verbesserte untere Schranke  $\underline{V}_{min}^k$  für das  $V_{min}^k$ -Problem bzw. eine verbesserte obere Schranke  $\bar{V}_{max}^k$ , falls ein  $V_{max}^k$ -Problem vorliegt. In beiden Fällen kann in der Regel durch das Lösen des relaxierten LPs die obere Schranke  $\underline{D}_n^*(X)$  gesenkt werden. Ist die Lösung des relaxierten Problems ganzzahlig, so können wir  $\underline{D}_n^*(X)$  auf den Zielfunktionswert dieser Lösung anheben.

Ist das  $\bar{D}_n^*(X)$  verursachende Problem schon einmal betrachtet und das zugehörige relaxierte LP bereits gelöst worden, so lösen wir das Problem durch ein Branch & Bound-Verfahren, wie es in [Thi01b] beschrieben wird. Bei diesem Verfahren wird das zugrundeliegende  $V_{min}^k$ -Problem bzw.  $V_{max}^k$ -Problem vollständig gelöst und wir können wieder

- $\bar{V}_{min}^l = \min\{V_{min}^k, \bar{V}_{min}^l\}$  für alle  $l \in \{2, \dots, k-1\}$
- $\underline{V}_{min}^l = \max\{V_{min}^k, \underline{V}_{min}^l\}$  für alle  $l \in \{k+1, \dots, n-2\}$

bzw.

- $\underline{V}_{max}^l = \max\{V_{max}^k, \underline{V}_{max}^l\}$  für alle  $l \in \{k+1, \dots, n-2\}$
- $\bar{V}_{max}^l = \min\{V_{max}^k, \bar{V}_{max}^l\}$  für alle  $l \in \{0, \dots, k-1\}$

setzen.

So verfahren wir, bis wir entweder die Gleichheit  $\bar{D}_n^*(X) = \underline{D}_n^*(X)$  erreicht wurde, wir also den Wert  $D_n^*(X)$  genau berechnet haben, oder aber die Differenz  $\bar{D}_n^*(X) - \underline{D}_n^*(X)$  so klein ist, dass wir uns mit der Kenntnis zufrieden geben, dass  $D_n^*(X)$  im Intervall  $[\bar{D}_n^*(X), \underline{D}_n^*(X)]$  liegt.

### 4.3.3 Vor- und Nachteile dieses Verfahrens

Ein wesentlicher Vorteil ist zunächst, dass wir nicht nur untere Schranken von  $D_n^*(X)$  bestimmen können, wie es bei vielen anderen, unter anderem auch bei dem von uns später noch näher betrachteten Verfahren von Winker und Fang der Fall ist, sondern durch Relaxierung der LPs auch obere Schranken des Wertes  $D_n^*(X)$  berechnen.

Wir können das oben angegebene Verfahren beenden, wenn das berechnete Intervall  $[\bar{D}_n^*(X), \underline{D}_n^*(X)]$  so klein ist, dass wir an einer genaueren Bestimmung von  $D_n^*(X)$  nicht weiter interessiert sind. Dieses wird in der Praxis häufig der Fall sein, da man sich in der

Regel schon allein wegen der Rechnergenauigkeit mit der Bestimmung von  $D_n^*(X)$  bis zu einer gegebenen Genauigkeit zufrieden geben muss.

Erinnern wir uns noch einmal an die Methode 2.2.4 zur Approximation der Sterndiskrepanz, so haben wir dort ein Verfahren kennengelernt, welches ebenfalls Intervalle  $I$  berechnet, in denen die tatsächliche Diskrepanz einer gegebenen Folge liegen muss. Jedoch ist in dem dort angegebenen Verfahren vor Beginn der Konstruktion einer Partition zunächst festzulegen, wie groß die Fehlertoleranz  $\varepsilon = |I|$  sein darf. Für andere Vorgaben von  $\varepsilon$  muss das Verfahren erneut durchlaufen werden. Zudem steigt die Kardinalität der zu berechnenden Partition  $\mathcal{B}_\varepsilon$  mit kleiner werdenden  $\varepsilon$  erheblich.

Thiémard merkt in seinem Artikel [Thi01b] an, dass die Berechnung der Sterndiskrepanz über  $k$ -Boxen seiner Konstruktion aus 2.2.4 in der Praxis deutlich überlegen ist. Diese Überlegenheit macht sich besonders in höherdimensionalen Räumen bemerkbar.

Ein Nachteil ist die dennoch unübersichtliche zeitliche Komplexität des Verfahrens. Je größer  $n$  ist, desto mehr  $V_{min}^k$ - und  $V_{max}^k$ -Probleme müssen von uns betrachtet werden. Dass die Berechnung von  $D_n^*(X)$  in einer von  $n$  und  $d$  polynomiell abhängigen Zeit im Allgemeinen nicht möglich ist, haben wir in 3.3.2.1 bewiesen, mussten allerdings in 3.2 auch einsehen, dass dieses schon für die  $V_{min}^k$ - und  $V_{max}^k$ -Probleme selbst gilt.



# Kapitel 5

## Berechnung der Sterndiskrepanz mithilfe von Threshold Accepting

In diesem Kapitel stellen wir eine alternative Methode zur Berechnung der Sterndiskrepanz einer gegebenen Folge im  $d$ -dimensionalen Einheitswürfel vor. In Kapitel 3 haben wir einsehen müssen, dass die effiziente Berechnung dieser Größe für allgemeine Punktemengen nicht möglich ist und es stellt sich daher die Frage nach guten Heuristiken, die eine akzeptable Laufzeit aufweisen.

Wir werden uns hier näher mit dem Threshold Accepting beschäftigen, ein Verfahren, welches im Jahr 1990 von Dueck und Scheuer in [DS90] vorgestellt wurde und eine Abwandlung des Simulated Annealing (simulierte Abkühlung, siehe auch 5.1.2) darstellt.

Dazu werden wir zunächst das Verfahren näher beschreiben und kurz den schon angesprochenen Algorithmus von Winker und Fang aus [WF97] vorstellen, um diesen dann zu modifizieren und schließlich die Algorithmen miteinander zu vergleichen.

Dabei werden wir auch die sogenannten Good Lattice Points kennenlernen. Diese sind einfach zu konstruierende Folgen, die bei geeigneter Wahl der erzeugenden ganzzahligen Vektoren eine geringe Diskrepanz aufweisen.

### 5.1 Threshold Accepting

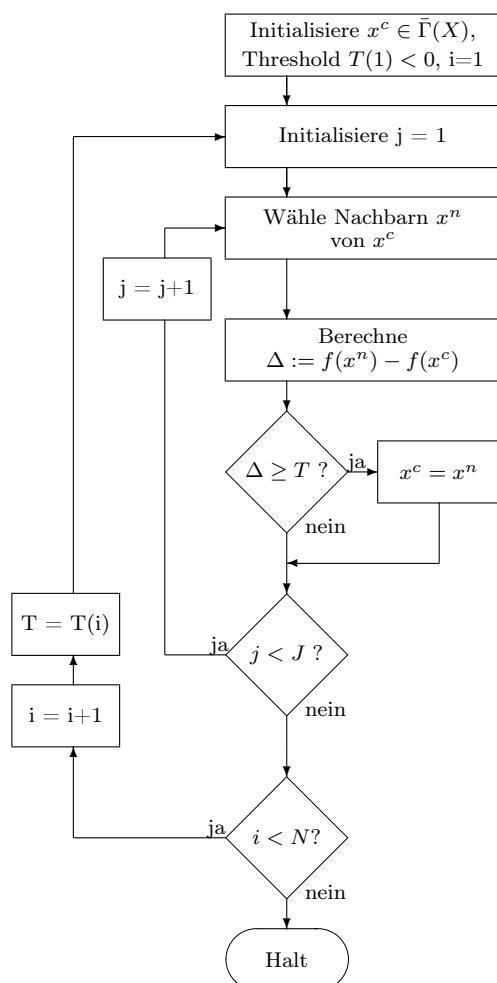
Threshold Accepting (TA) ist ein lokales Suchverfahren zur Optimierung einer Zielwertfunktion  $f$ , welches ausgehend von einem zufällig gewählten Startpunkt  $x^c$  in jeder Iteration zunächst einen zufälligen Punkt  $x^n$  der Nachbarschaft von  $x^c$  auswählt und den so erhaltenen Punkt genau dann als neuen Ausgangspunkt für die lokale Suche akzeptiert, wenn die

Abweichung der Zielwerte  $f(x^n) - f(x^c)$  der neuen von der alten Lösung nicht kleiner [bei Minimierungsproblemen größer] als ein vorzugebender Schwellenwert (engl.: threshold) ist. Die Schwellenwertfunktion ist dabei eine monoton gegen 0 strebende Funktion, monoton steigend bei Vorliegen eines Maximierungs- und monoton fallend im Falle eines Minimierungsproblems. Durch Akzeptanz auch solcher Nachbarn, welche eine hinsichtlich der Zielwertfunktion schlechtere Qualität aufweisen, können lokale Maxima bzw. Minima wieder verlassen werden, was bei klassischen lokalen Optimierungsverfahren nicht möglich ist.

### 5.1.1 Darstellung des Threshold Acceptings im Flussdiagramm

Eine gute Übersicht über den Ablauf des TAs bietet das folgende Flussdiagramm, welches den Verlauf des Verfahrens für Maximierungsprobleme darstellt.

In der hier vorgestellten Variante werden insgesamt  $I = J \cdot N$  Iterationen durchgeführt und der Schwellenwert  $T$  wird jeweils nach Durchführung von  $J$  Iterationen erhöht. Wir nennen  $N$  die Anzahl der äußeren Schleifen (engl.: outer loops) und  $J$  die Anzahl der inneren Schleifen (inner loops). Das Flussdiagramm veranschaulicht die Schleifenstruktur.



Zu spezifizierende Parameter sind neben der Anzahl der durchzuführenden Iterationen auch die Definition der Nachbarschaftsrelation, der Schwellenwertfunktion sowie der zugrundeliegenden Wahrscheinlichkeitsmaße. In der Anwendung des TAs auf das Sterndiskrepanzenproblem werden wir zwei verschiedene Spezifikationen dieser Größen kennenlernen.

### 5.1.2 Abgrenzung vom Verfahren des Simulated Annealing

Beim Verfahren des Simulated Annealing (SA) wird eine neue Lösung als Ausgangspunkt für die weitere lokale Suche akzeptiert, wenn ihre Qualität  $f(x^n)$  nicht schlechter ist als die ihres Vorgängers ist. Bei einer Verschlechterung des Zielfunktionswertes hingegen wird die neue Lösung nur mit einer bestimmten Wahrscheinlichkeit akzeptiert.

Diese Wahrscheinlichkeit ist zum einen abhängig vom Ausmaß der Verschlechterung, zum anderen aber auch von einem sogenannten Temperaturparameter. Dieser Temperaturparameter verändert sich mit jeder Iteration und bewirkt, dass die Wahrscheinlichkeit der Akzeptanz einer schlechteren Lösung mit der Anzahl der bereits durchgeführten Iterationen gegen 0 strebt.

In der Anwendung auf Maximierungsprobleme findet man häufig die Wahl von

$$\exp\left(-\frac{f(x^c) - f(x^n)}{\alpha^t \cdot T}\right)$$

als Akzeptanzwahrscheinlichkeit einer Nachbarschaftslösung  $x^n$  von  $x^c$  mit  $f(x^n) < f(x^c)$ . Dabei ist  $\alpha < 1$  die Kühlungskonstante und  $T$  ein fest gewählter Parameter („Temperatur“). Der Parameter  $t$  ist der Index der durchgeführten Iteration.

Man kann Threshold Accepting als deterministische Variante des SA verstehen, wenngleich zu beachten ist, dass die Wahl des Startpunktes sowie der Nachbarn nach wie vor zufällig erfolgt. Daher bietet sich der Komparativ „deterministischere“ Variante hier eher an.

### 5.1.3 Konvergenz des Threshold Acceptings

Althöfer und Koschnik konnten in [AK91] mithilfe von Markovkettentheorie zeigen, dass bei geeigneter Wahl der Strategieparameter das Verfahren des Threshold Acceptings gegen das globale Optimum konvergiert. Dabei nutzen sie die Kenntnis der Konvergenz des Simulated Annealings, welche bereits 1989 von Aarts und Korst in [AK89] gezeigt wurde. Leider ist dieser Beweis nicht konstruktiv, kann also keine Empfehlung für die Wahl der Strategieparameter geben.

Da die theoretische Analyse des Threshold Acceptings aufgrund der Zufallskomponente sehr schwierig ist (was sich nicht zuletzt darin äußert, dass es kaum Veröffentlichungen zur

theoretischen Untersuchung geeigneter Setzungen der Strategieparameter gibt), ist man bei praktischen Anwendungen auf vergleichende numerische Experimente angewiesen. Zum Vergleich mehrerer Algorithmen wird dabei neben der Qualität der besten im Verlauf des TA-Verfahrens besuchten Lösung auch der Mittelwert dieser bei Durchführung mehrerer Läufe berücksichtigt. Wir werden darauf in 5.4 näher eingehen.

#### 5.1.4 Anwendung von TA auf die Berechnung von $D_n^*(X)$

Die zur Berechnung der Sterndiskrepanz zu maximierende Zielwertfunktion ist die schon in 3.3.3.1 vorgestellte Funktion  $\delta^*$ , welche durch

$$\delta^* : [0, 1]^n \longrightarrow \mathbb{R}_{\geq 0}, \quad y \longmapsto \max \left\{ V_y - \frac{1}{n}A(y, X), \frac{1}{n}\bar{A}(y, X) - V_y \right\}$$

gegeben ist.

Alle anderen zu spezifizierenden Größen werden wir in den beiden Unterkapiteln 5.2 und 5.3 angeben.

## 5.2 Der Algorithmus von Winker und Fang

Nachdem wir nun den TA-Algorithmus in seiner Allgemeinheit kennengelernt haben, wollen wir ihn schließlich zur Berechnung der Sterndiskrepanz einsetzen. Eine mögliche Variante haben Winker und Fang in [WF97] untersucht. Wir wollen in diesem Unterkapitel die von ihnen vorgenommenen Spezifikationen vorstellen und beginnen mit der Definition der Nachbarschaften:

### 5.2.1 Die Nachbarschaftsrelation

Wir haben bereits im ersten Kapitel die von einer Folge  $X$  erzeugte Gitterstruktur  $\bar{\Gamma}(X)$  eingeführt und gesehen, dass die Diskrepanz  $D_n^*(X)$  in einem der Gitterpunkte angenommen wird. Diese Einsicht rechtfertigt eine Nachbarschaftsrelation, die sich an der Gitterstruktur orientiert.

In Analogie zu den koordinatenordnenden Funktionen aus 4.2.1 definieren wir nun für alle  $j \in \mathbb{N}_{\leq d}$  die Funktion  $\varphi_j$ , welche die Werte aus  $\bar{\Gamma}_j$  ordnet, dabei jedoch mehrfach vorkommende Werte ignoriert, d.h. es gilt  $\varphi_j \in \mathcal{F}(\mathbb{N}_{\leq n_j}, \bar{\Gamma}_j(X))$  mit

$$\varphi_j(1) < \varphi_j(2) < \dots < \varphi_j(n_j - 1) < \varphi_j(n_j) = 1,$$

wobei  $n_j = |\bar{\Gamma}_j|$  gesetzt sei.

Ist nun  $k \in \mathbb{N}$ , so definieren wir für alle  $j \in N_{\leq d}$  und  $y \in \bar{\Gamma}_j$  die Menge

$$\mathcal{N}_j^k(y) = \{ x_j \in \bar{\Gamma}_j ; \varphi_j(\max\{1, \varphi_j^{-1}(y) - k\}) \leq x_j \leq \varphi_j(\min\{n_j, \varphi_j^{-1}(y) + k\}) \}.$$

Sind schließlich  $mc \in N_{\leq d}$  und  $j_1, \dots, j_{mc} \in N_{\leq d}$ , so setzen wir weiter

$$\hat{\mathcal{N}}_{j_1, \dots, j_{mc}}^k(j, y) = \begin{cases} \mathcal{N}_j^k(y) & , \text{ falls } j \in \{j_1, \dots, j_{mc}\} \\ \{y_j\} & , \text{ sonst} \end{cases}$$

und

$$\mathcal{N}_{j_1, \dots, j_{mc}}^k(y) = \prod_{j=1}^d \hat{\mathcal{N}}_{j_1, \dots, j_{mc}}^k(j, y).$$

Die Menge  $\mathcal{N}_{j_1, \dots, j_{mc}}^k(y)$  wird mit dem Laplace-Wahrscheinlichkeitsmaß versehen, d.h. gegeben dass  $y$  der zur Zeit betrachtete Gitterpunkt ist, wird im nächsten Schritt ein Punkt  $x^n$  aus der Nachbarschaft von  $y$  wird mit einer Wahrscheinlichkeit von  $|\mathcal{N}_{j_1, \dots, j_{mc}}^k(y)|^{-1}$  ausgewählt.

Offenbar gilt  $|\mathcal{N}_{j_1, \dots, j_{mc}}^k(y)| \leq (2k+1)^{mc}$ . Winker und Fang zeigen in einer ihrer Analysen, dass die Verteilung der Abweichungen der Qualität eines Punktes von der seiner Nachbarn nur marginal von der Wahl des Parameters  $k$  abhängt.

Wird  $k$  sehr groß gewählt, ist die Wahrscheinlichkeit des Verharrens in einem lokalen Maximum geringer, die benötigte Anzahl der Iterationen zu einem relativ stabilen Zustand jedoch hoch.

Hier liegt ein „Trade-Off“ zwischen schnellen Verfahren mit kleinen Nachbarschaften und solchen mit besseren Ergebnissen und längerer Laufzeit vor. Dieser Trade-Off muss bei der Wahl des Strategieparameters berücksichtigt werden.

## 5.2.2 Wahl des Startpunktes

Auch der Startpunkt wird zufällig bezüglich des von  $\bar{\Gamma}(X)$  induzierten Laplace-Wahrscheinlichkeitsmaßes gewählt. Dazu ziehen wir für alle  $j \in N_{\leq d}$  eine Zahl  $i_j \in N_{\leq n_j}$  und setzen  $x^c = (\varphi_j(i_j))_{j=1}^d$ . Dabei ist für jedes  $j \in N_{\leq d}$  die Wahrscheinlichkeit, dass eine Zahl  $i_j$  gezogen wird, gerade  $\frac{1}{n_j}$ .

## 5.2.3 Die Schwellenwertfunktion

Wie im Flussdiagramm in Abschnitt 5.1.1 beschrieben, werden mit jedem Threshold  $T$  genau  $J$  Iterationen durchgeführt, bevor dieser Wert für eine erneute Durchführung von  $J$  Iterationen erhöht wird.

Winker und Fang setzen dabei  $J = \frac{\sqrt{I}}{\alpha}$ , wobei  $I$  die Anzahl der insgesamt zu durchlaufenden Iterationen und  $\alpha$  eine reelle Zahl zwischen 0 und 1 ist.

Zur Berechnung der demzufolge  $N = \alpha \cdot \sqrt{I}$  anzugebenden Thresholds werden zunächst  $\sqrt{I}$  zufällig gewählte Paare  $(x^i, y^i)$  mit  $x^i \in \bar{\Gamma}(X)$  und  $y^i \in \mathcal{N}_{j_1, \dots, j_{mc}}^k(x^i)$  berechnet und die Werte  $\{-|\delta^*(x^i) - \delta^*(y^i)| ; 1 \leq i \leq \sqrt{I}\}$  in aufsteigender Reihenfolge sortiert. Nun werden die  $\alpha \cdot \sqrt{I}$  größten Werte als Thresholds ausgewählt.

Die zugrunde liegenden Wahrscheinlichkeitsmaße für die Wahl der Werte  $j_1, \dots, j_{mc}$  sowie der Gitterpunkte  $x^i$  sind hier jeweils die zu  $\mathbb{N}_{\leq d}$  bzw.  $\bar{\Gamma}(X)$  gehörigen Laplace-Verteilungen. Die Parameter  $\alpha$ ,  $k$  und  $mc$  werden vor Beginn des Algorithmus festgesetzt. Eine Übersicht über die von Winker und Fang gewählten Werte für die Strategieparameter  $k$  und  $mc$  werden wir in 5.4 angeben. Die Wahl von  $\alpha$  kommentieren wir im folgenden Abschnitt.

### 5.2.3.1 Anmerkung zur Wahl der Thresholds

Auch wenn Winker und Fang sich durch die Wahl des Parameters  $\alpha$  einen positiven Effekt auf den Verlauf des Threshold Acceptings versprechen, konnten wir bei den von uns durchgeführten numerischen Experimenten keinen Einfluss dieses Parameters auf die Qualität der Ergebnisse feststellen, zumindest bei einer Wahl von  $\alpha$  im Intervall  $[0, 9; 1]$ . Die Experimente von Winker und Fang zeigen jedoch, dass nur eine solche Wahl von  $\alpha$  sinnvoll ist. Für kleinere Werte von  $\alpha$  verschlechtert sich die Qualität der Ergebnisse deutlich.

Wir haben daher in den unten angegebenen Ergebnissen stets  $\alpha = 0,995$  gesetzt.

## 5.3 Eine Alternative zum Algorithmus von Winker und Fang

In diesem Abschnitt wollen wir einen zweiten TA-Algorithmus zur Berechnung der Sterndiskrepanz vorstellen und werden dazu mehrere Veränderungen am Algorithmus von Winker und Fang vornehmen. Die einzelnen Alternativen werden wir schrittweise vorstellen und eine kurze Begründung für diese Modifikationen geben.

### 5.3.1 Variable Anzahl innerer Schleifen

Mit größerem Threshold  $T$  sinkt die Wahrscheinlichkeit, dass ein zufällig gewählter Nachbar  $x^n$  eines gegebenen Punktes  $x^c$  die Bedingung  $\delta^*(x^n) - \delta^*(x^c) \geq T$  erfüllt. Wir tragen dieser einfachen Beobachtung in unserem modifizierten Algorithmus Rechnung, indem wir die Anzahl der pro Threshold  $T(i)$  durchzuführenden Iterationen  $J(i)$ , die wir oben bereits als

„Inner Loops“ kennengelernt haben, mit der Anzahl der bereits durchgeführten Iterationen erhöhen.

Genauer setzen wir

$$J(i) = \begin{cases} \frac{T(1) \cdot I_{\max}}{100 \cdot T(i)} & , \text{ falls } \frac{T(1)}{100 \cdot T(i)} \leq 1 \\ I_{\max} & , \text{ sonst} \end{cases} ,$$

wobei  $I_{\max}$  eine vor Beginn des Threshold Acceptings festgesetzt werden muss und eine zu große Anzahl von Rechenschritten verhindert.

Man findet dieses Vorgehen in der Literatur unter anderem in [NP95]. Nissen und Paul wenden in diesem Artikel das Threshold Accepting auf das Quadratische Zuordnungsproblem an und zeigen durch numerische Experimente, dass die Durchführung eines längeren Laufes von Threshold Accepting der Durchführung mehrerer Durchläufe mit einer kleineren Anzahl von Iterationen überlegen ist. Diese Beobachtung werden durch die von uns durchgeführten Experimente bestätigt.

Ein Nachteil der oben beschriebenen Vorgehensweise der Zulassung einer variablen Anzahl von inneren Schleifen ist die Unvorhersehbarkeit der insgesamt durchgeführten Iterationen, welche sich jedoch nach oben immerhin ganz grob durch  $N \cdot I_{\max}$  abschätzen lässt. Unsere Experimente zeigen, dass die tatsächlich ausgeführte Anzahl von Iterationen in der Regel zwischen  $\frac{1}{8} \cdot N \cdot I_{\max}$  und  $\frac{1}{6} \cdot N \cdot I_{\max}$  liegt. Dabei bezeichne wieder  $N$  die Anzahl der äußeren Schleifen.

Führt man nun mehrere Tests durch, so stellt man zudem fest, dass die Anzahl  $\sum_{i \in \mathbb{N}_{\leq N}} J(i)$  der insgesamt durchgeführten Iterationen bei gegebener Punktemenge und Nachbarschaftsrelation trotz der zufälligen Wahl der Thresholds sehr stabil ist.

Wir werden später beim Vergleich der beiden in 5.2 und 5.3 vorgestellten Algorithmen  $I_{\max}$  jeweils so wählen, dass die Anzahl der in dieser Variante durchgeführten Iterationen von der im Algorithmus von Winker und Fang durchgeführten um weniger als fünf Prozent abweicht.

### 5.3.2 Die Nachbarschaftsrelation

Numerische Experimente bestätigen, dass folgende Modifikation an der Nachbarschaftsrelation ebenfalls einen positiven Effekt auf den Verlauf des Threshold Acceptings hat:

Betrachten wir zunächst noch einmal die von Winker und Fang gewählten Nachbarschaften, so stellen wir zwar fest, dass diese die Tatsache berücksichtigen, dass die Diskrepanz in einem der Gitterpunkte angenommen wird, jedoch keine weiteren Informationen über das Gitter selbst ausnutzt. Wie man leicht einsehen kann, hat jedoch auch der Abstand zweier Punkte  $x, y$  voneinander einen Einfluss auf die Differenz  $\delta^*(x) - \delta^*(y)$ . Wir führen daher

eine implizite Gewichtung der Punkte in der Nachbarschaft  $\mathcal{U}_{j_1, \dots, j_{mc}}^k(y)$  durch.

Für die Definition von  $\mathcal{U}_{j_1, \dots, j_{mc}}^k(y)$  wählen wir zunächst für alle  $j \in \mathbb{N}_{\leq d}$  wie in 5.2.1 die koordinatenordnende Funktionen  $\psi_j : \mathbb{N}_{\leq n_j+1} \rightarrow \bar{\Gamma}_j(X) \cup \{0\}$  mit

$$0 = \psi_j(1) < \psi_j(2) < \dots < \psi_j(n_j) < \psi_j(n_j + 1) = 1,$$

wobei in diesem Abschnitt  $n_j + 1 := |\bar{\Gamma}_j(X) \cup \{0\}|$  sei.

Nun betrachten wir für den gegebenen Wert  $mc \in \mathbb{N}_{\leq d}$  und für bezüglich des Laplace-Wahrscheinlichkeitsmaßes zufällig gewählte Elemente  $j_1, \dots, j_{mc} \in \mathbb{N}_{\leq d}$  die Nachbarschaft

$$\mathcal{U}_{j_1, \dots, j_{mc}}^k(y) = \prod_{j=1}^d \hat{\mathcal{U}}_{j_1, \dots, j_{mc}}^k(j, y)$$

mit

$$\hat{\mathcal{U}}_{j_1, \dots, j_{mc}}^k(j, y) = \begin{cases} \mathcal{I}(j, y), & \text{falls } j \in \{j_1, \dots, j_{mc}\} \\ \{y_j\} & \text{sonst} \end{cases}$$

und

$$\mathcal{I}(j, y) = [\psi_j(\max\{1, \psi_j^{-1}(y) - k\}), \psi_j(\min\{n_j + 1, \psi_j^{-1}(y) + k\})].$$

Die in dieser Umgebungen enthaltenen Gitterpunkte stimmen mit denen der Nachbarschaftsrelation von Winker und Fang überein, mit der Ausnahme, dass wir hier nunmehr zunächst auch solche Gitterpunkte betrachten, die in einer oder mehreren Koordinaten den Wert 0 aufweisen.

Einen weiteren Unterschied wird die Wahl des zugrundeliegenden Wahrscheinlichkeitsmaßes ausmachen. Diese wollen wir nun vorstellen:

Dazu definieren wir die Abbildung  $f : [0, 1] \rightarrow \mathbb{R}$ ,  $r \mapsto d \cdot r^{d-1}$  und versehen in einem ersten Schritt den Einheitswürfel  $[0, 1]^d$  mit dem polynomiellen Produktmaß

$$\pi_d(dx) = \otimes_{j \in \mathbb{N}_{\leq d}} f(x_j) \lambda(dx_j).$$

Warum wir dieses Maß betrachten, werden wir in 5.3.2.3 erläutern und verzichten aus Gründen der Übersichtlichkeit an dieser Stelle auf eine Begründung dieser Wahl.

Da nun die Nachbarschaft von  $y$  nicht der ganze Einheitswürfel  $[0, 1]^d$  ist, versehen wir  $\mathcal{U}_{j_1, \dots, j_{mc}}^k(y)$  mit dem von  $\pi_d$  induzierten Wahrscheinlichkeitsmaß. Zur Berechnung eines bezüglich diesen induzierten Maßes zufällig gewählten Punktes  $x^n$  der Nachbarschaft von  $y$  definieren wir zunächst die Funktion

$$\Psi(j, y) : \mathcal{I}(j, y) \rightarrow [0, 1], r \mapsto \frac{r^d - (\min \mathcal{I}(j, y))^d}{\max(\mathcal{I}(j, y))^d - (\min \mathcal{I}(j, y))^d}$$

mit Umkehrfunktion

$$\Psi^{-1}(j, y) : [0, 1] \rightarrow \mathcal{I}(j, y), s \mapsto \sqrt[d]{[(\max \mathcal{I}(j, y))^d - (\min \mathcal{I}(j, y))^d] \cdot s + (\min \mathcal{I}(j, y))^d}.$$



Nun ziehen wir für alle  $j \in \{j_1, \dots, j_{mc}\}$  eine Zufallszahl (bezüglich des Lebesgue-Maßes)  $s_j \in [0, 1]$  und setzen  $x_j^n = \Psi^{-1}(j, y)(s_j)$ . Für alle  $j \in \mathbb{N} \setminus \{j_1, \dots, j_{mc}\}$  setzen wir  $x_j^n = y_j$ .

Der so erhaltene Punkt  $x^n = (x_j^n)_{j=1}^d$  wird in der Regel kein Gitterpunkt sein. Wir führen daher nun folgenden Rundungsprozess durch, um die letztlich betrachteten Punkte  $x^{n,+}$  und  $x^{n,-}$  zu erhalten:

Wir setzen

$$x_j^{n,+} = \min\{x_j \in \bar{\Gamma}_j(X) ; x_j \geq x_j^n\}$$

und im Fall  $x_j^n \geq \min \Gamma_j(X)$

$$x_j^{n,-} = \max\{x_j \in \bar{\Gamma}_j(X) ; x_j \leq x_j^n\}.$$

Ist hingegen  $x_j^n < \min \Gamma_j(X)$ , so setzen wir

$$x_j^{n,-} = \max \Gamma_j(X).$$

Nun berechnen wir

$$\hat{\delta}(x^n) := \max\{\delta(x^{n,+}), \bar{\delta}(x^{n,-})\}.$$

Offensichtlich gilt  $A(x^{n,+}, X) = A(x^n, X)$  und damit

$$\delta(x^{n,+}) = V_{x^{n,+}} - A(x^{n,+}, X) \geq V_{x^n} - A(x^n, X) = \delta(x^n).$$

Zudem ist  $\bar{A}(x^{n,-}, X) = \bar{A}(x^n, X)$ , falls  $x_j^n \geq \min \Gamma_j(X)$ , und wir erhalten

$$\bar{\delta}(x^{n,-}) = \bar{A}(x^{n,-}, X) - V_{x^{n,-}} \geq \bar{A}(x^n, X) - V_{x^n} = \bar{\delta}(x^n).$$

Im Fall  $x_j^n < \min \Gamma_j(X)$  gilt wegen  $\bar{A}(x^n, X) = 0$  schon  $\bar{\delta}(x^n) < 0$ . Dann ist auch  $A(x^n, X) = 0$  und

$$\delta^*(x^n) = \delta(x^n) \leq \delta(x^{n,+}).$$

Zusammenfassend gilt somit

$$\hat{\delta}(x^n) \geq \delta^*(x^n),$$

was diese Rundungsprozedur rechtfertigt.

Ist nun  $x^c$  der im Threshold Accepting derzeitige betrachtete Ausgangspunkt für die lokale Suche, so wählen wir bei Vorliegen eines Schwellenwertes  $T$  im Fall  $\hat{\delta}(x^n) - \delta^*(x^c) \geq T$  den Punkt  $x^{n,+}$  als neuen Ausgangspunkt für die lokale Suche, falls  $\hat{\delta}(x^n) = \delta(x^{n,+})$ , den Punkt  $x^{n,-}$  andernfalls. Gilt hingegen  $\hat{\delta}(x^n) - \delta^*(x^c) < T$ , so wird die lokale Suche im Punkt  $x^c$  fortgesetzt.

Die Wahrscheinlichkeit eines Gitterpunktes  $x \in \mathcal{U}_{j_1, \dots, j_{mc}}^k(y)$  als Nachbar  $x^{n,+}$  von  $y$  ausgewählt zu werden, hängt somit anschaulich von seinem Abstand zu den anderen Gitterpunkten „links unterhalb“ seiner Position ab, die Wahrscheinlichkeit als Punkt  $x^{n,-}$  ausgewählt

zu werden, vom Abstand zu den Gitterpunkten „rechts oberhalb“ seiner Position. Wir haben somit eine Gewichtung der Punkte aus  $\mathcal{U}_{j_1, \dots, j_{mc}}^k(y)$  vorgenommen, ohne die Gewichte vorher berechnet haben zu müssen. Dieses Verfahren ist damit sehr rechenzeitfreundlich.

### 5.3.2.1 Wahl von $k$

Während Winker und Fang den Wert von  $k$  relativ stabil halten (vgl. 5.4.2), wählen wir stets  $k = \lfloor \frac{n}{8} \rfloor$ , sofern  $n \geq 100$ . Andernfalls setzen wir  $k = \lfloor \frac{n}{4} \rfloor$ . Für jede reelle Zahl  $r$  bezeichnet dabei  $\lfloor r \rfloor$  den ganzzahligen Anteil von  $r$ , i.e.  $\lfloor r \rfloor := \max\{p \in \mathbb{N} ; p \leq r\}$ .

Auch hier ist eine weitere Untersuchung der Wahl von  $k$  auf die Qualität des Ergebnisses sicher interessant. Wir geben uns hier mit der Tatsache zufrieden, dass wir in den Experimenten mit dieser Wahl die besten Ergebnisse erhalten haben. Zudem stehen die beiden Werte  $k$  und  $n$  enger zueinander in Beziehung als im Algorithmus von Winker und Fang.

### 5.3.2.2 Wahl von $mc$

Durch die Wahl von  $mc$  geben wir vor, in wie vielen Koordinaten die Werte des Nachbarn von denen der aktuellen Lösung Abweichungen aufweisen dürfen.

In niedrigdimensionalen Situationen setzen wir  $mc = \lfloor \frac{d}{2} \rfloor$ . In höherdimensionalen Räumen stellen wir bei unseren Experimenten fest, dass eine Wahl sehr kleiner Werte ( $mc \in \mathbb{N}_{\leq 5}$ ) durchweg bessere Ergebnisse liefert als die Wahl größerer Zahlen. Auch wenn wir  $k$  deutlich kleiner als oben angegeben wählen, tritt dieses Phänomen auf und lässt sich wie in 5.2.1 dadurch erklären, dass der Algorithmus bei dieser Wahl schneller in einen relativ stabilen Zustand gerät.

Es ist denkbar, dass der hier angegebene Algorithmus bei einer größeren Anzahl von Iterationen eine bessere Qualität aufweist, wenn  $mc$  größer gewählt wird. Da dieser Zusammenhang jedoch nicht so sehr im Fokus dieser Arbeit liegt, vernachlässigen wir die äußerst rechenzeitlastige Überprüfung dieser These.

### 5.3.2.3 Begründung der Wahl des Wahrscheinlichkeitsmaßes

Wir wollen in diesem Abschnitt die Wahl des Wahrscheinlichkeitsmaßes  $\pi_d$  anstelle von  $\lambda_d$  begründen und beginnen mit dem folgenden

**Lemma (zufällige Wahl der Punkte bezüglich  $\lambda_d$ )**

Ist  $x^* \in \bar{\Gamma}(X)$  mit

$$\delta(x^*) = \sup\{\delta(x) ; x \in [0, 1]^d\}$$

und  $\varepsilon \in ]0, V_{x^*}[$ , so gilt

$$\begin{aligned} \lambda_d(\{x \in [0, 1]^d ; \delta(x^*) - \delta(x) \leq \varepsilon\}) &\geq \frac{1}{d!} \cdot \frac{\varepsilon^d}{V_{x^*}^{d-1}} \\ &\geq \frac{\varepsilon^d}{d!} \end{aligned}$$

Zudem ist diese Ungleichung in dem Sinne scharf, dass wir eine Folge  $(X_m)_{m \in \mathbb{N}}$  von endlichen Folgen  $X_m = (x_m^i)_{i=1}^n$  in  $[0, 1]^d$  mit

$$\lim_{\varepsilon \rightarrow 0} \lim_{m \rightarrow \infty} \frac{d!}{\varepsilon^d} \cdot \lambda_d(\{x \in [0, 1]^d ; \delta(x_m^*) - \delta(x) \leq \varepsilon\}) = 1$$

finden können.

Sind ferner  $\delta > 0$ ,  $R \in \mathbb{N}$  gegeben und  $r_1, \dots, r_R \in [0, 1]^d$  bezüglich des Lebesgue-Maßes zufällig gewählt, so gilt im Fall

$$R \geq \left\lceil \frac{\ln(\delta)}{\ln(1 - \frac{\varepsilon^d}{d!})} \right\rceil$$

schon

$$P(\delta(x^*) - \max\{\delta(r_i) ; 1 \leq i \leq R\} \leq \varepsilon) \geq 1 - \delta.$$

Den eher technischen Beweis lassen wir an dieser Stelle aus. Er kann in [Gne07] nachgelesen werden.

Für  $\varepsilon \leq \frac{1}{2}$  ist  $\left| (\ln(1 - \frac{\varepsilon^d}{d!}))^{-1} \right| \sim \frac{d!}{\varepsilon^d}$ , so dass die Anzahl  $R$  der Punkte, die wir benötigen, um sicherzustellen, dass mit einer nach unten durch  $1 - \delta$  abschätzbaren Wahrscheinlichkeit die Ungleichung

$$\delta(x^*) - \max\{\delta(r_i) ; 1 \leq i \leq R\} \leq \varepsilon$$

gilt, super-exponentiell in  $d$  wächst.

Führen wir obige Analyse mit dem in 5.3.2 eingeführten Wahrscheinlichkeitsmaß  $\pi_d$  anstelle von  $\lambda_d$  durch, so erhalten wir das folgende, in [Gne07] nachzulesende

**Lemma (zufällige Wahl der Punkte bezüglich  $\pi_d$ )**

Ist wieder  $x^* \in \bar{\Gamma}(X)$  mit  $\delta(x^*) = \sup\{\delta(x) ; x \in [0, 1]^d\}$ ,  $\varepsilon \in ]0, V_{x^*}[$  und bezeichnet  $P_\varepsilon^\pi$  die Wahrscheinlichkeit des Ereignisses  $\{x \in [0, 1]^d ; \delta(x^*) - \delta(x) \leq \varepsilon\}$  (bezüglich  $\pi_d$ ), so gilt

$$P_\varepsilon^\pi \geq \varepsilon^d.$$

Diese Ungleichung ist scharf, d.h. wir finden eine Folge  $\hat{X} = (\hat{x}^i)_{i=1}^n$  in  $[0, 1]^d$  mit  $P_\varepsilon^\pi(\hat{X}) = \varepsilon^d$ .

Sind zudem erneut  $\delta > 0, R \in \mathbb{N}$  gegeben und  $r_1, \dots, r_R \in [0, 1]^d$  zufällig gewählt, diesmal bezüglich  $\pi_d$ , so erhalten wir

$$P(\delta(x^*) - \max\{\delta(r_i) ; 1 \leq i \leq R\} \leq \varepsilon) \geq 1 - \delta,$$

sofern

$$R \geq \left\lceil \frac{\ln(\delta)}{\ln(1 - \varepsilon^d)} \right\rceil.$$

Wieder ist die Anzahl der Punkte, die wir benötigen, um sicherzustellen, dass mit einer nach unten durch  $1 - \delta$  abschätzbaren Wahrscheinlichkeit die Ungleichung

$$\delta(x^*) - \max\{\delta(r_i) ; 1 \leq i \leq R\} \leq \varepsilon$$

gilt, exponentiell in  $d$  (wie aufgrund des Resultates aus 3.3.3.1 zu erwarten war), aber nicht super-exponentiell wie im Fall des Lebesgue-Maßes.

Analoge Resultate für  $\bar{\delta}$  wären an dieser Stelle eventuell interessant, dennoch sind oben angegebene Lemmata ein erster Indikator für die Verwendung des Maßes  $\pi_d$  anstelle von  $\lambda_d$ . Zudem wird die hier vorgestellte Untersuchung durch die besseren Ergebnisse des modifizierten Algorithmus bei Anwendung auf verschiedene Punktemengen (siehe hierzu auch 5.4) gestützt.

Ein weiteres Indiz für das Heranziehen von  $\pi_d$  ist die Tatsache, dass für den Mittelwert  $\eta$  der Koordinaten der Gitterpunkte, in denen die Diskrepanz der in 5.4 angegebenen Punktemengen angenommen wird, folgende Gleichheit gilt:

$$\eta = \begin{cases} 0,799743 & , \text{ für die Folgen in } [0, 1]^4 \\ 0,840825 & , \text{ für die Folgen in } [0, 1]^5 \\ 0,873523 & , \text{ für die Folgen in } [0, 1]^6 \end{cases}$$

Berechnen wir den Erwartungswert  $\mu_d = \frac{d}{d+1}$  des Wahrscheinlichkeitsmaßes  $\pi_d$ , so erhalten wir

$$\mu_d = \begin{cases} 0,8 & , \text{ im Fall } d = 4 \\ 0,8\bar{3} & , \text{ im Fall } d = 5 \\ 0,857143 & , \text{ im Fall } d = 6 \end{cases}$$

wohingegen das Lebesguemaß  $\lambda_d$  stets (also für alle  $d \in \mathbb{N}$ ) einen Erwartungswert von  $\frac{1}{2}$  aufweist.

### 5.3.3 Wahl des Startpunktes

Für die Wahl des Startpunktes  $x^c$  ziehen wir zunächst (bezüglich des Lebesgue-Maßes) zufällige Zahlen  $y_j \in [0, 1]$ ,  $j = 1, \dots, d$  und führen das oben angegebene Rundungsverfahren mit  $x = (\sqrt[d]{y_j})_{j=1}^d$  durch. Diese Wahl entspricht dem Ziehen eines bezüglich  $\pi_d$  zufälligen Vektors.

Schließlich setzen wir  $x^c = x^+$ , falls  $\hat{\delta}(x) = \delta(x^+)$  und  $x^c = x^-$  andernfalls.

### 5.3.4 Wahl der Thresholds

Wie bei Winker und Fang betrachten wir eine vorgegebene Anzahl  $N$  von Paaren  $(x^i, y^i) \in \bar{\Gamma}(X)$ , wobei  $x^i$  durch ein Vorgehen wie bei der Wahl des Startpunktes und  $y^i \in \{y^{i,+}, y^{i,-}\}$  aus der in 5.3.2 beschriebenen Nachbarschaft so gewählt sei, dass  $\delta^*(y^i) = \hat{\delta}(y^i)$  gilt.

Nun ordnen wir mithilfe der Abbildung  $T$  die Werte

$$\{-|\delta^*(x^i) - \delta^*(y^i)|; 1 \leq i \leq N\}$$

in aufsteigender Reihenfolge.  $T$  ist die im modifizierten Verfahren verwendeten Schwellenwertfunktion,  $N$  wie zuvor die Anzahl der äußeren Schleifen. Zum Vergleich der beiden Algorithmen aus 5.2 und 5.3 werden wir in 5.4 stets  $N$  wie im Vergleichsalgorithmus von Winker und Fang wählen.

Die Untersuchung geeigneter Wahlen von  $N$  ist sicherlich auch interessant, steht aber ebenfalls nicht im Fokus dieser Arbeit. Experimentelle Ergebnisse zeigen zudem auf den ersten Blick bei kleiner Variation von  $N$  keinen großen Einfluss des Wertes auf die Qualität der Ergebnisse, so dass wir uns mit dieser Wahl von  $N$  zufrieden geben.

## 5.4 Vergleich der vorgestellten Algorithmen

Um die beiden Algorithmen miteinander zu vergleichen werden wir sie zunächst auf solche Folgen anwenden, deren Diskrepanz wir kennen. Dadurch können wir auch eine Aussage über die Genauigkeit der Algorithmen treffen.

Wie wir in Kapitel 3, ist die Berechnung des exakten Wertes der Sterndiskrepanz in einer rechtfertigenden Zeit nur für niedrigdimensionale Folgen möglich. Wir werden daher in einem zweiten Schritt die Algorithmen aus 5.2 und 5.3 auch auf solche Folgen anwenden, deren Diskrepanz wir nicht kennen. Hier steht dann der relative Vergleich der Algorithmen im Vordergrund.

Die von uns betrachteten Folgen sind sogenannte Folgen von Good Lattice Points, die wir zunächst definieren wollen. In 5.4.2 werden wir dann die Ergebnisse der Tests vorstellen.

### 5.4.1 Good Lattice Points

Es gibt verschiedene Definitionen der Good Lattice Points, von denen wir hier die in [WF97] angegebene Variante heranziehen wollen. In 5.4.1.3 ist eine Übersicht über verschiedene Folgen von Good Lattice Points zu finden.

#### 5.4.1.1 Definition (Good Lattice Points)

Sind  $n, d \in \mathbb{N}$  und  $h_1, \dots, h_d \in \mathbb{N}_{<n}$  paarweise verschieden so, dass wenigstens ein  $j \in \mathbb{N}_{\leq d}$  mit  $\text{ggT}(h_j, n) = 1$  existiert, so setzen wir

$$X = (x^i)_{i=1}^n \quad \text{mit} \quad x^i = \left( \left\{ \frac{2 \cdot i \cdot h_j - 1}{2 \cdot n} \right\} \right)_{j=1}^d,$$

wobei für alle  $r \in \mathbb{R}$  mit  $\{r\}$  der fraktionale Anteil von  $r$ , i.e.:  $\{r\} := r - \lfloor r \rfloor$  bezeichnet sei.

$X$  heißt die von  $(n, h_1, \dots, h_d)$  erzeugte Folge von Good Lattice Points.

#### 5.4.1.2 Bemerkung (Good Lattice Points)

Für alle  $n, d \in \mathbb{N}_{\geq 2}$  existiert eine Folge  $X$  von Good Lattice Points mit

$$D_n^*(X) = O\left(\frac{\log(n)^d}{n}\right).$$

Ein Beweis dieser Tatsache kann in [Nie92] gefunden werden.

#### 5.4.1.3 Beispiele von Good Lattice Points

Die folgende Tabelle gibt eine Übersicht über zufällig gewählte Good Lattice Points. Die Diskrepanz der Folgen wurde von Winker und Fang mithilfe des Algorithmus von Bundschuh und Zhu berechnet und kaufmännisch gerundet. Zur Abkürzung werden wir die Punktemengen mit Namen versehen. Der Name setzt sich aus der Dimension  $d$  sowie der Anzahl der Punkte  $n$  zusammen.

Name	d	n	$h_1$	$h_2$	$h_3$	$h_4$	$h_5$	$h_6$	Diskrepanz
4.145	4	145	38	74	80	143			0.0730971
4.255	4	255	5	70	110	131			0.1093016
4.312	4	312	138	143	274	285			0.0616847
4.376	4	376	91	218	279	300			0.0753145
4.388	4	388	13	321	362	367			0.1297285
4.442	4	442	64	368	370	431			0.0619603
4.448	4	448	206	264	279	433			0.0547952

4.451	4	451	225	233	343	348			0.0270495
4.471	4	471	1	82	113	401			0.0286382
4.487	4	487	95	248	251	273			0.0412704
5.102	5	102	13	32	45	54	81		0.1215843
5.122	5	122	16	35	51	75	95		0.08601837
5.147	5	147	22	69	83	89	120		0.1455970
5.153	5	153	4	52	65	72	99		0.1074733
5.169	5	169	9	82	84	106	157		0.0755017
5.170	5	170	31	49	128	132	154		0.0860206
5.195	5	195	58	155	159	168	177		0.1573699
5.203	5	203	69	86	91	104	174		0.1674938
5.235	5	235	79	111	133	169	224		0.0786145
5.236	5	236	33	61	65	176	189		0.0581708
6.28	6	28	1	4	6	14	15	21	0.5360332
6.29	6	29	1	8	12	14	19	22	0.2531973
6.35	6	35	7	12	17	23	28	30	0.3430612
6.50	6	50	10	13	18	30	35	41	0.3148290
6.61	6	61	15	20	50	51	54	56	0.1937383
6.73	6	73	40	49	51	54	56	63	0.1485418
6.81	6	81	2	11	38	60	70	79	0.2500000
6.88	6	88	6	10	25	27	40	82	0.2658170
6.90	6	90	15	28	33	52	68	74	0.1991531
6.92	6	92	2	8	18	60	65	89	0.1635152

Zusätzlich zu den hier angegebenen Good Lattice Point haben wir wie angesprochen auch einige hochdimensionale Punktemengen betrachtet, deren erzeugende Vektoren wir hier aus Gründen der Übersichtlichkeit nicht angeben wollen. Eine Übersicht über die berechneten Werte wird in 5.4.2.2 angeben.

## 5.4.2 Ergebnisse der Experimente

### 5.4.2.1 Vergleich der Algorithmen für niedrigdimensionale Punktemengen

Wir haben für die oben angegebenen Mengen jeweils 100 Läufe mit je 10.000 (im Fall des modifizierten Algorithmus: etwa 10.000) Iterationen und jeweils 10 Läufe mit (etwa) 100.000 Läufen durchgeführt und dabei die unten dargestellten Ergebnisse erhalten. Dabei ist der jeweils bessere der beiden Werte der verschiedenen Algorithmen fett gedruckt.

In der Spalte „bester Wert =  $D_n^*(X)$  ?“ ist in dem Fall, dass in keinem der Läufe die tatsächliche Diskrepanz berechnet wurde, der größte berechnete Wert angegeben. Wurde die Diskrepanz wenigstens in einem der Läufe berechnet, so kennzeichnen wir dieses durch ein  $\surd$ .

Ergebnisse für 100 Läufe à 10.000 Iterationen									
Name	Algorithmus von Winker & Fang					modifizierter Algorithmus			
	$mc$	$k$	Mittelwert	bester Wert $= D_n^*(X)$ ?	# Beste	mc	Mittelwert	bester Wert $= D_n^*(X)$ ?	# Beste
4.145	2	20	0.0685530	✓	8	2	<b>0.0713699</b>	✓	<b>34</b>
4.255	2	20	<b>0.1039984</b>	✓	37	2	0.1034308	✓	<b>55</b>
4.312	2	20	0.0609991	✓	47	2	<b>0.0616664</b>	✓	<b>98</b>
4.376	2	20	0.0699586	✓	13	2	<b>0.0745983</b>	✓	<b>15</b>
4.388	2	20	0.1276026	✓	16	2	<b>0.1294377</b>	✓	<b>31</b>
4.442	2	20	0.0560917	✓	9	2	<b>0.0613634</b>	✓	<b>23</b>
4.448	2	20	0.0542652	✓	21	2	<b>0.0545602</b>	✓	<b>31</b>
4.451	2	20	0.0231735	✓	4	2	<b>0.0247233</b>	✓	1
4.471	2	20	0.0212515	✓	9	2	<b>0.0253546</b>	✓	3
4.487	2	20	0.0340007	0.0411668	0	2	<b>0.0402131</b>	✓	6
5.102	2	20	0.1110855	✓	3	2	<b>0.1201658</b>	✓	<b>6</b>
5.122	2	20	0.0807898	0.0858382	0	2	<b>0.0721217</b>	✓	<b>1</b>
5.147	2	20	0.1349240	✓	2	2	<b>0.1454122</b>	✓	<b>71</b>
5.153	2	20	0.1029979	✓	8	2	<b>0.1046833</b>	✓	<b>67</b>
5.169	2	20	0.0670616	0.0744305	0	2	<b>0.0714578</b>	✓	<b>10</b>
5.170	2	20	0.0761575	✓	1	2	<b>0.0808647</b>	✓	<b>46</b>
5.195	2	20	0.1495163	✓	1	2	<b>0.1570987</b>	✓	<b>79</b>
5.203	2	20	0.1651218	✓	6	2	<b>0.1646376</b>	✓	<b>57</b>
5.235	2	20	0.0670771	0.0781176	0	2	<b>0.0775982</b>	✓	<b>34</b>
5.236	2	20	0.0518718	0.0578633	0	2	<b>0.0525930</b>	✓	1
6.28	3	5	<b>0.5360332</b>	✓	<b>100</b>	3	0.5064167	✓	75
6.29	3	5	0.2466770	✓	26	3	<b>0.2517523</b>	✓	<b>64</b>
6.35	3	5	<b>0.3274384</b>	✓	<b>76</b>	3	0.2805028	✓	1
6.50	3	5	<b>0.3093912</b>	✓	<b>82</b>	3	0.3036223	✓	1
6.61	3	10	0.1764319	✓	12	3	<b>0.1898090</b>	✓	<b>35</b>
6.73	3	10	0.1422024	✓	5	3	<b>0.1438144</b>	✓	5
6.81	3	10	<b>0.2491100</b>	✓	<b>16</b>	3	0.2489546	✓	11
6.88	3	10	0.2573444	✓	50	3	<b>0.2576588</b>	✓	<b>89</b>
6.90	3	10	0.1936678	✓	66	3	<b>0.1990691</b>	✓	<b>98</b>
6.92	3	10	0.1521884	✓	9	3	<b>0.1630862</b>	✓	<b>73</b>

Wie wir sehen, ist der modifizierte Algorithmus dem von Winker und Fang bis auf in wenigen Ausnahmen überlegen.

Dieses Bild bestätigt sich auch dann, wenn man nicht die Zahlen unserer Implementierung, sondern die in [WF97] angegebenen Zahlen als Vergleichswerte heranzieht, sofern man die Größe der Abweichungen berücksichtigt.

Es ist auffällig, dass die in [WF97] angeführten Ergebnisse deutlich besser sind als diejenigen, die wir bei Anwendung des Algorithmus von Winker und Fang auf die genannten Punktemengen erhalten. Wir vermuten hier, dass diese Tatsache auf verschiedene Pseu-



dozufallszahlgeneratoren zurückzuführen ist. Leider steht uns der von Winker und Fang verwendeten Zufallsgenerator GAUSS 3.2 nicht zur Verfügung und wir können daher diese These nicht verifizieren. Im Zuge der vorliegenden Arbeit wurden beide Algorithmen mit verschiedenen Zufallsgeneratoren getestet, einen Einfluss auf die Qualität der Ergebnisse konnte jedoch nicht festgestellt werden.

Auch bei Durchführung weniger Läufe mit einer größeren Anzahl an Iterationen erhalten wir ein ähnliches Bild. Man kann zudem erkennen, dass die so erhaltenen Werte eine deutlich bessere Qualität aufweisen als die derjenigen Tests, bei denen wir eine kleinere Anzahl von Iterationen je Lauf durchführen

Ergebnisse für 10 Läufe à 100.000 Iterationen									
Name	Algorithmus von Winker & Fang					modifizierter Algorithmus			
	$mc$	$k$	Mittelwert	bester Wert $= D_n^*(X)?$	# Beste	$mc$	Mittelwert	bester Wert $= D_n^*(X)?$	# Beste
4.145	2	20	0.0719488	✓	3	2	<b>0.0730971</b>	✓	<b>10</b>
4.255	2	20	0.1093016	✓	10	2	0.1093016	✓	10
4.312	2	20	0.0616847	✓	10	2	0.0616847	✓	10
4.376	2	20	0.0723916	✓	5	2	<b>0.0752071</b>	✓	<b>6</b>
4.388	2	20	0.1285870	✓	7	2	<b>0.1297285</b>	✓	<b>10</b>
4.442	2	20	0.0587303	✓	4	2	<b>0.0619602</b>	✓	<b>9</b>
4.448	2	20	0.0546665	✓	3	2	<b>0.0547088</b>	✓	<b>5</b>
4.451	2	20	0.0255887	✓	1	2	<b>0.0263468</b>	✓	<b>4</b>
4.471	2	20	0.0261538	✓	5	2	<b>0.0282533</b>	✓	<b>8</b>
4.487	2	20	0.0383164	✓	1	2	<b>0.0411233</b>	✓	<b>5</b>
5.102	3	20	0.1189783	✓	1	2	<b>0.1211102</b>	✓	<b>5</b>
5.122	3	20	<b>0.0841287</b>	✓	1	2	0.0825975	✓	1
5.147	3	20	0.1441100	✓	4	2	<b>0.1455970</b>	✓	<b>10</b>
5.153	3	20	0.1061391	✓	3	2	<b>0.1074733</b>	✓	<b>10</b>
5.169	3	20	0.0729793	✓	3	2	<b>0.0733372</b>	✓	<b>4</b>
5.170	3	20	0.0836306	✓	4	2	<b>0.0849748</b>	✓	<b>9</b>
5.195	3	20	0.1529967	✓	2	2	<b>0.1573699</b>	✓	<b>10</b>
5.203	3	20	<b>0.1672632</b>	✓	<b>9</b>	2	0.1659358	✓	7
5.235	3	20	0.0731559	✓	1	2	<b>0.0786145</b>	✓	<b>10</b>
5.236	3	20	<b>0.0574027</b>	✓	<b>2</b>	2	0.0555602	✓	1
6.28	3	5	5.360332	✓	10	3	5.360332	✓	10
6.29	3	5	2.526792	✓	4	3	<b>0.2531973</b>	✓	<b>10</b>
6.35	3	5	3.430612	✓	10	3	0.3430612	✓	10
6.50	3	5	<b>3.148290</b>	✓	<b>10</b>	3	0.3066884	✓	1
6.61	3	10	1.856141	✓	<b>6</b>	3	<b>0.1909096</b>	✓	4
6.73	3	10	<b>1.470112</b>	✓	<b>3</b>	3	0.1464848	✓	1
6.81	3	10	<b>2.499848</b>	✓	<b>9</b>	3	0.2491247	✓	4
6.88	3	10	0.2658170	✓	10	3	0.2658170	✓	10
6.90	3	10	0.1991531	✓	10	3	0.1991531	✓	10
6.92	3	10	0.1633393	✓	7	3	<b>0.1635070</b>	✓	<b>9</b>

### 5.4.2.2 Vergleich der Algorithmen für hochdimensionale Punktemengen

Wir werden nun noch die Ergebnisse der beiden Algorithmen bei Anwendung auf Folgen von Good Lattice Points in höheren Dimensionen angeben. Hier erkennt man die Überlegenheit des modifizierten Algorithmus deutlich.

Ergebnisse für 10 Läufe à 100.000 Iterationen			
		Algorithmus von Winker & Fang	modifizierter Algorithmus
Name	$mc$	bester Wert	bester Wert
27.8059	2	0.018848	0.045625
27.8059	3	0.024276	0.043777
27.8059	5	0.009646	0.040673
31.5189	2	0.013479	0.069721
31.5189	3	0.031610	0.059156
31.5189	5	0.013895	0.052627
40.6577	2	0.090679	0.126714
40.6577	3	0.058229	0.126614
40.6577	5	0.007757	0.066687

Bei Durchführung dieser Tests haben wir im Algorithmus von Winker und Fang den Wert  $k$  auf 350 fixiert. Für den modifizierten Algorithmus gilt  $k = \lfloor \frac{n}{16} \rfloor$ . Bei Anwendung der Algorithmen mit den Werten  $k = 250$  bzw.  $k = \lfloor \frac{n}{8} \rfloor$  erhalten wir vergleichbare, aber leicht schlechter Ergebnisse.

Da wir die tatsächlichen Diskrepanzen der oben angegebenen Folgen nicht berechnen können (die Anzahl der zu betrachtenden Gitterpunkte ist etwa  $8060^{27}$  für die Instanz 27.8059, der Algorithmus von Bundschuh und Zhu würde selbst auf der schnellsten Rechenmaschine für diese Berechnung viele Jahre benötigen), können wir hier keine Aussage über die Qualität der absoluten Werte der Ergebnisse treffen. Der Vergleich zeigt jedoch, dass die vom modifizierten Algorithmus berechneten Werte deutlich größer als die des Algorithmus von Winker und Fang sind und daher eine bessere Approximation der Diskrepanzen darstellen.

Weitere Untersuchungen für die geeignete Wahl der Parameter  $mc$ ,  $k$  und  $N$  können den hier vorgestellten Algorithmus eventuell weiter verbessern. Uns genügt an dieser Stelle die Einsicht, dass selbst bei einer relativ willkürlichen Wahl dieser Werte eine dem im 5.2 vorgestellten Algorithmus überlegene Heuristik gefunden wurde.

Die hier präsentierten Ergebnisse lassen hoffen, dass anstelle der effizienten Berechnung der Sterndiskrepanz wenigstens eine akzeptable Approximation der tatsächlichen Diskrepanz mithilfe von Heuristiken möglich ist.

# Kapitel 6

## Ausblick

### Offene Fragen bezüglich der Komplexität

Wir haben in dieser Arbeit gesehen, dass die Berechnung der Sterndiskrepanz einer gegebenen Folge  $\mathcal{NP}$ -vollständig ist und wir nicht erwarten können, dass für das Optimierungsproblem ein effizienter Algorithmus existiert, der die Diskrepanz beliebig gut approximiert. An diese Feststellung knüpft sich nun die Frage nach einer konstanten Approximierbarkeit der Diskrepanz an, i.e.:

Existiert ein polynomieller Algorithmus  $\mathcal{A}$  und eine reelle Zahl  $C$  so, dass für alle endlichen Folgen  $X$  die Ungleichung

$$\frac{D_n^*(X)}{\delta^*(\mathcal{A}(X))} \leq C$$

gilt?

Zudem wäre eine Einsicht in die Komplexität der Approximation der Diskrepanz relevanter Punktemengen, etwa der Good Lattice Points, interessant.

### Konstruktion von Folgen geringer Diskrepanz

Wir haben in 2.1.3 gesehen, dass für die numerische Integration solche Folgen von Interesse sind, die eine niedrige Diskrepanz aufweisen.

Hier wäre zum einen interessant, die tatsächliche Diskrepanz bekannter Punktemengen auch in höheren Dimension zu testen. Erste Untersuchungen zu Faure-, Halton- und Sobol-folgen im 7-dimensionalen Einheitswürfel wurden im Artikel [Thi01b] von Eric Thiémond vorgestellt.

Zum anderen ist es auch denkbar, dass die in dieser Arbeit vorgestellten Verfahren einen Beitrag zur Konstruktion von Folgen geringer Diskrepanz liefern können, da sie die Berechnung von Diskrepanzen auch höherdimensionaler Räume ermöglichen.



# Literaturverzeichnis

- [AK89] AARTS, E. und J. KORST: *Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing*. John Wiley & Sons, Inc., New York, NY, USA, 1989.
- [AK91] ALTHÖFER, I. und K.-U. KOSCHNIK: *On the Convergence of Threshold Accepting*. Applied Mathematics and Optimization, 24:183–195, 1991.
- [BZ93] BUNDSCHUH, P. und Y. ZHU: *A method for exact calculation of the discrepancy of lowdimensional finite point sets*. Abhandlungen aus dem Math. Seminar der Univ. Hamburg, 63:115–133, 1993.
- [Cha01] CHAZELLE, B.: *The Discrepancy Method*. Cambridge University Press, 2001.
- [Cle86] CLERCK, L. DE: *A method for exact calculation of the star discrepancy of plane sets applied to the sequence of Hammersley*. Monatshefte für Mathematik, 101(4):261–278, 1986.
- [DEM96] DOBKIN, D. P., D. EPPSTEIN und D. P. MITCHELL: *Computing the discrepancy with applications to supersampling patterns*. ACM Trans. Graph., 15(4):354–376, 1996.
- [DGS05] DOERR, B., M. GNEWUCH und A. SRIVASTAV: *Bounds and constructions for the star-discrepancy via  $\delta$ -covers*. Journal of Complexity, 21(5):691–709, 2005.
- [DS90] DUECK, G. und T. SCHEUER: *Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing*. J. Comput. Phys., 90(1):161–175, 1990.
- [FH96] FRANK, K. und S. HEINRICH: *Computing the discrepancies of Smolyak quadrature rules*. Journal of Complexity, 12:287–314, 1996.
- [GJ79] GAREY, M. R. und D. S. JOHNSON: *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. W. H. Freeman, 1979.

- [Gne07] GNEWUCH, M.: *A randomized algorithm to compute the star-discrepancy of a given  $n$ -point set*. Preprint, 2007.
- [Hla61] HLAWKA, E.: *Funktionen von beschränkter Variatiou in der Theorie der Gleichverteilung*. Annali di Matematica Pura ed Applicata, 54(1):325–333, 1961.
- [HSW04] HICKERNELL, F. J., I. H. SLOAN und G. W. WASILKOWSKI: *On tractability of weighted integration over bounded and unbounded regions in  $R^s$* . Mathematics of Computation, 73:1885–1901, 2004.
- [Joh87] JOHNSON, D. S.: *The NP-completeness column: an ongoing guide*. J. Algorithms, 8(3):438–448, 1987.
- [KN74] KUIPERS, L. und H. NIEDERREITER: *Uniform Distribution of Sequences (Pure & Applied Mathematics Monograph)*. John Wiley & Sons Inc, 1974.
- [Kok43] KOKSMA, J. F.: *A general theorem from the uniform distribution mod 1 (holländisch)*. Matematica B (Zutphen), (1):7–11, 1942/43.
- [Mat99] MATOUSEK, J.: *Geometric Discrepancy: An Illustrated Guide (Algorithms and Combinatorics)*. Springer, 1999.
- [Nie92] NIEDERREITER, H.: *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM CBMS-NSF, 1992.
- [NP95] NISSEN, V. und H. PAUL: *A modification of threshold accepting and its application to the quadratic assignment problem*. OR Spectrum, 17(2):205–210, 1995.
- [Sma98] SMALE, S.: *Mathematical Problems for the Next Century*. Mathematical Intelligencer, 1998.
- [Thi01a] THIÉMARD, E.: *An algorithm to compute bounds for the star discrepancy*. Journal of Complexity, 17(4):850–880, 2001.
- [Thi01b] THIÉMARD, E.: *Optimal volume subintervals with  $k$  points and star discrepancy via integer programming*. Mathematical Methods of Operations Research, 54(1):21–45, 2001.
- [Tur37] TURING, A. M.: *On Computable Numbers, with an Application to the Entscheidungsproblem*. Proc. London Math. Soc., s2-42(1):230–265, 1937.
- [Weg05] WEGENER, I.: *Theoretische Informatik. - eine algorithmenorientierte Einführung*. Teubner, 2005.

- 
- [Wey16] WEYL, H.: *Über die Gleichverteilung von Zahlen mod. Eins.* Math. Ann., 77:313–52, 1916.
- [WF97] WINKER, P. und K.-T. FANG: *Application of Threshold-Accepting to the Evaluation of the Discrepancy of a Set of Points.* SIAM Journal on Numerical Analysis, 34(5):2028–2042, 1997.





Kiel, den 31. Juli 2007

Hiermit versichere ich, Carola Winzen (Studentin der Mathematik an der Christian-Albrechts-Universität zu Kiel, Matrikelnummer 718017), an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe. Die Arbeit wurde in dieser oder ähnlicher Form noch keiner Prüfungskommission vorgelegt.

CAROLA WINZEN