UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Fakultät für Elektrotechnik, Informatik und Mathematik
Institut für Informatik

Reduktionen von CVP mit wenigen Lösungen auf CVP mit eindeutigen Lösungen

Studienarbeit

von
Martin Niemeier

vorgelegt bei
Prof. Dr. Johannes Blömer

Betreuer
Prof. Dr. Johannes Blömer
Dipl. Math. Stefanie Naewe

13. Juni 2007

Erklärung

Ich versichere, dass ich die beiliegende Studienarbeit ohne Hilfe Dritter und ohne anderer als der angegebenen Quellen und Hilfsmittel angefertigt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Paderborn, den 13. Juni 2007

Martin Niemeier

Danksagung

Diese Studienarbeit ist im Rahmen meines Informatikstudiums an der Universität Paderborn entstanden. Ich möchte mich hiermit bei Prof. Dr. Johannes Blömer, und ganz besonders bei Dipl.-Math. Stefanie Naewe für die sehr gute Betreuung bedanken. Mein Dank richtet sich auch an Grisha Ende, Christian Ikenmeyer und Jaroslaw Klose für das Korrekturlesen und die hilfreichen Kommentare.

Inhaltsverzeichnis

1 Einleitung

Um 1900 begründete Hermann Minkowski die *Geometrie der Zahlen*. Es geht dabei um die Verknüpfung von diskreten Aspekten wie diskreten Punktmengen und Ganzzahligkeit mit stetigen Elementen der Geometrie. Sein Ziel war es, geometrische Methoden, etwa aus der Konvexgeometrie zu verwenden, um zahlentheoretische Erkenntnisse, beispielsweise die Diophantische Approximation, zu fördern [Hil11].

Gegenstand in der Untersuchung der Geometrie der Zahlen sind sogenannte *Gitter*. Dabei handelt es sich um die Mengen der ganzzahligen Linearkombinationen von Vektoren im Vektorraum $\mathbb{R}^m$.

Gitter finden vielfach Anwendung in der theoretischen Informatik. Beispielsweise in der ganzzahligen Optimierung. Während die Gültigkeitsbereiche linearer Programme (LPs) als konvexe Polyeder in Vektorräumen darstellbar sind, kann man den Gültigkeitsbereich von ganzzahligen Programmen (IPs) als Schnitt aus Polyedern und Gittern modellieren. Ein bedeutendes Resultat in der ganzzahligen Optimierung wurde 1981 von Lenstra [Len83] unter Verwendung der Gittertheorie erbracht. Es handelt sich um einen Polynomialzeitalgorithmus, der ganzzahlige Programme (IPs) in fester Dimension des Lösungsraumes, also fester Anzahl an Variablen, löst.

Auch in der Kryptographie spielen Gitter eine wichtige Rolle. Beispielsweise basiert die Sicherheit bestimmter Kryptosysteme auf der Schwierigkeit bestimmter Gitterprobleme.

Aus diesem Grund ist die Untersuchung von Gittern aus komplexitätstheoretischer Sicht interessant, insbesondere die Untersuchung des *Problems des kürzesten Gittervektors* sowie des *Problems des nächsten Gittervektors*. Beim *Problem des kürzesten Gittervektors* wird ein kürzester, von 0 verschiedener Vektor aus einem Gitter gesucht, während beim *Problem des nächsten Gittervektors* zu einem vorgegebenen Zielvektor ein Vektor des Gitters mit minimalem Abstand gesucht wird.

Diese Probleme sind unter anderem deshalb interessant, weil die Sicherheit der oben erwähnten Kryptosysteme auf der Schwierigkeit dieser beiden Probleme basiert. Es gibt auch Ansätze, mit Hilfe von approximierten Lösungen dieser Probleme Angriffe auf Kryptosysteme, wie zum Beispiel RSA, zu formulieren.

Während das Problem des nächsten Gittervektors NP-hart ist, wurde entsprechendes für das Problem des kürzesten Gittervektors bisher nur unter randomisierten Reduktionen gezeigt.

Diese Arbeit wird sich mit der Fragestellung beschäftigen, ob die Schwierigkeit des Problems des nächsten Vektors abnimmt unter dem Wissen, dass die Lösung eindeutig ist, dass also genau ein Gittervektor mit minimalem Abstand zum Zielvektor existiert. Diese Frage wird sich mit Nein beantworten lassen, zumindest für bestimmte Normen.

Valiant und Vazirani [VV86] haben eine entsprechende Aussage für das Erfüllbarkeitsproblem für Boolesche Formeln (SAT) gezeigt, indem sie eine randomisierte Reduktion von SAT auf eine Variation dieses Problems, bei der nur Boolesche Formeln mit höchstens einer erfüllenden Belegung zugelassen sind, formuliert haben.

Diese Technik haben Kumar und Sivakumar in einer Veröffentlichung aus dem Jahre 1999 [KS99] aufgegriffen um zu zeigen, dass sich die Komplexität des Problems des kürzesten Gittervektors unter dem Wissen, dass der kürzeste Vektor eindeutig ist, nicht verringert.

Ihr Beweis wird in dieser Arbeit ausgeführt, und die verwendete Technik auf das Problem des nächsten Vektors übertragen.

Die Arbeit ist wie folgt gegliedert:

Kapitel 2: In diesem Kapitel wird eine Einführung in die Grundlagen der Gittertheorie gegeben. Es werden alle Aspekte der Gittertheorie behandelt, die für das Verständnis dieser Arbeit erforderlich sind. Darüber hinaus werden die beiden Probleme des kürzesten und nächsten Gittervektors sowie ihre Eindeutigkeitsvarianten definiert.

Kapitel 3: Inhalt dieses Kapitels ist die Herleitung und Analyse eines Algorithmus zur Generierung eines Gittersystems. Dieser Algorithmus bildet die Grundlage der Reduktionen, die in den Kapiteln 4 und 5 behandelt werden.

Kapitel 4: Dieses Kapitel beschäftigt sich mit der Schwierigkeit der Eindeutigkeitsvariante des Problems des kürzesten Gittervektors. Im Wesentlichen geht es hier um die Konstruktion einer randomisierten Reduktion vom Problem des kürzesten Gittervektors auf seine Eindeutigkeitsvariante, auf Grundlage der Veröffentlichung von Kumar und Sivakumar [KS99].

Kapitel 5: In diesem Kapitel wird die Schwierigkeit der Eindeutigkeitsvariante des Problems des nächsten Gittervektors untersucht. Es wird wie in Kapitel 4 eine randomisierte Reduktion vom Problem des nächsten Gittervektors auf seine Eindeutigkeitsvariante konstruiert.

2 Grundlagen

Das Ziel dieses Kapitels ist es, alle Grundlagen der Gittertheorie zu vermitteln, die zum Verständnis der Arbeit erforderlich sind.

In Abschnitt 2.2 werden grundlegende Definitionen und Eigenschaften eingeführt, während in Abschnitt 2.3 die beiden in der Komplexitätstheorie bedeutenden Gitterprobleme des kürzesten und nächsten Gittervektors beschrieben werden.

Im folgenden Abschnitt werden Notationen festgelegt, die in dieser Arbeit verwendet werden, und grundlegende Vereinbarungen, beispielsweise bezüglich verwendeter Normen, getroffen.

2.1 Notation und Vereinbarungen

Wie schon in der Einleitung erwähnt, handelt es sich bei Gittern um bestimmte Mengen von Vektoren. Diese Vektoren stammen aus dem m-dimensionalen Vektorraum über $\mathbb{R}$, dem $\mathbb{R}^m$.

Sofern nicht anders definiert, wird in dieser Arbeit für einen Vektor $v \in \mathbb{R}^m$ seine i-te Komponente mit v_i bezeichnet.

Unter $\langle \cdot, \cdot \rangle$ ist das Standardskalarprodukt im $\mathbb{R}^m$ zu verstehen. Für zwei Vektoren $v, w \in \mathbb{R}^m$ gilt also

$$\langle v, w \rangle := \sum_{i=1}^{m} v_i w_i.$$

Um mit Begriffen wie einem „kürzesten“ oder „nächsten“ Vektor arbeiten zu können, ist es notwendig, auf dem Vektorraum eine Norm einzuführen. Also einer Abbildung

$$\|\cdot\| : \mathbb{R}^m \to \mathbb{R}_{\geq 0},$$

welche die Bedingungen der Definitheit, Homogenität und der Dreiecksungleichung erfüllt, also für alle $u, v \in \mathbb{R}^m$ und alle $\alpha \in \mathbb{R}$ gilt:

- $\|v\| = 0 \Leftrightarrow v = 0$
- $\|\alpha \cdot x\| = |\alpha| \cdot \|x\|$
- $\|x + y\| \leq \|x\| + \|y\|$

In der gesamten Arbeit werden hierfür ausschließlich die l_p-Normen verwendet. Sie sind für jeden Vektor $v \in \mathbb{R}^m$ definiert wie folgt:

$$\|v\|_p := \begin{cases} \left(\sum_{i=1}^{m} |v_i|^p\right)^{1/p}, & \text{falls } 1 \leq p < \infty \text{ gilt,} \\ \max\{|v_i| : 1 \leq i \leq m\}, & \text{im Fall } p = \infty. \end{cases}$$

In diesem Zusammenhang bezeichnet man die folgende Menge als abgeschlossene Kugel $B(v,r)$ um einen Vektor $v \in \mathbb{R}^m$ mit Radius $r \in \mathbb{R}_{\geq 0}$:

$$B(v,r) := \left\{ u \in \mathbb{R}^m \ : \ \|u-v\|_p \leq r \right\}.$$

Entsprechend definiert man die offene Kugel $\tilde{B}(v,r)$ um einen Vektor $v \in \mathbb{R}^m$ mit Radius $r \in \mathbb{R}_{\geq 0}$ als:

$$\tilde{B}(v,r) := \left\{ u \in \mathbb{R}^m \ : \ \|u-v\|_p < r \right\}.$$

Bei allen Laufzeitanalysen in dieser Arbeit wird als zugrundeliegendes Rechenmodell eine Erweiterung des Registermaschienen-Modells (RAM) verwendet. Eine detaillierte Beschreibung des RAM-Modells ist in [Pap94] zu finden. Es wird hier als bekannt vorausgesetzt. Das hier verwendete Rechenmodell arbeitet mit Registern, die rationale Zahlen speichern können. Jede rationale Rechenoperation, beispielsweise Zuweisung, Addition, Subtraktion oder Multiplikation auf rationalen Zahlen benötigt in diesem Modell einen Zeitschritt.

Eine RAM kann mit polynomiellem Zeitaufwand bezüglich der Kodierungslänge der Eingabe rationale Rechenoperationen simulieren. Zur Speicherung einer rationalen Zahl verwendet sie drei Register, jeweils eines für Vorzeichen, ganzzahligen Zähler und Nenner der rationalen Zahl. Zähler und Nenner werden in gekürzter Darstellung gespeichert, welche sich effizient unter Verwendung des Euklidischen Algorithmus zur Berechnung des größten gemeinsamen Teilers bestimmen lässt. Eine rationale Rechenoperation mit Transformation des Ergebnisses in die gekürzte Darstellung lässt sich auf dieser Kodierung also, unter Verwendung bekannter Rechenregeln (Bruchrechnung), mit einer polynomiell beschränkten Anzahl von Schritten simulieren.

Eine RAM selbst kann wiederum auf einer Turingmaschine mit nur polynomiellem Mehraufwand bezüglich der Kodierungslänge der Eingabe simuliert werden ([Pap94], Theorem 2.5).

Das hier verwendete Rechenmodell ist also polynomiell reduzierbar auf das Modell der Turingmaschine. Jeder Algorithmus, der nach dem hier verwendeten Modell polynomiell zeitbeschränkt ist, kann demnach auch auf einer Turingmaschine in polynomieller Zeit simuliert werden.

Da für alle Laufzeitaussagen in dieser Arbeit nur von Bedeutung ist, dass sie polynomiell zeitbeschränkt sind, ist es also vollkommen ausreichend, dieses einfacher zu handhabende Modell zu verwenden.

2.2 Einführung in die Gittertheorie

Dieser Abschnitt soll eine kurze Einführung in die Gittertheorie geben. Er basiert auf [MG02] und ist so konzipiert, dass alle Grundlagen behandelt werden, die für das Verständnis dieser Arbeit erforderlich sind. Für einen umfassenden Einstieg in die Gittertheorie ist [MG02] empfehlenswert.

Definition 2.2.1 (Gitter)**.** *Sei $B = \{b_1, \ldots, b_n\}$ eine Menge von n linear unabhängigen Vektoren $b_i \in \mathbb{R}^m$. Unter einem* Gitter *versteht man die Menge der ganzzahligen Linearkombinationen von $b_1, \ldots, b_n$:*

$$\Lambda = \mathcal{L}(b_1, \ldots, b_n) := \left\{ \sum_{i=1}^{n} x_i b_i \ : \ x_i \in \mathbb{Z} \right\}.$$

In Analogie zu Vektorräumen bezeichnet man die Menge B als *Gitterbasis*. Die Anzahl n an Basisvektoren definiert den *Rang* des Gitters. Die *Dimension* des Gitters entspricht der Dimension m des zugrundeliegenden Vektorraumes. Gilt $n = m$, so nennt man das Gitter *volldimensional*. Eine Teilmenge eines Gitters Λ, welche selbst wieder ein Gitter bildet, bezeichnet man als *Untergitter* von Λ.

Wenn man die Gitterbasis als diejenige $(m \times n)$-Matrix

$$B = [b_1, b_2 \ldots, b_n] \in \mathbb{R}^{m \times n}$$

definiert, welche die Vektoren $b_1, \ldots, b_n$ in dieser Reihenfolge als Spaltenvektoren enthält, so kann man Gitter vereinfacht schreiben als

$$\mathcal{L}(B) = \{Bx \ : \ x \in \mathbb{Z}^n\}.$$

Für einen Gittervektor $v = Bx \in \mathcal{L}(B)$ bezeichnet man den Vektor x als den *Koeffizientenvektor* von v bezüglich der Basis B. Dieser Vektor ist eindeutig bestimmt durch v und B. In dieser Arbeit wird sowohl die Mengennotation, als auch die Matrixnotation für Gitterbasen verwendet. B wird je nach Kontext sowohl die Bedeutung einer Menge von Basisvektoren, als auch die einer Matrix haben, die sich aus Basisvektoren zusammensetzt. Welche Bedeutung im konkreten Fall gemeint ist, wird immer aus dem Kontext ersichtlich werden.

Man kann ein Gitter Λ der Dimension m auch auffassen als diskrete (additive) Untergruppe des $\mathbb{R}^m$.

Die Basis eines Vektorraumes ist im Allgemeinen nicht eindeutig. Gleiches gilt auch für Gitterbasen. Zwei Basen, die das gleiche Gitter erzeugen, nennt man *äquivalent*.

Den Zusammenhang äquivalenter Basen kann man algebraisch mit Hilfe von unimodularen Matrizen charakterisieren. Eine *unimodulare Matrix* ist eine Matrix mit ganzzahligen Einträgen und Determinante ± 1.

Lemma 2.2.2. *Zwei Basen $B_1, B_2 \in \mathbb{R}^{m \times n}$ sind genau dann äquivalent, wenn es eine unimodulare Matrix U gibt mit*

$$B_1 = B_2 U.$$

Beweis. Der Beweis basiert auf einer Vorlesungsmitschrift von Regev [Reg04]. Gelte zunächst, dass die Basen B_1 und B_2 äquivalent sind. Die von ihnen erzeugten Gitter sind also identisch:

$$\mathcal{L}(B_1) = \mathcal{L}(B_2).$$

Seien $b_1, \ldots, b_n \in \mathbb{R}^m$ die Basisvektoren von B_1, gelte also $B_1 = [b_1, \ldots, b_n]$. Jeder Basisvektor b_i ist in $\mathcal{L}(B_1)$ enthalten, und damit auch im identischen Gitter $\mathcal{L}(B_2)$. Das bedeutet aber, dass es für jeden Basisvektor b_i einen Koeffizientenvektor $x \in \mathbb{Z}^n$ gibt, der $b_i = B_2 x$ erfüllt. Zusammen bilden diese Koeffizientenvektoren eine Matrix $X \in \mathbb{Z}^{n \times n}$, für die

$$B_1 = B_2 X \tag{1}$$

gilt. Durch vertauschen der Rollen von B_1 und B_2 zeigt man, dass es auch eine Matrix $Y \in \mathbb{Z}^{n \times n}$ mit

$$B_2 = B_1 Y \tag{2}$$

gibt.

Durch Einsetzen von (1) in (2) erhält man

$$B_2 = B_1 Y = B_2 (XY).$$

Es folgt

$$B_2^T B_2 = (XY)^T B_2^T B_2 (XY).$$

Mit dem Determinantenproduktsatz schließt man

$$\det(B_2^T B_2) = \det(XY)^2 \det(B_2^T B_2).$$

Als Matrix mit linear unabhängigen Spaltenvektoren ist B_2 nicht singulär. Der Term $\det(B_2^T B_2)$ ist also von Null verschieden und kann gekürzt werden. Man erhält:

$$\pm 1 = \det(XY) = \det(X) \det(Y).$$

Da X und Y ganzzahlige Matrizen sind, haben sie auch ganzzahlige Determinanten, und es folgt:

$$\det(X) = \pm 1 \text{ und } \det(Y) = \pm 1.$$

X ist also unimodular. Setzt man $U := X$, so folgt die erste Implikation der Behauptung.

Gebe es nun eine unimodulare Matrix U mit $B_1 = B_2 U$. Als unimodulare Matrix ist U ganzzahlig. Man kann die i-te Spalte von U also auffassen als ganzzahlige Linearkombination des

i-ten Basisvektors von B_1 durch B_2. Alle Basisvektoren von B_1 sind also in $\mathcal{L}(B_2)$ enthalten und es folgt

$$\mathcal{L}(B_1) \subseteq \mathcal{L}(B_2). \tag{3}$$

Das Inverse einer unimodularen Matrix ist ebenfalls unimodular. Es folgt

$$B_2 = B_1 U^{-1}$$

mit einer unimodularen Matrix U^{-1}. Durch vertauschen der Rollen von B_2 und B_1 kann man nun völlig analog argumentieren und erhält, dass auch

$$\mathcal{L}(B_2) \subseteq \mathcal{L}(B_1) \tag{4}$$

gilt. (3) und (4) liefern zusammen

$$\mathcal{L}(B_1) = \mathcal{L}(B_2).$$

B_1 und B_2 sind also äquivalent. □

Wichtige Gitterkonstanten sind die sogenannten *sukzessiven Minima*. Das i-te sukzessive Minimum eines Gitters ist der minimale Radius einer abgeschlossenen Kugel um 0, so dass mindestens i linear unabhängige Gittervektoren in dieser Kugel enthalten sind. Formal:

Definition 2.2.3 (Sukzessive Minima)**.** *Sei Λ ein Gitter der Dimension m. Das i-te* sukzessive Minimum *von Λ ist definiert wie folgt:*

$$\lambda_i(\Lambda) := \min\{r \,:\, \dim(\mathrm{span}(\Lambda \cap B(0,r))) \geq i\}.$$

Dabei bezeichnet $\mathrm{span}(M)$ *den von der Menge M aufgespannten Vektorraum.*

Für jedes sukzessive Minimum des Gitters Λ kann man zeigen, dass es Gittervektoren mit Länge $\lambda_i(\Lambda)$ gibt. Das erste sukzessive Minimum, $\lambda_1(\Lambda)$, entspricht gerade der Länge eines kürzesten, von 0 verschiedenen Gittervektors.

Die Werte, die die sukzessiven Minima annehmen, hängen von der für die Kugel $B(0,r)$ verwendeten Norm ab. Folgendes Beispiel soll diesen Sachverhalt verdeutlichen: Sei $\Lambda = \mathcal{L}(b_1, b_2)$ ein Gitter (siehe Abbildung 1), erzeugt durch die beiden Basisvektoren

$$b_1 = \begin{pmatrix} 2 \\ 0 \end{pmatrix} \text{ und } b_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Bezüglich der l_1-Norm ist b_1 ein kürzester, von 0 verschiedener Gittervektor und es gilt

$$\lambda_1(\Lambda) = \|b_1\|_1 = 2.$$

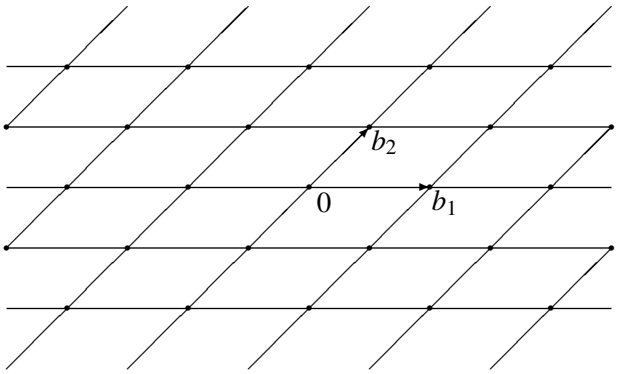

Abbildung 1: Das Gitter Λ

Bezüglich der l_2-Norm aber ist b_2 mit

$$\lambda_1(\Lambda) = \|b_2\|_2 = \sqrt{2}$$

ein kürzester Vektor, b_1 mit $\|b_1\|_2 = 2$ hingegen echt länger als b_2. Man sieht, dass sich nicht nur die Werte der sukzessiven Minima abhängig von der gewählten Norm ändern, sondern auch die Gittervektoren, die diese Werte als Länge annehmen.

Man kann eine obere Schranke für die Anzahl an Gittervektoren in Kugeln um 0 beweisen:

Lemma 2.2.4. *Sei Λ ein volldimensionales Gitter vom Rang n, und sei $r > 0$. Dann gilt:*

$$|\Lambda \cap B(0,r)| \leq \left(\frac{2r + \lambda_1(\Lambda)}{\lambda_1(\Lambda)}\right)^n.$$

Beweis. Alle Gittervektoren aus Λ haben einen Abstand von mindestens $d := \lambda_1(\Lambda)$. Denn der Abstand von zwei Gittervektoren wird bestimmt durch die Länge ihres Differenzvektors. Dieser ist ebenfalls ein Gittervektor, besitzt also mindestens eine Länge von $\lambda_1(\Lambda) = d$.

Da d der Mindestabstand zweier Gittervektoren ist, lässt sich um jeden Gittervektor eine offene Kugel vom Radius $\frac{d}{2}$ legen, ohne dass sich diese Kugeln überschneiden:

$$\forall v, w \in \Lambda,\ v \neq w : \tilde{B}\left(v, \frac{d}{2}\right) \cap \tilde{B}\left(w, \frac{d}{2}\right) = \emptyset.$$

Für die Kugeln um Gittervektoren, die in $\Lambda \cap B(0,r)$ enthalten sind, gilt zudem, dass sie in der Kugel

$$B\left(0, r + \frac{d}{2}\right)$$

enthalten sind.

Es kann also nicht mehr Elemente in $\Lambda \cap B(0,r)$ geben, als offene Kugeln vom Radius $\frac{d}{2}$ paarweise disjunkt in einer Kugel vom Radius $r+\frac{d}{2}$ angeordnet werden können. Die Kardinalität dieser Menge ist wiederum beschränkt durch den Quotienten aus dem Volumen von $B(0,r+\frac{d}{2})$ und dem Volumen der offenen Kugeln mit Radius $\frac{d}{2}$.

Das Volumen einer offenen Kugel stimmt mit dem einer abgeschlossen Kugel vom gleichem Radius überein. Da die Dimension des zugrundeliegenden Vektorraums n ist, gilt außerdem für die l_p-Normen nach [For84] die Beziehung

$$B(0,r) = r^n \cdot B(0,1).$$

Deshalb kann man nun wie folgt abschätzen:

$$|\Lambda \cap B(0,r)| \leq \frac{\mathrm{vol}(B(0,r+\frac{d}{2}))}{\mathrm{vol}(B(0,\frac{d}{2}))} = \frac{(r+\frac{d}{2})^n \,\mathrm{vol}(B(0,1))}{(\frac{d}{2})^n \,\mathrm{vol}(B(0,1))} = \frac{(r+\frac{d}{2})^n}{(\frac{d}{2})^n} = \left(\frac{2r+\lambda_1(\Lambda)}{\lambda_1(\Lambda)}\right)^n.$$

□

Zum Abschluss der Einführung werden einige Bezeichnungen für bestimmte, in dieser Arbeit relevante Klassen von Gittervektoren definiert.

Definition 2.2.5 (Gerader Gittervektor)**.** *Sei Λ ein Gitter mit Basis B, und $v = Bx \in \Lambda$ ein Gittervektor mit $x \in \mathbb{Z}^n$.*

Der Vektor v wird als gerader Gittervektor *bezeichnet, falls für jede Komponente x_i des Koeffizientenvektors x die Kongruenz*

$$x_i \equiv 0 \mod 2$$

erfüllt ist, x_i also eine gerade Zahl ist.

Ist dies nicht der Fall, ist also mindestens eine Komponente des Koeffizientenvektors eine ungerade Zahl, so wird v als ungerader Gittervektor *bezeichnet.*

Die Klassifikation von Gittervektoren in gerade und ungerade Gittervektoren ist basisunabhängig. Diese Beobachtung wird in folgendem Lemma bewiesen:

Lemma 2.2.6. *Sei Λ ein Gitter, und seien B und B' zwei Basen von Λ.*

Jeder Gittervektor v, der gerade ist bezüglich der Basis B, ist auch gerade bezüglich B'.

Beweis. Sei n der Rang von Λ, und v ein gerader Gittervektor bezüglich B. Es gilt folglich $v = Bx$ für einen Koeffizientenvektor $x \in \mathbb{Z}^n$, der in jeder Komponente kongruent zu $0 \mod 2$ ist. Es folgt $\frac{1}{2}x \in \mathbb{Z}^n$, und damit

$$\frac{1}{2}v = B\left(\frac{1}{2}x\right) \in \Lambda.$$

Da auch B' eine Gitterbasis von Λ ist, existiert auch bezüglich B' ein Koeffizientenvektor $y \in \mathbb{Z}^n$ für den Gittervektor $\frac{1}{2}v$. Es gilt also $\frac{1}{2}v = B'y$, und es folgt $v = B'(2y)$. Der Koeffizientenvektor von v bezüglich der Basis B' ist demnach der Vektor $2y$. Dieser ist aber in jeder Komponente kongruent $0 \mod 2$. v ist also auch bezüglich B' gerade. □

In diesem Beweis wurde implizit gezeigt, dass die Äquivalenz

$$v \text{ ist gerader Vektor in } \Lambda \Leftrightarrow \frac{1}{2}v \in \Lambda \tag{5}$$

gilt. „$\frac{1}{2}v \in \Lambda$“ ist also eine alternative Charakterisierung für gerade Gittervektoren.

Eine weitere Klasse von Gittervektoren sind die sogenannten *primitiven Gittervektoren*:

Definition 2.2.7 (Primitiver Gittervektor)**.** *Sei Λ ein Gitter, und $v \in \Lambda$ ein Gittervektor.*
Der Vektor v heißt primitiv*, wenn er kein ganzzahliges Vielfaches eines anderen Gittervektors ist, also für alle $n \in \mathbb{N}$ mit $n \geq 2$ gilt:*

$$\frac{1}{n}v \notin \Lambda.$$

Insbesondere sind alle primitiven Vektoren auch ungerade Vektoren, was direkt aus (5) folgt.

2.3 Das Problem des kürzesten Gittervektors und das Problem des nächsten Gittervektors

Bei Untersuchungen zur Berechnungskomplexität von Problemen ist es üblich und sinnvoll, sich auf „rationale Problemstellungen“ zu beschränken. Deshalb werden alle in diesem Abschnitt behandelten Probleme nur für *rationale Gitter*, also Gitter mit rationalen Basisvektoren, definiert werden.

Zwei bedeutende Probleme in der Theorie der Gitter sind das *Problem des kürzesten Gittervektors* und das *Problem des nächsten Gittervektors*. Diese Probleme werden nun formal definiert:

Definition 2.3.1 (Problem des kürzesten Gittervektors, SVP)**.** *Sei Λ ein rationales Gitter. Unter dem Problem des kürzesten Gittervektors, kurz* SVP *vom englischen „Shortest Vector Problem“, versteht man das Problem, einen kürzesten, von 0 verschiedenen Gittervektor $v \in \Lambda$ zu finden. Gesucht ist also ein Vektor $v \in \Lambda$ mit*

$$\|v\|_p = \min\left\{ \|u\|_p \,:\, u \in \Lambda \backslash \{0\} \right\}.$$

Man hätte SVP auch so definieren können, dass nach einem Gittervektor v mit $\|v\|_p = \lambda_1(\Lambda)$ gesucht wird. Diese Formulierung ist äquivalent, da das erste sukzessive Minimum gerade der Länge der kürzesten Vektoren entspricht.

Die oben definierte Form von SVP ist die sogenannte *Suchvariante*. Häufig von Interesse ist aber auch die *Entscheidungsvariante*. In dieser Version wird nicht nach einem kürzesten Vektor gesucht, sondern es wird geprüft, ob die kürzesten Vektoren eines Gitters eine vorgegebene Länge r überschreiten:

Definition 2.3.2 (Problem des kürzesten Gittervektors, SVP (Entscheidungsvariante))**.**
Sei (Λ, r) *ein Tupel, wobei* Λ *ein rationales Gitter, und* $r > 0$ *eine rationale Zahl ist.*
Das Tupel (Λ, r) *ist eine* YES*-Instanz von* SVP*, wenn ein Vektor* $v \in \Lambda \backslash \{0\}$ *existiert mit*

$$\|v\|_p \leq r.$$

Falls nicht, so ist das Tupel (Λ, r) *eine* NO*-Instanz.*

Die Entscheidungsvariante von SVP ist für die l_∞-Norm ein NP-vollständiges Problem. Der Beweis wurde 1981 von Emde Boas [vEB81] erbracht. Für alle anderen l_p-Normen ist die Frage nach der NP-Vollständigkeit noch ungelöst. Zumindest unter randomisierten Reduktionen jedoch ist es Ajtai [Ajt98] im Jahr 1998 gelungen zu zeigen, dass die Entscheidungsvariante von SVP für die l_2-Norm ein NP-hartes Problem ist.

Definition 2.3.3 (Problem des nächsten Gittervektors, CVP (Suchvariante))**.** *Sei* Λ *ein rationales Gitter der Dimension m, und* $t \in \mathbb{Q}^m$ *ein rationaler Zielvektor. Unter dem Problem des nächsten Gittervektors, kurz* CVP *vom englischen „Closest Vector Problem", versteht man das Problem, einen Gittervektor* $v \in \Lambda$ *mit minimalem Abstand zum Zielvektor t zu finden.*
Gesucht ist also ein Vektor $v \in \Lambda$ *mit*

$$\|v - t\|_p = \min\left\{ \|u - t\|_p \ : \ u \in \Lambda \right\}.$$

Analog zu SVP kann man auch hier eine Entscheidungsvariante definieren:

Definition 2.3.4 (Problem des nächsten Gittervektors, CVP (Entscheidungsvariante))**.**
Sei (Λ, t, r) *ein Tripel, wobei* Λ *ein rationales Gitter der Dimension m,* $t \in \mathbb{Q}^m$ *ein rationaler Vektor, und* $r > 0$ *eine rationale Zahl ist.*
Das Tripel (Λ, t, r) *ist eine* YES*-Instanz von* CVP*, wenn es einen Vektor* $v \in \Lambda$ *gibt mit*

$$\|v - t\| \leq r.$$

Falls nicht, so ist das Tripel (Λ, t, r) *eine* NO*-Instanz.*

Die Entscheidungsversion von CVP ist NP-vollständig für alle l_p-Normen. Auch dieses Aussage wurde 1981 von Emde Boas [vEB81] gezeigt. Für ganzzahlige Gitter, also Gitter mit ganzzahliger Basis, und ganzzahlige Zielvektoren t ist ein schöner Beweis in Kapitel 3.2 von [MG02] nachzulesen.

In Kapitel 3.1 von [MG02] wird für die euklidische Norm gezeigt, dass die Suchvariante von CVP nicht wesentlich schwieriger ist als die Entscheidungsversion. Das bedeutet, dass die Suchvariante von CVP unter Verwendung eines Orakels für die Entscheidungsvariante in polynomieller Zeit lösbar ist.

Die schnellsten, zur Zeit bekannten Algorithmen zum Lösen der Suchvariante von SVP für Gitter vom Rang n sind randomisierte Algorithmen mit Laufzeit $2^{O(n)}$. Ein Algorithmus für die

l_2-Norm wurde von Ajtai, Kumar und Sivakumar im Jahre 2001 veröffentlicht [AKS01]. Blömer und Naewe haben im Jahr 2007 einen allgemeineren Ansatz entwickelt, aus dem randomisierte Algorithmen zum Lösen der Suchvariante von SVP und zur Approximation von CVP mit einem Faktor $1+\varepsilon$, für ein $\varepsilon > 0$, für beliebige l_p-Normen mit Laufzeit $2^{O(n)}$ hervorgehen [BN07].

Der schnellste bekannte deterministische Algorithmus zur Berechnung einer Lösung der Suchvariante von CVP stammt von Blömer aus dem Jahr 2000. Für ein Gitter vom Rang n löst der Algorithmus das CVP in Zeit $n! \cdot s^{O(1)}$, wobei s die Kodierungslänge der Eingabe bezeichnet [Blö00].

Die besten, bisher bekannten Polynomialzeitalgorithmen für die Approximation von SVP und CVP liefern im schlechtesten Fall Ergebnisse, die um einen Faktor von der exakten Lösung abweichen, der einfach exponentiell im Rang des Gitters ist.

Für SVP leistet dies der im Jahre 1982 von Lenstra, Lenstra und Lovász [LLL82] entwickelte LLL-Algorithmus. Basierend auf dem LLL-Algorithmus approximiert der Nearest Plane Algorithmus von Babai [Bab86] aus dem Jahre 1986 CVP.

Wie in der Einleitung angekündigt, befasst sich diese Arbeit mit der Berechnungskomplexität der Eindeutigkeitsvarianten von SVP und CVP. Was genau unter diesen Eindeutigkeitsvarianten zu verstehen ist, wird im Folgenden erklärt.

Ein Gitter kann niemals nur einen kürzesten, von 0 verschiedenen Vektor enthalten. Der Grund ist, dass das Gitter mit einem kürzesten Vektor v auch immer sein additiv Inverses $-v$ enthält, und beide Vektoren dieselbe Länge besitzen.

Die Eindeutigkeit der Lösung von SVP ist also so zu verstehen, dass in einem Gitter nur zwei kürzeste Vektoren existieren. Nämlich ein Vektor v und sein Inverses $-v$.

Einfacher lässt sich dieser Zusammenhang mit Hilfe von Äquivalenzklassen modellieren. Sei also $\rho \subseteq \Lambda \times \Lambda$ eine Äquivalenzrelation, definiert wie folgt:

$$v \,\rho\, w \;:\Longleftrightarrow\; u = v = 0 \;\vee\; (u, v \neq 0 \text{ und } u, v \text{ sind linear abhängig}). \tag{6}$$

Man prüft leicht nach, dass ρ reflexiv, symmetrisch und transitiv ist, es sich also um eine Äquivalenzrelation handelt. Es wird von nun an folgende Schreibweise für die Äquivalenzklassen nach ρ verwendet:

$$\text{Für } v \in \Lambda \text{ bezeichnet } [v]_\rho \text{ die Äquivalenzklasse von } v \text{ nach } \rho.$$

Wenn in Zukunft eine Menge M von Gittervektoren als eindeutig bezüglich ρ bezeichnet wird, so bedeutet dies, dass alle Vektoren dieser Menge in derselben Äquivalenzklasse enthalten sind:

$$\exists [\hat{v}]_\rho \forall v \in M : v \in [\hat{v}]_\rho.$$

Unter Verwendung der Äquivalenzrelation ρ wird nun die Eindeutigkeitsvariante UNIQUE-SVP definiert:

Definition 2.3.5 (UNIQUE-SVP (Suchvariante))**.** *Sei Λ ein rationales Gitter mit der Eigenschaft, dass die Menge der kürzesten, von Null verschiedenen Vektoren*

$$S := \left\{ u \in \Lambda \ : \ \|u\|_p = \lambda_1(\Lambda) \right\}$$

eindeutig bezüglich ρ ist.
Unter UNIQUE-SVP *versteht man das Problem, einen kürzesten, von 0 verschiedenen Vektor $v \in S$ im Gitter Λ zu finden.*

Diese Definition über die Äquivalenzrelation entspricht der Forderung, dass nur ein Vektor v und sein additiv Inverses $-v$ kürzeste Vektoren sind. In jeder Äquivalenzklasse sind primitive Vektoren und ihre ganzzahligen Vielfachen enthalten. Wenn nun alle kürzesten Gittervektoren in einer Äquivalenzklasse enthalten sind, so sind sie paarweise linear abhängig, also Vielfache voneinander. Es gibt aber nur zwei längenerhaltende Skalierungen für Vektoren ungleich 0, nämlich die Skalierungen um die Faktoren 1 und -1. Deshalb kann es nur zwei kürzeste Vektoren in der Äquivalenzklasse geben.

Auch hier kann man wieder eine Entscheidungsvariante definieren:

Definition 2.3.6 (UNIQUE-SVP (Entscheidungsvariante))**.** *Sei (Λ, r) ein Tupel, wobei Λ ein rationales Gitter, und $r > 0$ eine rationale Zahl ist. Λ erfüllt die Eigenschaft, dass die Menge der von 0 verschiedenen Gittervektoren mit Länge höchstens r,*

$$\left\{ u \in \Lambda \backslash \{0\} \ : \ \|u\|_p \leq r \right\},$$

eindeutig bezüglich ρ ist.

Das Tupel (Λ, r) ist eine YES*-Instanz von* UNIQUE-SVP*, wenn ein $v \in \Lambda \backslash \{0\}$ existiert mit $\|v\|_p \leq r$. Falls nicht, so ist das Tupel (Λ, r) eine* NO*-Instanz.*

Kumar und Sivakumar haben 1999 in [KS99] bewiesen, dass die Entscheidungsvariante von UNIQUE-SVP unter der l_2-Norm, genau wie SVP, ein NP-hartes Problem unter randomisierten Reduktionen ist.

In Kapitel 4 wird die von Kumar und Sivakumar verwendete Beweistechnik aufgegriffen, um zu zeigen, dass für l_p-Normen mit $p \in (1, \infty)$ die Suchvarianten der Probleme SVP und UNIQUE-SVP bis auf polynomielle Abweichung die gleiche Komplexität besitzen.

Es folgt nun die Definition der Eindeutigkeitsvariante für CVP. Hier gibt es Gitter und Zielvektoren, für welche die Lösung echt eindeutig ist, also genau ein Gittervektor mit minimalem Abstand zum Zielvektor existiert.

Definition 2.3.7 (UNIQUE-CVP)**.** *Sei $t \in \mathbb{Q}^m$ ein Zielvektor, und Λ ein rationales Gitter der Dimension m, welches genau einen Gittervektor mit minimalem Abstand zu t besitzt.*
Sei also

$$D_t := \min\left\{ \|u-t\|_p \ : \ u \in \Lambda \right\},$$

und enthalte die Menge

$$\left\{ u \in \Lambda \ : \ \|u-t\|_p = D_t \right\}$$

genau ein Element.
Unter UNIQUE-CVP *versteht man das Problem, den Vektor $v \in \Lambda$ mit $\|v-t\|_p = D_t$ zu finden.*

Kapitel 5 beschäftigt mit der Komplexität von UNIQUE-CVP. Anders als beim Problem des kürzesten Gittervektors in Kapitel 4 wird hier jedoch nicht direkt das Verhältnis zu CVP untersucht, sondern zum Problem FEW-CVP$_c$. Dieses Problem ist abhängig von einer ganzzahligen positiven Konstante c definiert wie folgt:

Definition 2.3.8 (FEW-CVP$_c$)**.** *Sei $t \in \mathbb{Q}^m$ ein Zielvektor, und Λ ein rationales Gitter vom Rang n und Dimension m, welches höchstens 2^{cn} Gittervektoren mit minimalem Abstand zu t besitzt.*
Sei also

$$D_t := \min\left\{ \|u-t\|_p \ : \ u \in \Lambda \right\},$$

und übersteige die Kardinalität der Menge

$$\left\{ u \in \Lambda \ : \ \|u-t\|_p = D_t \right\}$$

die Zahl 2^{cn} nicht.
Unter FEW-CVP$_c$ *versteht man das Problem, einen Vektor $v \in \Lambda$ mit $\|v-t\|_p = D_t$ zu finden.*

Wie auch UNIQUE-CVP unterscheidet sich dieses Problem also von CVP nur durch eine Restriktion an die maximale Anzahl an Gittervektoren mit minimalem Abstand zum Zielvektor.

In Kapitel 5 wird für die l_p-Normen mit $p \in (1,\infty)$ gezeigt, dass dieses Problem nicht schwieriger als UNIQUE-CVP ist, indem eine randomisierte Reduktion von FEW-CVP$_c$ auf UNIQUE-CVP konstruiert wird. Darüber hinaus wird in Abschnitt 5.3 gezeigt, dass die Probleme FEW-CVP$_c$ und CVP für die l_p-Normen mit $p \in (1,\infty)$ identisch sind. Implizit erhält man also, dass auch CVP für diese l_p-Normen nicht schwieriger als UNIQUE-CVP ist.

3 Ein randomisierter Algorithmus zur Gittererzeugung

Dieses Kapitel dient der Vorbereitung der Reduktionen von SVP und FEW-CVP$_c$ auf ihre Eindeutigkeitsvarianten, mit denen sich die Kapitel 4 und 5 befassen werden.

Diese Reduktionen basieren auf einem randomisierten Algorithmus, der zu einem rationalen Gitter ein System von Untergittern erzeugt. Im folgenden Abschnitt 3.1 wird dieser Algorithmus spezifiziert. Der Abschnitt 3.2 befasst sich mit der stochastischen Analyse einzelner (randomisierter) Konstruktionsschritte, und führt ein Werkzeug zum Beweis von Eindeutigkeitsaussagen für bestimmte stochastische Prozesse ein.

3.1 Der Algorithmus LATTICEGENERATOR

Die Idee der Reduktionen aus den Kapiteln 4 und 5 ist es, zum Eingabegitter randomisiert ein System von Untergittern zu erzeugen und anschließend zufällig eines der erzeugten Untergitter zu wählen, welches mit genügend hoher Wahrscheinlichkeit ein zulässiges Reduktionsergebnis ist.

Beide Reduktionen verwenden denselben Gittererzeugungsalgorithmus. Bei Eingabe eines Tupels (L,γ), bestehend aus einem rationalen Gitter L vom Rang n, und einer natürlichen Zahl γ, erzeugt der Algorithmus ein Untergittersystem

$$L = L_0 \supseteq L_1 \supseteq \ldots \supseteq L_\gamma$$

auf folgende Weise:

- $L_0 = L$,
- Für jedes $0 < k \leq \gamma$ wird gemäß der Gleichverteilung und unabhängig von allen vorherigen Wahlen eine Menge $W \subseteq \{1,\ldots,n\}$ gewählt, und der charakteristische Vektor $w \in \mathbb{Z}^n$ von W berechnet, welcher komponentenweise definiert ist durch

$$w_i := \begin{cases} 1, \text{ falls } i \in W \\ 0, \text{ sonst.} \end{cases}$$

  Anschließend wird

$$L_k := \{B_{k-1}x \,:\, x \in \mathbb{Z}^n, \langle x, w\rangle \equiv 0 \mod 2\} \tag{7}$$

  gebildet, wobei B_{k-1} eine Basis von L_{k-1} bezeichnet.

Diese Beschreibung des Algorithmus wirft zwei Fragen auf: Zum Einen ist zunächst unklar, dass die in (7) definierten Mengen überhaupt Gitter sind. Zum Anderen wird eine effiziente Methode zur Berechnung von Basen dieser Gitter benötigt, da eine Basis des Vorgängergitters für die Konstruktion des jeweiligen Folgegitters erforderlich ist.

Zur Beantwortung dieser Fragen wird folgendes Lemma beitragen:

Lemma 3.1.1. *Sei Λ ein Gitter vom Rang n mit Basis $B = [b_1, \ldots, b_n]$, und sei $W \subseteq \{1, \ldots, n\}$ eine nicht leere Menge mit zugehörigem charakteristischen Vektor $w \in \mathbb{Z}^n$. Dann ist*

$$\Lambda_w := \{Bx \,:\, x \in \mathbb{Z}^n, \langle x, w\rangle \equiv 0 \mod 2\}$$

ein Untergitter von Λ vom Rang n.

Ferner gilt für jedes $i \in W$, dass die Menge

$$B' = [b'_1, \ldots, b'_n] \text{ mit } b'_j := \begin{cases} b_j, & \text{falls } j \notin W \\ b_j - b_i, & \text{falls } j \in W, j \neq i \\ 2b_i, & \text{falls } i = j \end{cases}$$

eine Basis von Λ_w bildet.

Beweis. Sei $i \in W \neq \emptyset$ beliebig. Die Beweisstrategie ist Folgende: Zunächst zeigt man, dass die Elemente der Menge B' linear unabhängig sind, und damit eine zulässige Gitterbasis bilden. Ist dies erfolgt, so weist man nach, dass die Identität $\Lambda_w = \mathcal{L}(B')$ gilt. Denn dann lässt sich schließen, dass Λ_w ein Gitter vom Rang n mit Basis B' ist. Die noch zu zeigende Untergittereigenschaft erhält man schließlich aus der Tatsache, dass Λ_w nach Definition eine Teilmenge von Λ ist.

Zum Beweis der linearen Unabhängigkeit der Elemente von B' genügt es zu zeigen, dass die Matrix B' nicht singulär ist, ihre Determinante also von Null verschieden ist. Es gilt

$$\det(B') = 2\det(\underbrace{[b'_1, b'_2, \cdots, \frac{1}{2}b'_i, \cdots, b'_n]}_{:=M}).$$

Aus der Definition der b'_j wird deutlich, dass man die Matrix M aus der Matrix B durch elementare Spaltenumformungen erhält, nämlich der Subtraktion der i-ten Spalte von B von allen anderen Spalten, deren Index in W enthalten ist. Demnach sind die Determinanten von M und B identisch. Als Gitterbasis ist die Matrix B nicht singulär. Zusammen erhält man:

$$\det(B') = 2\det(M) = 2\det(B) \neq 0.$$

Die Determinante von B' ist also von Null verschieden. Die Spaltenvektoren von B' sind demnach linear unabhängig und bilden eine Gitterbasis des Gitters $\mathcal{L}(B')$.

Es bleibt also nur noch zu zeigen, dass die Identität

$$\Lambda_w = \mathcal{L}(B')$$

erfüllt ist. Dies ist genau dann der Fall, wenn beide Inklusionen $\Lambda_w \subseteq \mathcal{L}(B')$ und $\Lambda_w \supseteq \mathcal{L}(B')$ gelten.

Zum Beweis der Beziehung $\Lambda_w \supseteq \mathcal{L}(B')$ zeigt man für jeden Vektor $v \in \mathcal{L}(B')$ dass er auch in Λ_w enthalten ist. Sei also $v \in \mathcal{L}(B')$, und $x' \in \mathbb{Z}^n$ der Koeffizientenvektor von v bezüglich B'. Es gilt also $v = B'x'$.

Durch Einsetzen der Definition der Vektoren b'_j kann man den Koeffizientenvektor von v bezüglich der Basis B berechnen:

$$\begin{aligned}
v &= B'x' \\
&= \sum_{j=1}^{n} x'_j b'_j \\
&= \sum_{j \notin W} x'_j b_j + \sum_{j \in W, j \neq i} x'_j (b_j - b_i) + 2x'_i b_i \\
&= \sum_{j \notin W} \underbrace{x'_j}_{=:x_j} b_j + \sum_{j \in W, j \neq i} \underbrace{x'_j}_{=:x_j} b_j + \underbrace{(2x'_i - \sum_{j \in W, j \neq i} x'_j)}_{=:x_i} b_i \\
&= Bx, \text{ für ein } x \in \mathbb{Z}^n
\end{aligned}$$

Nun gilt:

$$\langle x, w \rangle = \sum_{j=1}^{n} x_j w_j = \sum_{j \in W} x_j = \sum_{j \in W, j \neq i} x'_j + 2x'_i - \sum_{j \in W, j \neq i} x'_j = 2x'_i \equiv 0 \mod 2.$$

Nach Definition von Λ_w gilt demnach $v \in \Lambda_w$ und es folgt $\Lambda_w \supseteq \mathcal{L}(B')$.

Die andere Beziehung, $\Lambda_w \subseteq \mathcal{L}(B')$, zeigt man wie folgt:
Für jeden Vektor $v \in \Lambda_w$ gibt es einen Koeffizientenvektor $x \in \mathbb{Z}^n$ mit $v = Bx$, welcher

$$\langle x, w \rangle = \sum_{j \in W} x_j \equiv 0 \mod 2 \tag{8}$$

erfüllt.

Durch Rechnen und Einsetzen der Definition der b'_j lässt sich v als ganzzahlige Linearkombination durch die Basis B' wie folgt darstellen:

$$\begin{aligned}
v &= \sum_{j=1}^{n} x_j b_j \\
&= \sum_{j\notin W} x_j b_j + \sum_{j\in W, j\neq i} x_j b_j + x_i b_i \\
&= \sum_{j\notin W} x_j b_j + \sum_{j\in W, j\neq i} x_j b_j - \sum_{j\in W, j\neq i} x_j b_i + \sum_{j\in W, j\neq i} x_j b_i + x_i b_i \\
&= \sum_{j\notin W} x_j b'_j + \sum_{j\in W, j\neq i} x_j b'_j + \sum_{j\in W, j\neq i} x_j b_i + x_i b_i \\
&= \sum_{j\notin W} x_j b'_j + \sum_{j\in W, j\neq i} x_j b'_j + \underbrace{\sum_{j\in W} x_j}_{\equiv 0 \mod 2 \text{ nach (8)}} b_i \\
&= \sum_{j\notin W} x_j b'_j + \sum_{j\in W, j\neq i} x_j b'_j + \underbrace{\left(\frac{1}{2}\sum_{j\in W} x_j\right)}_{\in\mathbb{Z}} b'_i
\end{aligned}$$

Es folgt $v \in \mathcal{L}(B')$, und damit $\Lambda_w \subseteq \mathcal{L}(B')$.

Zusammen mit $\Lambda_w \supseteq \mathcal{L}(B')$, was bereits oben gezeigt wurde, folgt Mengengleichheit. □

Das Lemma 3.1.1 liefert die Antworten auf die oben formulierten Fragen. Nun lässt sich nämlich zeigen, dass die in (7) definierten Mengen tatsächlich Gitter sind. Falls $W \neq \emptyset$ gilt, liefert das Lemma diese Aussage. Im Falle $W = \emptyset$ ist w der Nullvektor und es folgt $\langle x, w\rangle = 0$ für alle $x \in \mathbb{Z}^n$. In diesem Fall stimmt die neue Menge also mit ihrem Vorgänger überein und übernimmt damit auch dessen Gittereigenschaft. Lemma 3.1.1 liefert außerdem effizient zu berechnende Basen für diese Gitter.

Es folgt nun eine formale Beschreibung des Algorithmus (siehe Algorithmus 1). Er erhält als Eingabe ein Tupel (B, γ). B ist eine geeignete Kodierung einer Basis eines rationalen Gitters L vom Rang n und Dimension m, für welches das Untergittersystem erzeugt werden soll. Der Parameter γ ist eine natürliche Zahl und legt die Anzahl der zu erzeugenden Gitter fest. Die Ausgabe des Algorithmus ist das generierte Untergittersystem, in Form einer Liste von Basen dieser Gitter.

Die Wahl der Teilmenge W in Zeile 4 soll in jedem Schritt unabhängig von allen vorherigen Wahlen geschehen. In Zeile 8 wird i auf das Minimum von W gesetzt. Diese Wahl ist willkürlich. Man hätte auch jedes andere $i \in W$ verwenden können, denn nach Lemma 3.1.1 erhält man mit jedem beliebigen $i \in W$ dasselbe Gitter. Die Bezeichnung $\mathrm{col}(B, j)$ in den Zeilen 11, 14 und 16 soll auf die j-te Spalte der Matrix B verweisen.

Algorithmus 1 Randomisierter Algorithmus LATTICEGENERATOR zur Gittererzeugung

function LATTICEGENERATOR(B, c)
  $B_0 \leftarrow B$
  for $k = 1, \ldots, \gamma$ **do**
    Wähle zufällig gleichverteilt eine Teilmenge W von $\{1, \ldots, n\}$
    if $W = \emptyset$ **then**
      $B_k \leftarrow B_{k-1}$
    else
      $i \leftarrow \min W$
      for $j = 1, \ldots, n$ **do**
        if $j \notin W$ **then**
          $b_j \leftarrow \mathrm{col}(B_{k-1}, j)$
        else
          if $j \neq i$ **then**
            $b_j \leftarrow \mathrm{col}(B_{k-1}, j) - \mathrm{col}(B_{k-1}, i)$
          else
            $b_j \leftarrow 2\,\mathrm{col}(B_{k-1}, i)$
          end if
        end if
      end for
      $B_k \leftarrow [b_1, \ldots, b_n]$
    end if
  end for
  return $B_0, \ldots, B_\gamma$
end function

Im Folgenden wird der k-te Durchlauf des Schleifenrumpfes, der sich von Zeile 4 bis Zeile 21 erstreckt, als der *k-te Konstruktionsschritt* des Algorithmus bezeichnet.

Der folgende Satz fasst die Ergebnisse dieses Abschnittes zusammen:

Satz 3.1.2. *Der Algorithmus* LATTICEGENERATOR *erzeugt bei Eingabe eines Tupels (B,γ), bestehend aus einer Gitterbasis B eines rationalen Gitters L vom Rang n und Dimension m, und einer Zahl $\gamma \in \mathbb{N}$, Basen für ein Untergittersystem*

$$L = L_0 \supseteq L_1 \supseteq \ldots \supseteq L_\gamma$$

in folgender Weise:

(a) $L_0 = L$,

(b) Für jedes $0 < k \leq \gamma$ *gilt*

$$L_k = \{B_{k-1}x \,:\, x \in \mathbb{Z}^n, \langle x, w\rangle \equiv 0 \mod 2\}, \tag{9}$$

wobei B_{k-1} eine Basis von L_{k-1}, und $w \in \mathbb{Z}^n$ der charakteristische Vektor einer Menge W ist, die gemäß der Gleichverteilung unabhängig von allen vorherigen Wahlen als Teilmenge von $\{1,\ldots,n\}$ gewählt wird.

Die Laufzeit des Algorithmus ist polynomiell beschränkt in m, n und γ.

Beweis. Es wird zunächst gezeigt, dass der Konstruktionsprozess des Algorithmus den Eigenschaften (a) und (b) genügt, anschließend erfolgt die Laufzeitanalyse.

In Zeile 2 des Algorithmus wird die Basis von L_0 als die Basis B von L festgesetzt. Es gilt also

$$L_0 = L,$$

wie in Eigenschaft (a) gefordert.

Im k-ten Konstruktionsschritt, $0 < k \leq \gamma$, wird eine Basis des Gitters L_k erzeugt. Man muss sich davon überzeugen, dass diese Konstruktion der Eigenschaft (b) genügt. In Zeile 4 wird die Menge W zufällig gleichverteilt und unabhängig von allen vorherigen Wahlen als Teilmenge von $\{1,\ldots,n\}$ gewählt. Damit ist sie so gewählt, wie in Eigenschaft (b) gefordert.

Falls $W = \emptyset$ gilt, so ist w der Nullvektor. Es gilt also $\langle x, w\rangle = 0$ für alle $x \in \mathbb{Z}^n$. Eigenschaft (b) fordert für diesen Fall also $L_k = L_{k-1}$. Genau dies setzt der Algorithmus in Zeile 6 um, indem er als Basis für L_k die Basis von L_{k-1} festsetzt.

Falls $W \neq \emptyset$ gilt, so folgt mit Lemma 3.1.1, dass für die in den Zeilen 9 bis 20 generierte Basis B_k gilt:

$$L_k = \mathcal{L}(B_k) = \{B_{k-1}x \,:\, x \in \mathbb{Z}^n, \langle x, w\rangle \equiv 0 \mod 2\}.$$

Auch in diesem Fall ist die Eigenschaft (b) also erfüllt.

Das Gittersystem $L_0,\ldots,L_\gamma$ erfüllt also die Konstruktionsbedingungen (a) und (b).

Nun zur Laufzeitanalyse: In Zeile 2 wird eine $m \times n$-Matrix kopiert. Es sind also mn Einträge zu schreiben. Dafür benötigt der Algorithmus Zeit $O(mn)$.

In jedem Konstruktionsschritt wird zufällig gleichverteilt eine Teilmenge $W \subseteq \{1,\ldots,n\}$ gewählt. Dazu ist der Algorithmus in Zeit $O(n)$ in der Lage. Falls in einem Konstruktionsschritt $W = \emptyset$ gilt, so wird erneut eine Matrix kopiert. Wie oben bereits gesehen, wird dazu Zeit $O(mn)$ benötigt. Gilt hingegen $W \neq \emptyset$, so wird die neue Basis vektorweise konstruiert. Jeder der n Vektoren entsteht entweder durch Kopieren eines anderen Vektors, durch Bilden der Differenz zweier Vektoren oder durch Skalarmultiplikation. Jede dieser drei Operationen benötigt m rationale Rechenoperationen. Alle n Vektoren können also in Zeit $O(mn)$ berechnet werden. Diese Vektoren werden anschließend in eine Matrix kopiert, was wiederum in Zeit $O(mn)$ durchführbar ist. Insgesamt wird pro Konstruktionsschritt also Zeit $O(mn)$ benötigt.

Der Algorithmus führt γ Konstruktionsschritte durch. Damit beträgt die Gesamtlaufzeit für den Konstruktionsprozess $O(mn\gamma)$.

Bei der Ausgabe in Zeile 23 sind $mn\gamma$ Matrixeinträge zu kodieren. Die hierfür benötigte Zeit ist beschränkt durch $O(mn\gamma)$.

Der Algorithmus LATTICEGENERATOR ist also polynomiell zeitbeschränkt in m, n und γ. □

3.2 Wahrscheinlichkeitsanalyse

Wie schon zu Beginn des Kapitels angekündigt, soll in diesem Abschnitt der Konstruktionsprozess des Algorithmus LATTICEGENERATOR aus stochastischer Sicht betrachtet werden.

Benötigt werden Aussagen über die „Überlebenswahrscheinlichkeit“ bestimmter Gittervektoren in einem Konstruktionsschritt. Also die Wahrscheinlichkeit dafür, dass ein Gittervektor, der in einem Gitter L_{k-1} enthalten ist, bei der Konstruktion von Gitter L_k nicht verworfen wird. Des Weiteren soll die Korrelation der „Überlebensereignisse“ zweier Gittervektoren pro Konstruktionsschritt untersucht werden.

Man erinnere sich an die Klassifikation von Gittervektoren als *gerade* und *ungerade* Vektoren in Definition 2.2.5: Gerade Gittervektoren sind diejenigen Vektoren, deren Koeffizientenvektoren bezüglich einer Basis in jeder Komponente kongruent 0 modulo 2 sind.

Nach Definition der Gitter L_k in (9) wird ein Vektor $v \in L_{k-1}$ mit Koeffizientenvektor x im k-ten Konstruktionsschritt genau dann nicht verworfen, wenn die Kongruenz $\langle x, w\rangle \equiv 0 \mod 2$ erfüllt ist. Ist v ein gerader Vektor, so ist dies offensichtlich unabhängig von w immer erfüllt, da dann $\langle x, w\rangle$ ein ganzzahliges Vielfaches von 2 ist. Gerade Vektoren „überleben“ einen Konstruktionsschritt also mit Wahrscheinlichkeit 1. In den späteren Kapiteln wird jedoch vor allem die „Überlebenswahrscheinlichkeit“ der ungeraden Vektoren interessant sein. Diese wird im folgenden Lemma untersucht:

Lemma 3.2.1. *Sei Λ ein Gitter vom Rang n mit Basis B. Ist w der charakteristische Vektor einer zufällig gleichverteilt gewählten Menge $W \subseteq \{1,\ldots,n\}$, und wird das Untergitter Λ_w in Abhängigkeit von W konstruiert wie folgt:*

$$\Lambda_w = \{Bx \,:\, x \in \mathbb{Z}^n, \langle x, w\rangle \equiv 0 \mod 2\},$$

so gilt für jeden ungeraden Gittervektor $v \in \Lambda$:

$$Pr[v \in \Lambda_w] = \frac{1}{2}$$

(gemäß der Wahl von W).

Beweis. Sei $v \in \Lambda$ ein ungerader Gittervektor, und sei x der Koeffizientenvektor von v bezüglich der Basis B. Da v ungerade ist, besitzt sein Koeffizientenvektor x mindestens eine ungerade Komponente. Sei j der größte Index dieser Komponenten:

$$j := \max\{i \,:\, 1 \leq i \leq n,\, x_i \equiv 1 \mod 2\}. \tag{10}$$

Nun gilt

$$\langle x, w\rangle = \sum_{i=1}^{n} x_i w_i \overset{(*)}{\equiv} \sum_{i=1}^{j-1} x_i w_i + w_j \mod 2. \tag{11}$$

Die Kongruenz $(*)$ ist gültig, da nach (10) die Kongruenz $x_j \equiv 1 \mod 2$ erfüllt ist, und zudem $x_i \equiv 0 \mod 2$ gilt für jedes i mit $n \geq i > j$.

Da W gemäß der Gleichverteilung als Teilmenge von $\{1,\ldots,n\}$ gewählt wird, gilt

$$Pr[j \in W] = \frac{1}{2},$$

und damit

$$Pr[w_j = 1] = \frac{1}{2}. \tag{12}$$

Es können zwei Fälle eintreten, nämlich $\sum_{i=1}^{j-1} x_i w_i \equiv 0 \mod 2$ und $\sum_{i=1}^{j-1} x_i w_i \equiv 1 \mod 2$.

Im ersten Fall tritt nach (11) das Ereignis $\langle x, w\rangle \equiv 0 \mod 2$ genau dann ein, wenn auch das Ereignis $w_j = 0$ eintritt. Letzteres Ereignis tritt als Gegenereignis von (12) mit Wahrscheinlichkeit $\frac{1}{2}$ ein.

Im zweiten Fall gilt entsprechend: Das Ereignis $\langle x, w\rangle \equiv 0 \mod 2$ tritt genau dann ein, wenn auch das Ereignis $w_j = 1$ eintritt. Letzteres ist nach (12) mit Wahrscheinlichkeit $\frac{1}{2}$ der Fall.

In beiden Fällen erhält man also

$$Pr[\langle x, w\rangle \equiv 0 \mod 2] = \frac{1}{2}.$$

Nach Definition von Λ_w gilt $v \in \Lambda_w$ genau dann, wenn die Kongruenz

$$\langle x, w\rangle \equiv 0 \mod 2$$

erfüllt ist. Man erhält also

$$Pr[v \in \Lambda_w] = \frac{1}{2}.$$

□

Es folgt die Untersuchung der Korrelation der „Überlebensereignisse“ von Gittervektoren im k-ten Konstruktionsschritt, also die Untersuchung der Korrelation der Ereignisse $[u \in L_k]$ und $[v \in L_k]$ für zwei Gittervektoren u und v aus L_{k-1}. Sind sowohl u, als auch v und der Differenzvektor $v-u$ ungerade Gittervektoren in L_{k-1}, so sind diese Ereignisse unabhängig:

Lemma 3.2.2. *Sei Λ ein Gitter vom Rang n mit Basis B. Ist w der charakteristische Vektor einer zufällig gleichverteilt gewählten Menge $W \subseteq \{1,\ldots,n\}$, und wird das Untergitter Λ_w in Abhängigkeit von W konstruiert wie folgt:*

$$\Lambda_w = \{Bx \,:\, x \in \mathbb{Z}^n, \langle x,w\rangle \equiv 0 \mod 2\},$$

so gilt für alle ungeraden Gittervektoren u und v aus Λ, deren Differenzvektor $v-u$ ebenfalls ungerade in Λ ist, dass die Ereignisse

$$[u \in \Lambda_w] \text{ und } [v \in \Lambda_w]$$

stochastisch unabhängig sind (gemäß der Wahl von W).

Beweis. Seien $u = Bx$ und $v = By$ mit $x,y \in \mathbb{Z}^n$ ungerade Gittervektoren von Λ mit ungeradem Differenzvektor $v-u$.

Es gibt also ein $j \in \{1,\ldots,n\}$, so dass $y_j - x_j$ eine ungerade Zahl ist. Das bedeutet, dass entweder y_j oder x_j ungerade ist. Sei ohne Einschränkung y_j ungerade, und damit x_j gerade.

Ob $j \in W$ oder $j \notin W$ gilt, hat demnach keinen Einfluss auf das Ereignis $u \in \Lambda_w$. Sei $\mathcal{M}$ die Menge der Teilmengen $W \subseteq \{1,\ldots,n\}$, für die $u \in \Lambda_w$ gilt. Man macht sich leicht klar, dass für die Hälfte der Mengen $W \in \mathcal{M}$ gilt, dass $j \in W$ erfüllt ist, und für die andere Hälfte $j \notin W$ gilt.

Es soll nun die Wahrscheinlichkeit für das Ereignis

$$[v \in \Lambda_w \mid u \in \Lambda_w]$$

untersucht werden.

Man kann ähnlich zum Beweis von Lemma 3.2.1 vorgehen. Es gilt

$$\langle y,w\rangle = \sum_{i=1}^{n} y_i w_i \equiv \sum_{i=1, i\neq j}^{n} y_i w_i + w_j \mod 2,$$

da $y_j \equiv 1 \mod 2$ gilt.

Man unterscheidet wieder die Fälle $\sum_{i=1,i\neq j}^{n} y_i w_i \equiv 0 \mod 2$ und $\sum_{i=1,i\neq j}^{n} y_i w_i \equiv 1 \mod 2$.

Im ersten Fall gilt $\langle y,w\rangle \equiv 0 \mod 2$ genau dann, wenn $w_j = 0$, also $j \notin W$ gilt. Im zweiten Fall erhält man entsprechend $\langle y,w\rangle \equiv 0 \mod 2$ genau dann, wenn $j \in W$ erfüllt ist.

Gelte nun $u \in \Lambda_w$. Es folgt $W \in \mathcal{M}$. Da W gemäß der Gleichverteilung gewählt wurde, ist jede Wahl von $W \in \mathcal{M}$ gleichwahrscheinlich. Wie oben bereits gesehen, gilt $j \in W$ und $j \notin W$ jeweils für die Hälfte der Mengen aus $\mathcal{M}$. Es folgt also für beide oben diskutierte Fälle:

$$Pr[\langle y,w\rangle \equiv 0 \mod 2 \mid u \in \Lambda_w] = \frac{1}{2}.$$

Damit folgt nach Definition von Λ_w:

$$Pr[v \in \Lambda_w \mid u \in \Lambda_w] = \frac{1}{2}.$$

Da v ein ungerader Vektor ist, gilt nach Lemma 3.2.1

$$Pr[v \in \Lambda_w] = \frac{1}{2}.$$

Es folgt

$$Pr[v \in \Lambda_w] = Pr[v \in \Lambda_w \mid u \in \Lambda_w],$$

die Ereignisse $[u \in \Lambda_w]$ und $[v \in \Lambda_w]$ sind also unabhängig. □

Zum Abschluss dieses Abschnittes wird ein Werkzeug vorgestellt, mit dem sich Eindeutigkeitsaussagen beweisen lassen. Dieses Werkzeug wird eine Schlüsselrolle beim Korrektheitsbeweis der Reduktionen in den Kapiteln 4 und 5 spielen.

Satz 3.2.3. *Sei $T \neq \emptyset$ eine Menge mit höchstens 2^{cn} Elementen für ein $c \in \mathbb{N}$, und sei*

$$T = T_0 \supseteq T_1 \supseteq \cdots \supseteq T_{(c+1)n}$$

ein Teilmengensystem von T, welches über einen stochastischen Prozess definiert wird, der folgenden Bedingungen genügt:

(a) Für jedes $k \in \mathbb{N}$ mit $0 < k \leq (c+1)n$ und jedes $v \in T$ gilt:

$$Pr[v \in T_k \mid v \in T_{k-1}] = \frac{1}{2}.$$

(b) Für jedes $k \in \mathbb{N}$ mit $0 < k \leq (c+1)n$ und alle $u, v \in T_{k-1}$, $u \neq v$ gilt:
Die Ereignisse $[u \in T_k]$ und $[v \in T_k]$ sind unabhängig.

(c) Für jedes $v \in T$ und alle k, l mit $0 < k < l \leq (c+1)n$ sind

$$[v \in T_k \mid v \in T_{k-1}] \text{ und } [v \in T_l \mid v \in T_{l-1}]$$

unabhängige Ereignisse.

Dann enthält eine der Mengen T_k mit einer Wahrscheinlichkeit von mindestens $\frac{2}{3} - 2^{-n}$ genau ein Element.

Beweis. Zunächst wird bewiesen, dass mit Wahrscheinlichkeit von mindestens $1-2^{-n}$ die Menge $T_{(c+1)n}$ leer ist. Da es sich bei den T_k um ein Teilmengensystem handelt, gilt für jedes $v \in T$:

$$v \in T_{(c+1)n} \iff \bigwedge_{k=1}^{(c+1)n} v \in T_k.$$

Es folgt

$$Pr\left[v \in T_{(c+1)n}\right] = Pr\left[\bigwedge_{k=1}^{(c+1)n} v \in T_k\right].$$

Aufgrund von Eigenschaft (c) sind die Ereignisse $[v \in T_k \mid v \in T_{k-1}]$ für jedes k unabhängig und es folgt

$$Pr\left[v \in T_{(c+1)n}\right] = Pr\left[\bigwedge_{k=1}^{(c+1)n} v \in T_k\right] = \prod_{k=1}^{(c+1)n} Pr[v \in T_k \mid v \in T_{k-1}] \overset{\text{(a)}}{=} \left(\frac{1}{2}\right)^{(c+1)n} = 2^{-(c+1)n}.$$

Die Wahrscheinlichkeit dafür, dass $T_{(c+1)n}$ mindestens ein Element enthält, ist höchstens die Summe der Wahrscheinlichkeiten für jedes $v \in T$, dass $v \in T_{(c+1)n}$ gilt. Da T höchstens 2^{cn} viele Elemente enthält, folgt:

$$Pr\left[T_{(c+1)n} \neq \emptyset\right] \leq 2^{cn} \cdot 2^{-(c+1)n} = 2^{-n}.$$

Man erhält also:

$$Pr\left[T_{(c+1)n} = \emptyset\right] \geq 1 - 2^{-n}.$$

Ist k beliebig mit $0 \leq k < (c+1)n$ gewählt, so gilt mit (a) für zwei Vektoren $u, v \in T_k$ gerade $Pr[u \in T_{k+1}] = Pr[v \in T_{k+1}] = \frac{1}{2}$. Außerdem sind die Ereignisse $[u \in T_{k+1}]$ und $[v \in T_{k+1}]$ nach (b) unabhängig. Deshalb tritt jedes der Ereignisse

$$[u \in T_{k+1}, v \notin T_{k+1}],\ [u \notin T_{k+1}, v \in T_{k+1}],\ [u, v \notin T_{k+1}] \text{ und } [u, v \in T_{k+1}]$$

mit gleicher Wahrscheinlichkeit $\frac{1}{4}$ auf.

Man betrachte von nun an den Fall, dass $T_{(c+1)n} = \emptyset$ gilt und setzt

$$k := \max\{l : |T_l| \geq 2\}.$$

Dann gilt $|T_{k+1}| \leq 1$. Gesucht ist die Wahrscheinlichkeit dafür, dass $|T_{k+1}| = 1$ gilt.

Seien $u, v \in T_k$ mit $u \neq v$. Es gilt $|T_{k+1}| \leq 1$. Also kann das Ereignis $[u, v \in T_{k+1}]$ nicht eintreten. Unter dieser Bedingung treten die restlichen drei Ereignisse

$$[u \in T_{k+1}, v \notin T_{k+1}],\ [u \notin T_{k+1}, v \in T_{k+1}] \text{ und } [u, v \notin T_{k+1}]$$

mit Wahrscheinlichkeit von jeweils $\frac{1}{3}$ auf. Es gilt also insbesondere

$$Pr\left[u,v \notin T_{k+1} \mid T_{(c+1)n} = \emptyset\right] = \frac{1}{3}.$$

Das Gegenereignis beschreibt gerade den Fall, dass $|T_{k+1}| = 1$ gilt:

$$Pr\left[|T_{k+1}| = 1 \mid T_{(c+1)n} = \emptyset\right] = 1 - Pr\left[u,v \notin T_{k+1} \mid T_{(c+1)n} = \emptyset\right] = \frac{2}{3}.$$

Wenn $T_{(c+1)n} = \emptyset$ und $|T_{k+1}| = 1$ gilt, so gilt insbesondere, dass eine Menge des Teilmengensystems genau ein Element enthält. Man erhält also mit

$$Pr\left[T_{(c+1)n} = \emptyset\right] \cdot Pr\left[|T_{k+1}| = 1 \mid T_{(c+1)n} = \emptyset\right] \geq (1 - 2^{-n}) \cdot \frac{2}{3} \geq \frac{2}{3} - 2^{-n}$$

eine untere Schranke an die Wahrscheinlichkeit dafür, dass eine Menge des Systems genau ein Element enthält. Damit ist die Behauptung bewiesen. □

Wenn man als den zugrundeliegenden stochastischen Prozess für diesen Satz den Konstruktionsprozess des LATTICEGENERATOR Algorithmus wählt, so stellt man fest, dass die Bedingungen (a) und (b) den Aussagen der Lemmata 3.2.1 und 3.2.2 in etwa entsprechen. Es liegt also nahe, diese Lemmata für die Korrektheitsbeweise der Reduktionen zu verwenden.

Mit dem in diesem Kapitel vorgestellten Konstruktionsalgorithmus LATTICEGENERATOR und den in diesem Abschnitt bewiesenen Aussagen ist die Grundlage für die Reduktionen in den Kapiteln 4 und 5 und deren Korrektheitsbeweise geschaffen.

4 Untersuchung der Schwierigkeit von UNIQUE-SVP

Sowohl SVP als auch UNIQUE-SVP befassen sich mit von 0 verschiedenen Gittervektoren v aus Gittern Λ mit minimaler Länge. Für sie gilt $\|v\|_p = \lambda_1(\Lambda)$. Diese Vektoren werden im Folgenden nur noch als kürzeste Vektoren bezeichnet.

In diesem Kapitel soll die Berechnungkomplexität der beiden Probleme SVP und UNIQUE-SVP in Beziehung gesetzt werden. UNIQUE-SVP ist offensichtlich nicht schwieriger als SVP, da es ein Spezialfall von letzterem Problem darstellt.

Kumar und Sivakumar haben 1999 in [KS99] für die Entscheidungsversion von UNIQUE-SVP gezeigt, dass sie für die l_2-Norm NP-hart ist unter randomisierten Reduktionen. Teil ihres Beweises war eine randomisierte Reduktion einer SVP Probleminstanz mit gewissen Eigenschaften auf UNIQUE-SVP.

Die dort verwendete Beweistechnik wird in diesem Kapitel nun aufgegriffen. Es wird gezeigt, dass eine polynomiell zeitbeschränkte, randomisierte Reduktion von der Suchvariante von SVP auf diese Variante von UNIQUE-SVP existiert, für alle l_p-Normen mit $p \in (1,\infty)$.

Diese randomisierte Reduktion wird nur mit einer Wahrscheinlichkeit von $\Omega(n^{-1})$ in Abhängigkeit des Ranges n des Eingabegitters erfolgreich sein. In Abschnitt 4.3 wird aber unter Verwendung dieser Reduktion ein SVP-Löser konstruiert, der in seiner Laufzeit und Anzahl an Aufrufen eines UNIQUE-SVP-Orakels polynomiell beschränkt ist in Rang und Dimension des Eingabegitters, jedoch eine Erfolgswahrscheinlichkeit hat, die exponentiell nahe an 1 liegt.

Das Ergebnis dieses Kapitels wird also sein, dass die Probleme SVP und UNIQUE-SVP gleichschwierig zu lösen sind sind für alle l_p-Normen mit $p \in (1,\infty)$. Man kann jedes der beiden Probleme mit höchstens polynomiellem Mehraufwand mit Hilfe des jeweils anderen Problems lösen.

4.1 Reduktion von SVP auf UNIQUE-SVP

Zu konstruieren ist eine Reduktion, die aus einer Probleminstanz für SVP eine gültige Probleminstanz für UNIQUE-SVP berechnet. Dabei muss die Lösung für die errechnete UNIQUE-SVP Probleminstanz auch eine korrekte Lösung für die SVP Probleminstanz sein.

Eine SVP Probleminstanz ist ein rationales Gitter L, und eine korrekte Lösung ist ein kürzester Vektor von L. Eine gültige Probleminstanz von UNIQUE-SVP hingegen ist ein rationales Gitter $\hat{L}$, in welchem alle kürzesten Vektoren eindeutig bezüglich der in (6) definierten Äquivalenzrelation ρ der linear abhängigen Vektoren sind. Eine Lösung ist ein kürzester Vektor von $\hat{L}$.

Das bedeutet, dass die Reduktion aus einem rationalen Gitter L ein Gitter $\hat{L}$ zu berechnen hat, dessen kürzeste Vektoren auch kürzeste Vektoren von L sind. Außerdem müssen alle kürzesten Vektoren von $\hat{L}$ eindeutig bezüglich ρ sein.

Folgende Reduktion h wird dies leisten. Sie erhält als Eingabe ein rationales Gitter L vom Rang n und Dimension m, spezifiziert durch eine geeignet kodierte Basis B. Sie verwendet den Algorithmus LATTICEGENERATOR, um Basen für ein System von $2n+1$ Untergittern

$$L_0, \ldots, L_{2n} \tag{13}$$

zu erzeugen, und wählt anschließend eines dieser Gitter zufällig gleichverteilt aus. Dieses Gitter $\hat{L}$ bildet die Ausgabe der Reduktion, in Form einer geeignet kodierten Basis. Es folgt die formale Beschreibung der Reduktion:

Algorithmus 2 Reduktion h

function $h(B)$
  $B_0, \ldots, B_{2n} \leftarrow$LATTICEGENERATOR$(B, 2n)$
  Wähle $i \in \{0, \ldots, 2n\}$ zufällig unabhängig gleichverteilt
  return B_i
end function

Nach Satz 3.1.2 ist der Algorithmus LATTICEGENERATOR polynomiell zeitbeschränkt in Dimension m, Rang n und Anzahl der zu erzeugenden Gitter. Letztere ist in diesem Fall $2n+1$, also selbst polynomiell zeitbeschränkt in n. Damit erfolgt die Erzeugung der Gitter (13) mit einem Zeitaufwand, der polynomiell in m und n ist. Das Wählen eines Gitters und Kodieren der Ausgabe ist in Zeit $O(mn)$ möglich. Die Reduktion h ist also polynomiell zeitbeschränkt in m und n.

Es bleibt noch zu zeigen, dass die Reduktion mit genügend hoher Wahrscheinlichkeit Erfolg hat. Wie oben erläutert, hat h Erfolg, wenn die kürzesten Vektoren des Ausgabegitters $\hat{L}$ auch kürzeste Vektoren vom Eingabegitter L sind, und sie eindeutig bezüglich ρ sind. Da $\hat{L}$ immer ein Untergitter von L ist, sind die kürzesten Vektoren des Ausgabegitters genau dann auch kürzeste Vektoren des Eingabegitters, wenn $\lambda_1(L) = \lambda_1(\hat{L})$ gilt.

Deshalb genügt es, folgende Aussage zu beweisen:

Theorem 4.1.1. *Sei L das Eingabegitter von h mit Rang n. Die zugrundeliegende Norm sei die l_p-Norm mit $p \in (1, \infty)$.*

Mit einer Wahrscheinlichkeit von $\Omega(n^{-1})$ wird h ein Gitter $\hat{L}$ ausgeben, das folgende zwei Bedingungen erfüllt:

- $\lambda_1(L) = \lambda_1(\hat{L})$,
- *Die Menge* $\left\{ v \in \hat{L} \,:\, \|v\|_p = \lambda_1(L) \right\}$ *ist eindeutig bezüglich* ρ

Der restliche Abschnitt wird sich mit dem Beweis dieses Theorems beschäftigen.

In Abschnitt 3.2 wurden Wahrscheinlichkeitsaussagen über die einzelnen Konstruktionsschitte des Algorithmus LATTICEGENERATOR bewiesen. Außerdem wurde ein Werkzeug vorgestellt, um gewisse Eindeutigkeitsaussagen treffen zu können. Um dies für den Beweis des Theorems einsetzen zu können, müssen zunächst die Eigenschaften der kürzesten Vektoren analysiert werden.

Man stellt fest, dass alle kürzesten Vektoren primitiv sind:

Lemma 4.1.2. *Sei Λ ein Gitter. Dann gilt für jeden Gittervektor $v \in \Lambda$ mit $\|v\|_p = \lambda_1(\Lambda)$, dass v primitiv ist.*

Beweis. Sei $v \in \Lambda$ ein Gittervektor, der $\|v\|_p = \lambda_1(\Lambda)$ erfüllt.

Angenommen, der Vektor v ist nicht primitiv. Dann ist v ein ganzzahliges Vielfaches eines anderen Gittervektors. Es gibt dann eine Zahl $n \in \mathbb{N}$, $n \geq 2$, so dass

$$u := \frac{1}{n} v \in \Lambda$$

gilt. Damit gilt

$$0 < \|u\|_p = \left\| \frac{1}{n} v \right\|_p < \|v\|_p = \lambda_1(\Lambda),$$

und man erhält einen Widerspruch. Denn die Länge eines von 0 verschiedenen Gittervektors ist nun echt kleiner als $\lambda_1(\Lambda)$. □

In Abschnitt 2.2 wurde gezeigt, dass alle primitiven Vektoren auch ungerade sind. Damit sind die kürzesten Vektoren also ebenfalls ungerade.

Es folgt eine Aussage über die Differenzvektoren kürzester Gittervektoren:

Lemma 4.1.3. *Sei Λ ein Gitter der Dimension m, und sei die zugrundeliegende Norm eine l_p-Norm mit $p \in (1, \infty)$.*

Dann gilt für jedes Paar von Gittervektoren $u, v \in \Lambda$ mit $\|u\|_p = \|v\|_p = \lambda_1(\Lambda)$ und $u \notin [v]_\rho$, dass $v - u$ ein ungerader Gittervektor ist.

Beweis. Auch dieser Beweis wird als Widerspruchsbeweis geführt. Seien $u, v \in \Lambda$ mit

$$\|u\|_p = \|v\|_p = \lambda_1(\Lambda),$$

und gelte $u \notin [v]_\rho$.

Angenommen, der Vektor $v - u$ sei gerade. Die alternative Charakterisierung gerader Vektoren durch (5) liefert $\frac{1}{2}(v-u) \in \Lambda$. Damit gilt aber auch $\frac{1}{2}(v+u) = \frac{1}{2}(v-u) + u \in \Lambda$.

Die Dreiecksungleichung für Normen sagt aus:

$$\|u\|_p + \|v\|_p \geq \|u+v\|_p . \tag{14}$$

Einsetzen von $\lambda_1(\Lambda)$ für $\|u\|_p$ und $\|v\|_p$, sowie umformen liefert:

$$\lambda_1(\Lambda) \geq \left\| \frac{1}{2}(u+v) \right\|_p . \tag{15}$$

Da $u \notin [v]_\rho$ vorausgesetzt wurde, gilt insbesondere $u+v \neq 0$. Es folgt, dass die Abschätzung (15) mit Gleichheit erfüllt wird, da kein von 0 verschiedener Gittervektor kürzer ist als $\lambda_1(\Lambda)$. Man erhält also:

$$\lambda_1(\Lambda) = \left\| \frac{1}{2}(u+v) \right\|_p = \frac{1}{2} \|u+v\|_p .$$

Die Dreiecksungleichung (14) ist also mit Gleichheit erfüllt. Nun sind die l_p-Normen für $p \in (1,\infty)$ streng konvex. Das bedeutet, dass gilt:

$$\forall x,y \in \mathbb{R}^m : \ \Big(\|x\|_p = \|y\|_p \text{ und } \|x\|_p + \|y\|_p = \|x+y\|_p \Big) \implies x = y.$$

Ein Beweis dieser Aussage ist im Anhang A.1 nachzulesen.

Da in diesem Fall gerade $\|u\|_p = \|v\|_p$ gilt, und (14) mit Gleichheit erfüllt ist, folgt also $u = v$ im Widerspruch zur Voraussetzung $u \notin [v]_\rho$. □

Aus dieser Aussage lässt sich eine obere Schranke für die Anzahl an Äquivalenzklassen mit kürzesten Gittervektoren folgern:

Lemma 4.1.4. *Sei Λ ein Gitter vom Rang n, und sei die zugrundeliegende Norm eine l_p-Norm mit $p \in (1,\infty)$. Dann gibt es höchstens 2^n Äquivalenzklassen von Gittervektoren aus Λ bezüglich ρ, die einen kürzesten Vektor enthalten.*

Beweis. Sei B eine Basis von Λ, und v ein Gittervektor mit Koeffizientenvektor x bezüglich B. Der Paritätsvektor $p(v) = (p_1, p_2, \ldots, p_n)^T \in \mathbb{Z}_2^n$ eines Gittervektors v bezüglich B ist komponentenweise definiert wie folgt:

$$p_i := \begin{cases} 0, \text{ falls } x_i \equiv 0 \mod 2 \\ 1, \text{ sonst.} \end{cases}$$

Der Paritätsvektor gibt also gerade an, welche Komponenten des Koeffizientenvektors von v gerade Zahlen, und welche ungerade Zahlen sind.

Seien nun $u = Bx$ und $v = By$ zwei kürzeste Vektoren aus zwei verschiedenen Äquivalenzklassen.

Angenommen, u und v hätten den gleichen Paritätsvektor. Dann wäre jede Komponente x_i eine gerade Zahl genau dann, wenn y_i dies wäre. Das bedeutet aber, dass der Vektor $y-x$ nur gerade Komponenten enthält. $y-x$ ist der Koeffizientenvektor von $v-u$. Damit ist $v-u$ ein gerader Gittervektor im Widerspruch zur Aussage von Lemma 4.1.3, nach der dieser Gittervektor ungerade ist.

Je zwei kürzeste Vektoren aus verschiedenen Äquivalenzklassen haben also nie den gleichen Paritätsvektor. Damit kann es höchstens so viele Äquivalenzklassen geben, die einen kürzesten Vektor enthalten, wie es verschiedene Paritätsvektoren gibt. Die Anzahl an verschiedenen Paritätsvektoren wiederum ist gegeben durch die Anzahl an Vektoren in $\mathbb{Z}_2^n$, nämlich 2^n. □

Die drei Lemmata werden verwendet, um folgenden Satz zu beweisen:

Satz 4.1.5. *Sei $p \in (1,\infty)$, und $T_0,\ldots,T_{2n}$ ein Mengensystem, definiert über den Konstruktionsprozess der Gitter* (13) *durch*

$$T_k := \left\{ [v]_\rho \;:\; v \in L_k,\ \|v\|_p = \lambda_1(L) \right\} \tag{16}$$

mit L_k aus (13)*, und $[v]_\rho$ gebildet über L.*

Mit einer Wahrscheinlichkeit von mindestens $\frac{2}{3} - 2^{-n}$ werden die Gitter L_k so erzeugt, dass mindestens eine Menge des Systems (16) *genau ein Element enthält.*

Beweis. Die Behauptung kann man unter Verwendung des Werkzeuges aus Abschnitt 3.2, dem Satz 3.2.3, beweisen. Es ist lediglich zu überprüfen, ob das Mengensystem (16) alle Voraussetzungen erfüllt. Es besteht aus $2n$ Mengen, demnach wählt man $c = 1$. Satz 3.2.3 stellt folgende Bedingungen:

(a) $0 < |T_0| \leq 2^n$

(b) $T_0 \supseteq T_1 \supseteq \cdots \supseteq T_{2n}$

(c) $\forall k \in \mathbb{N}, 0 < k \leq 2n\ \forall v \in T_0$: $Pr[v \in T_k \mid v \in T_{k-1}] = \frac{1}{2}$

(d) $\forall k \in \mathbb{N}, 0 < k \leq 2n\ \forall u, v \in T_{k-1}, u \neq v$:
Die Ereignisse $[u \in T_k]$ und $[v \in T_k]$ sind unabhängig.

(e) $\forall v \in T_0\ \forall k, l \in \mathbb{N}, 0 < k < l \leq 2n$:
Die Ereignisse $[v \in T_k \mid v \in T_{k-1}]$ und $[v \in T_l \mid v \in T_{l-1}]$ sind unabhängig.

Jedes Gitter enthält einen kürzesten Vektor. Dieser ist in einer Äquivalenzklasse enthalten und es folgt $|T_0| > 0$. Nach Lemma 4.1.4 hat jedes Gitter höchstens 2^n Äquivalenzklassen, die kürzeste Vektoren enthalten. Es folgt $|T_0| \leq 2^n$. Bedingung (a) ist also erfüllt.

Da das Untergittersystem (13) mit dem Algorithmus LATTICEGENERATOR erzeugt wird, erfüllt es

$$L_0 \supseteq L_1 \supseteq \cdots \supseteq L_{2n}.$$

Nun gilt für alle $k \in \mathbb{N}$ mit $0 < k \leq 2n$: Wenn eine Äquivalenzklasse $[\hat{v}]_\rho$ in T_k enthalten ist, so gibt es einen kürzesten Gittervektor v in L mit $v \in [\hat{v}]_\rho$ und $v \in L_k$. Da $L_{k-1} \supseteq L_k$ gilt, ist er demnach auch in L_{k-1} enthalten und es folgt $[\hat{v}]_\rho \in T_{k-1}$. Damit gelten folgende Inklusionsbeziehungen:

$$T_0 \supseteq T_1 \supseteq \cdots \supseteq T_{2n}$$

und es folgt die Gültigkeit von (b).

Sei im Folgenden $k \in \mathbb{N}$ beliebig gewählt mit $0 < k \leq 2n$.

Es wird nun die Wahrscheinlichkeit dafür untersucht, dass eine Äquivalenzklasse aus T_{k-1} nach der Erzeugung von L_k auch in T_k enthalten ist. Sei $[\hat{v}]_\rho$ eine Äquivalenzklasse aus T_{k-1}, und sei $v \in [\hat{v}]_\rho$, $v \in L_{k-1}$ ein kürzester Vektor von L. In Abschnitt 2.3 wurde bereits festgestellt, dass dann v und $-v$ die einzigen kürzesten Vektoren in $[\hat{v}]_\rho$ sind. Es gilt $[\hat{v}]_\rho \in T_k$ also genau dann, wenn $v \in L_k$ oder $-v \in L_k$ gilt. Da L_k ein Gitter ist, gilt mit $v \in L_k$ auch automatisch $-v \in L_k$ und umgekehrt. Das Ereignis $\left[[\hat{v}]_\rho \in T_k\right]$ tritt also genau dann ein, wenn das Ereignis $[v \in L_k]$ eintritt. v ist als kürzester Vektor von L auch ein kürzester Vektor von L_{k-1}. Nach Lemma 4.1.2 ist er also primitiv, und damit ungerade in L_{k-1}. Für ungerade Vektoren aus L_{k-1} gilt nach Lemma 3.2.1 mit Wahrscheinlichkeit $\frac{1}{2}$, dass sie auch in L_k enthalten sind. Es folgt

$$Pr\left[[\hat{v}]_\rho \in T_k \mid [\hat{v}]_\rho \in T_{k-1}\right] = \frac{1}{2},$$

was zeigt, dass die Bedingung (c) erfüllt ist.

Seien $[\hat{u}]_\rho, [\hat{v}]_\rho \in T_{k-1}$ zwei Äquivalenzklassen, und seien $u \in [\hat{u}]_\rho$ und $v \in [\hat{v}]_\rho$ Vektoren aus L_{k-1} mit $\|u\|_p = \|v\|_p = \lambda_1(L)$. Wie soeben gesehen, treten die Ereignisse $\left[[\hat{u}]_\rho \in T_k\right]$ bzw. $\left[[\hat{v}]_\rho \in T_k\right]$ genau dann ein, wenn die Ereignisse $[u \in L_k]$ bzw. $[v \in L_k]$ eintreten. Da nach Lemma 4.1.3 der Vektor $v - u$ ein ungerader Gittervektor ist, lässt sich Lemma 3.2.2 anwenden und liefert, dass die Ereignisse $[u \in L_k]$ und $[v \in L_k]$ unabhängig sind.

Also sind auch die Ereignisse $\left[[\hat{u}]_\rho \in T_k\right]$ und $\left[[\hat{v}]_\rho \in T_k\right]$ unabhängig, Eigenschaft (d) ist also erfüllt.

Sei wieder $[\hat{v}]_\rho \in T_0$ eine Äquivalenzklasse, und $v \in [\hat{v}]_\rho$ ein kürzester Vektor. Dann sind für $k \neq l$ auch die Ereignisse $[v \in L_k \mid v \in L_{k-1}]$ und $[v \in L_l \mid v \in L_{l-1}]$ unabhängig, da in jedem Schritt des Algorithmus LATTICEGENERATOR die Menge W unabhängig von allen vorherigen Wahlen neu gewählt wird. Es folgt, dass $\left[[\hat{v}]_\rho \in T_k \mid [\hat{v}]_\rho \in T_{k-1}\right]$ und $\left[[\hat{v}]_\rho \in T_l \mid [\hat{v}]_\rho \in T_{l-1}\right]$ unabhängig sind für $k \neq l$. Damit ist auch die Eigenschaft (e) gezeigt.

Das Mengensystem (16) erfüllt also alle Voraussetzungen von Satz 3.2.3 mit $c = 1$. Anwenden des Satzes liefert die Behauptung. □

Es folgt der Beweis des Theorems:

Beweis. [von Theorem 4.1.1]

Man betrachte noch einmal die Definition des Mengensystems (16). Wenn ein T_k genau ein Element enthält, so bedeutet dies zum einen, dass das Gitter L_k kürzeste Vektoren von L enthält. Es gilt also $\lambda_1(L) = \lambda_1(L_k)$. Zum anderen befinden sich alle kürzesten Vektoren in einer Äquivalenzklasse, sie sind dann also eindeutig bezüglich ρ. Wenn dieses Gitter L_k also als Ausgabe der Reduktion gewählt wird, so erfüllt die Ausgabe die geforderten Eigenschaften.

Nach Satz 4.1.5 gilt, dass mit einer Wahrscheinlichkeit von mindestens $\frac{2}{3} - 2^{-n}$ eine Menge T_k genau ein Element enthält. Der Algorithmus LATTICEGENERATOR erzeugt also mit Wahrscheinlichkeit von mindestens $\frac{2}{3} - 2^{-n}$ ein Gitter, das die geforderten Eigenschaften erfüllt.

Die Reduktion wählt zufällig gleichverteilt eines der $2n+1$ Gitter des Systems (13). Die Wahrscheinlichkeit dafür, dass ein bestimmtes Gitter L_k gewählt wird, ist also $\frac{1}{2n+1}$.

Da die Erzeugung des Gittersystems und die Wahl des Gittersystems unabhängig sind, ist die Wahrscheinlichkeit dafür, dass ein Gitter mit den erforderlichen Eigenschaften erzeugt und ausgewählt wird also mindestens

$$\left(\frac{2}{3} - 2^{-n}\right) \cdot \frac{1}{2n+1} = \Omega(n^{-1}).$$

□

Für die l_p-Normen mit $p \in (1,\infty)$ funktioniert die Reduktion h also wie gewünscht. Damit ist gezeigt, dass SVP bezüglich dieser Normen nicht schwieriger ist als UNIQUE-SVP (unter randomisierten Reduktionen). Für die l_1-Norm und die l_∞-Norm lässt sich diese Beweistechnik so nicht anwenden. Dieser Sachverhalt wird im nächsten Abschnitt genauer untersucht.

4.2 Untersuchung der Normen l_1 und l_∞

Die l_p-Norm ist für $p = 1$ und $p = \infty$ nicht streng konvex. Deshalb lässt sich der Beweis von Lemma 4.1.3, also dass die Differenz von zwei kürzesten Vektoren aus verschiedenen Äquivalenzklassen ein ungerader Gittervektor ist, für diese Normen so nicht führen. Tatsächlich ist diese Aussage im Allgemeinen nicht korrekt. Dies sieht man leicht an folgenden Beispielen:

Beispiel 1: Gegenbeispiel für die l_∞-Norm

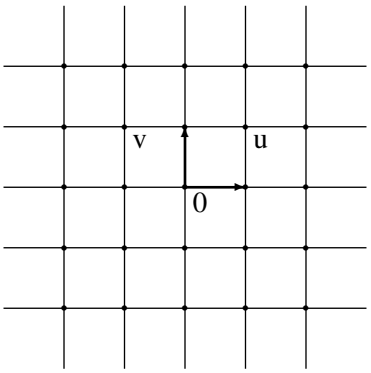

Abbildung 2: Gegenbeispiel für die l_∞-Norm

Man betrachte das Gitter $\mathbb{Z}^2 = \mathcal{L}\left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right)$ (siehe Abbildung 2). Es sind sowohl $u = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, als auch $v = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$ kürzeste Vektoren im Gitter $\mathbb{Z}^2$. Sie sind linear unabhängig und stammen damit aus verschiedenen Äquivalenzklassen. Jedoch ist die Differenz

$$v - u = \begin{pmatrix} -2 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} -2 \\ 0 \end{pmatrix}$$

ein gerader Gittervektor.

Beispiel 2: Gegenbeispiel für die l_1-Norm

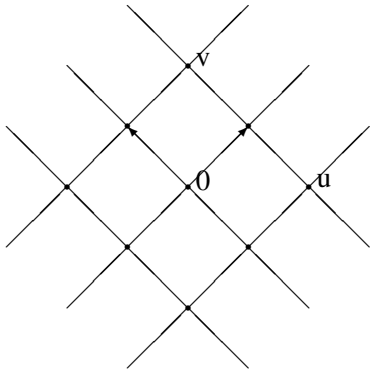

Abbildung 3: Gegenbeispiel für die l_1-Norm

Sei

$$B = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$$

eine Gitterbasis von $\Lambda := \mathcal{L}(B)$ (siehe Abbildung 3).

Die Vektoren $u = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$ und $v = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$ sind kürzeste Vektoren von Λ. Sie sind linear unabhängig und stammen damit aus verschiedenen Äquivalenzklassen. Jedoch ist die Differenz

$$v - u = \begin{pmatrix} -2 \\ 2 \end{pmatrix} = B \cdot \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

ein gerader Gittervektor in Λ.

Ohne die Gültigkeit von Lemma 4.1.3 wird ein Nachweis für die Korrektheit von h für diese Normen erheblich erschwert. Unter anderem verliert auch das Lemma 4.1.4 seine Gültigkeit. Damit ist die Anzahl an kürzesten Vektoren nicht mehr durch 2^n beschränkt. Dieses Problem

lässt sich jedoch Lösen, indem man mit Hilfe von Lemma 2.2.4 eine nicht viel gröbere Abschätzung für die maximale Anzahl an kürzesten Vektoren herleitet.

Ein viel gravierenderes Problem ist jedoch folgendes: Da der Differenzvektor zweier kürzester Vektoren nicht mehr unbedingt ungerade ist, gilt auch nicht mehr für alle Paare von kürzesten Vektoren u und v aus verschiedenen Äquivalenzklassen, dass die Ereignisse $[u \in L_{k+1} \mid u \in L_k]$ und $[v \in L_{k+1} \mid v \in L_k]$ unabhängig sind. Diese Ereignisse sind dann im Gegenteil sogar identisch. Deshalb müssen diese Vektoren im Konstruktionsprozess allesamt entfernt werden, damit die Reduktion erfolgreich ist. Man müsste also untersuchen, wie groß der Anteil dieser Vektoren an der Gesamtzahl der kürzesten Gittervektoren maximal sein kann, um schließlich Aussagen über die Wahrscheinlichkeit dafür treffen zu können, dass alle diese Vektoren entfernt werden.

Bei derartigen Untersuchungen kann sich durchaus herausstellen, dass die Reduktion auch für die l_1- und l_∞-Normen erfolgreich ist. Eine weitere Untersuchung dieses Sachverhaltes würde den Rahmen dieser Arbeit jedoch übersteigen.

4.3 Erhöhung der Erfolgswahrscheinlichkeit durch Wiederholung

In diesem Abschnitt wird unter Verwendung der Reduktion h und eines UNIQUE-SVP-Orakels ein Löser für SVP konstruiert, welcher polynomiell zeitbeschränkt ist in Dimension und Rang des Eingabegitters. Seine Erfolgswahrscheinlichkeit wird dabei exponentiell nahe an 1 liegen.

Das UNIQUE-SVP-Orakel beantwortet eine Anfrage immer in einem Zeitschritt, und seine Antwort ist immer korrekt, falls die Anfrage eine gültige UNIQUE-SVP Probleminstanz ist. Ist hingegen die Anfrage keine gültige Probleminstanz, so darf das Orakel falsche Antworten liefern.

Die Idee für den Löser ist Folgende: Man führt die Reduktion h mehrfach durch, und verwendet deren Ergebnisse als Anfragen für das UNIQUE-SVP-Orakel. Falls eine der Reduktionen erfolgreich ist, wird das UNIQUE-SVP-Orakel eine korrekte Lösung für SVP liefern. Für eine erfolglose Reduktion hingegen liefert das UNIQUE-SVP-Orakel eine undefinierte Antwort. Man kann jedoch in polynomieller Zeit prüfen, ob diese Antwort einen Vektor kodiert, und ob es sich um einen Gittervektor handelt. Letzteres beschränkt sich nämlich im Wesentlichen auf das Lösen des Gleichungssystems $Bx = v$, wobei B eine Basis des Gitters, und v der Ausgabevektor des Orakels ist. Dies ist zum Beispiel mit dem Gaußschen Eliminationsverfahren in kubischer Zeit möglich.

Der Löser wählt den kürzesten, von Null verschiedenen Gittervektor aus allen Antworten der Orakelanfragen, und gibt ihn aus. Dieser Vektor ist mindestens dann eine korrekte Lösung für SVP, wenn eine der Reduktionen erfolgreich war.

Es folgt eine formale Beschreibung des Lösers (siehe Algorithmus 3). Als Eingabe erhält er eine geeignet kodierte Basis B eines rationalen Gitters, für welches SVP gelöst werden soll:

v_∞ bezeichnet ein Symbol für einen „längsten Vektor“. Dieser Algorithmus benötigt polynomielle Zeit in Abhängigkeit von d und der Spaltenanzahl n und Zeilenanzahl m von B, da die

Algorithmus 3 SVP-Löser

function SVPSOLVER(B, d)
  $v_{\text{best}} \leftarrow v_\infty$
  for i= 1, ..., d **do**
    $B' \leftarrow h(B)$
    $v \leftarrow$ UNIQUE-SVP(B')
    if $v \in \mathcal{L}(B)\backslash\{0\}$ und $\|v\|_p < \|v_{\text{best}}\|_p$ **then**
      $v_{\text{best}} \leftarrow v$
    end if
  end for
  if $v_{\text{best}} = v_\infty$ **then**
    return FAIL
  end if
  return v_{best}
end function

Reduktion h polynomiell zeitbeschränkt in m und n ist und d mal aufgerufen wird. Es erfolgen d Anfragen an das UNIQUE-SVP-Orakel.

Es folgt eine Untersuchung der Erfolgswahrscheinlichkeit für den SVP-Löser in Abhängigkeit von d. Dazu bezeichnen $X_1, X_2, \ldots, X_d$ Zufallsvariablen mit

$$X_i = \begin{cases} 1, \text{ falls der } i\text{-te Aufruf von } h \text{ erfolgreich ist} \\ 0, \text{ sonst.} \end{cases}$$

Die Erfolgswahrscheinlichkeit von h ist mindestens $\Omega(n^{-1})$, also mindestens $\frac{c}{n}$ für eine geeignete Konstante $c > 0$. Es folgt für jedes $i \in \{1, 2, \ldots, d\}$:

$$Pr[X_i = 1] \geq \frac{c}{n}. \tag{17}$$

Der SVP-Löser ist, wie oben schon erwähnt, höchstens dann nicht erfolgreich, wenn alle Aufrufe von h fehlschlagen. Man kann davon ausgehen, dass die Misserfolgsereignisse für die einzelnen Aufrufe von h stochastisch unabhängig sind und erhält:

$$Pr\left[\text{Löser erfolgreich}\right] \geq 1 - \prod_{i=1}^{d} Pr\left[X_i = 0\right] \geq 1 - Pr\left[\sum_{i=1}^{d} X_i \leq 0\right]. \tag{18}$$

Für ein Sequenz von d unabhängigen Bernoulli-Experimenten mit $Pr\left[X_i = 1\right] = p$ für jedes $i \in \{1, \ldots, d\}$ liefert die zweite Chernoff-Schranke [Che52] folgende Abschätzung:

$$Pr\left[\sum_{i=1}^{d} X_i \leq (1-\delta) \cdot pd\right] \leq \exp\left(-\frac{\delta^2}{2} pd\right).$$

Man kann die Aufrufe der Reduktion h als Sequenz von unabhängigen Bernoulli-Experimenten auffassen. Nach (17) gilt $p = \frac{c}{n}$. Wählt man $\delta = 1$, so kann man die zweite Chernoff-Schranke auf (18) anwenden und erhält:

$$\begin{aligned} Pr[\text{Löser erfolgreich}] &\geq 1 - \exp\left(-\frac{1}{2}\frac{cd}{n}\right) \\ &= 1 - \exp\left(-\tilde{c}\frac{d}{n}\right) \end{aligned}$$

für eine geeignete Konstante $\tilde{c} > 0$. Ersetzt man d durch das Polynom n^2, so arbeitet der SVP-Löser in polynomieller Zeit in Abhängigkeit von m und n, und mit einer polynomiell beschränkten Anzahl an Aufrufen des UNIQUE-SVP-Orakels in Abhängigkeit von n.

Es folgt:

$$Pr[v \text{ ist korrekte Lösung}] \geq 1 - \exp(-\tilde{c}n).$$

Die Erfolgswahrscheinlichkeit des SVP-Lösers ist dann also exponentiell nahe an 1.

Das Ergebnis dieses Kapitels ist also, dass die Probleme SVP und UNIQUE-SVP gleichschwierig zu lösen sind für alle l_p-Normen mit $p \in (1, \infty)$. Denn man kann jedes der beiden Probleme mit höchstens polynomiellem Mehraufwand mit Hilfe des jeweils anderen Problems lösen.

5 Untersuchung der Schwierigkeit von UNIQUE-CVP

Wie schon in Abschnitt 2.3 beschrieben, befassen sich das Problem des nächsten Gittervektors und seine Varianten mit Gittervektoren aus Gittern Λ, welche zu einem Zielvektor t aus dem zugrundeliegenden Vektorraum minimalen Abstand haben. Es handelt sich also um Gittervektoren v, die

$$\|v-t\|_p = \min\left\{\|u-t\|_p : u \in \Lambda\right\}$$

erfüllen. Diese Vektoren werden im Folgenden einfach als nächste Vektoren zu t bezeichnet.

In diesem Kapitel wird UNIQUE-CVP aus komplexitätstheoretischer Sicht untersucht. Im Wesentlichen geht es darum, seine Schwierigkeit in Beziehung zu der von FEW-CVP$_c$ zu setzen. Da UNIQUE-CVP Probleminstanzen Spezialfälle von FEW-CVP$_c$ Probleminstanzen sind, ist UNIQUE-CVP offensichtlich nicht schwieriger als FEW-CVP$_c$. In Abschnitt 5.1 wird gezeigt, dass aber auch FEW-CVP$_c$ nicht schwieriger als UNIQUE-CVP ist für l_p-Normen mit $p \in (1,\infty)$, indem eine randomisierte Reduktionsfunktion angegeben wird. Die Reduktion wird derjenigen aus Kapitel 4 sehr ähneln, und man wird sehen, dass sich die dort verwendete Beweistechnik weitgehend übertragen lässt.

Auch diese randomisierte Reduktion wird eine Erfolgswahrscheinlichkeit von nur $\Omega(n^{-1})$ haben. Man kann, wie bereits für SVP gesehen, einen FEW-CVP$_c$-Löser konstruieren, welcher polynomiellen Zeitbedarf in Dimension und Rang des Eingabegitters hat, und mit einer polynomiellen Anzahl von Aufrufen eines UNIQUE-CVP-Orakels auskommt, dabei aber eine Erfolgswahrscheinlichkeit hat, die exponentiell nahe an 1 liegt. Da diese Konstruktion völlig analog zur Konstruktion des SVP-Lösers in Abschnitt 4.3 möglich ist, wird sie aber nicht explizit durchgeführt. In Abschnitt 5.3 wird man sehen, dass die Probleme FEW-CVP$_c$ und CVP für jedes $c \in \mathbb{N}$ und l_p-Normen mit $p \in (1,\infty)$ identisch sind.

Das Ergebnis dieses Kapitels wird also sein, dass die Suchvarianten der Probleme CVP, FEW-CVP$_c$ und UNIQUE-CVP gleichschwierig zu lösen sind für alle l_p-Normen mit $p \in (1,\infty)$. Man kann jedes der drei Probleme mit höchstens polynomiellem Mehraufwand mit Hilfe der anderen Probleme lösen.

5.1 Reduktion von FEW-CVP$_c$ auf UNIQUE-CVP

Zu konstruieren ist eine Reduktion, die FEW-CVP$_c$ Probleminstanzen auf UNIQUE-CVP Probleminstanzen abbildet, und zwar derart, dass eine Lösung der generierten UNIQUE-CVP$_c$ Probleminstanz auch eine korrekte Lösung der FEW-CVP$_c$ Probleminstanz ist.

Eine FEW-CVP$_c$ Probleminstanz besteht aus einem rationalen Gitter L der Dimension n und einem Zielvektor t, wobei das Gitter L höchstens 2^{cn} nächste Vektoren zu t enthält. Eine Lösung dieser Probleminstanz ist ein nächster Vektor aus L zu t. Eine UNIQUE-CVP Probleminstanz unterscheidet sich von einer FEWCVP$_c$ Probleminstanz nur durch die stärkere Einschränkung, dass das Gitter L genau einen nächsten Vektor zu t enthält.

Die Reduktion muss also aus einem rationalen Gitter L, welches höchstens 2^{cn} viele nächste Vektoren besitzt, ein Gitter $\hat{L}$ berechnen, welches genau einen nächsten Vektor zu t enthält. Dieser nächste Vektor aus $\hat{L}$ muss außerdem auch ein nächster Vektor aus L sein.

Folgende Reduktion h_c wird dies leisten: Als Eingabe erhält h_c ein Tupel (L,t), wobei L ein rationales Gitter vom Rang n und Dimension m, und $t \in \mathbb{Q}^m$ der Zielvektor ist. Das Gitter L wird dabei spezifiziert durch eine geeignet kodierte Basis B. Genau wie die Reduktion h aus Abschnitt 4.1 verwendet sie den Algorithmus LATTICEGENERATOR, um Basen für ein System von $(c+1)n+1$ Untergittern

$$L_0, \ldots, L_{(c+1)n} \tag{19}$$

zu erzeugen. Anders als h erzeugt sie anschließend zusätzlich ein Gitter L', welches nur aus den geraden Gittervektoren von L besteht. Eine Basis dieses Gitters lässt sich einfach durch Skalierung der Basis von L um den Faktor 2 berechnen. Anschließend wählt h_c zufällig gleichverteilt eines der Gitter $L', L_0, \ldots, L_{(c+1)n}$ und gibt es aus, in Form einer geeignet kodierten Basis des Gitters. Es folgt eine formale Beschreibung der Reduktion:

Algorithmus 4 Reduktion h_c

function $h_c(B,t)$
  $B_0, \ldots, B_{(c+1)n} \leftarrow$ LATTICEGENERATOR$(B, (c+1)n)$
  $B' \leftarrow 2 \cdot B$
  Wähle $i \in \{0, \ldots, (c+1)n+1\}$ zufällig unabhägig gleichverteilt
  if $i = (c+1)n+1$ **then**
    return (B', t)
  end if
  return (B_i, t)
end function

In Satz 3.1.2 wurde gezeigt, dass der Algorithmus LATTICEGENERATOR polynomiell zeitbeschränkt ist in Dimension m, Rang n und Anzahl zu erzeugender Gitter. Letztere ist in diesem Fall $(c+1)n+1$, und ist damit selbst polynomiell beschränkt in n. Die Gitter des Systems (19) werden also in polynomieller Zeit erzeugt. Die Multiplikation der Matrix B mit einem Skalar kann man in Zeit $O(mn)$ durchführen. Auch für das Wählen eines Gitters und kodieren der Ausgabe ist nur polynomieller Zeitaufwand nötig. Die Reduktion h_c ist also polynomiell zeitbeschränkt in m und n.

Im Folgenden wird gezeigt, dass die Reduktion mit einer Wahrscheinlichkeit von mindestens $\Omega(n^{-1})$ Erfolg hat. Soeben wurde festgestellt, dass h_c erfolgreich ist, wenn die Ausgabe $\hat{L}$ genau einen nächsten Vektor zu t besitzt, und dieser auch in L enthalten ist. Da $\hat{L}$ immer ein Untergitter von L ist, ist letztere Bedingung genau dann erfüllt, wenn der nächste Vektor aus $\hat{L}$ den gleichen Abstand zu t hat wie die nächsten Vektoren von L.

Deshalb genügt es, folgende Aussage zu beweisen:

Theorem 5.1.1. *Sei (L,t) die Eingabe von h_c, wobei L ein Gitter vom Rang n und Dimension m, und $t \in \mathbb{Q}^m$ ein Vektor ist. Ist $D_t := \min\left\{\|v-t\|_p \,:\, v \in L\right\}$ der minimale Abstand eines Gittervektors aus L zu t bezüglich einer l_p-Norm mit $p \in (1,\infty)$, und enthält L höchstens 2^{cn} Vektoren mit Abstand D_t zu t, so wird die Reduktion mit einer Wahrscheinlichkeit von mindestens $\Omega(n^{-1})$ ein Ausgabegitter $\hat{L}$ erzeugen, das genau einen Vektor mit Abstand D_t zu t enthält.*

Ziel des restlichen Abschnittes ist es, dieses Theorem zu beweisen. Analog zum Vorgehen in Abschnitt 4.1 werden einige Eigenschaften der nächsten Vektoren bewiesen, so dass man in der Lage ist, die Aussagen des Abschnitts 3.2 über die Konstruktionsschritte des Algorithmus LATTICEGENERATOR zu verwenden.

Es wird sich herausstellen, dass die nächsten Vektoren ähnliche Eigenschaften wie die kürzesten Vektoren haben.

Lemma 5.1.2. *Sei Λ ein Gitter der Dimension m, und $t \in \mathbb{R}^m$ ein Zielvektor. Dann gilt für jedes Paar u und v, $u \neq v$, von nächsten Gittervektoren aus Λ zu t bezüglich einer l_p-Norm mit $p \in (1,\infty)$, dass der Differenzvektor $v-u$ ein ungerader Gittervektor ist.*

Beweis. Der Nachweis lässt sich ähnlich führen wie der von Lemma 4.1.3, welches eine entsprechende Aussage für kürzeste Gittervektoren trifft.

Sei

$$D_t := \min\left\{\|v-t\|_p \,:\, v \in \Lambda\right\},$$

und seien $u,v \in \Lambda$ zwei verschiedene Vektoren mit

$$\|u-t\|_p = \|v-t\|_p = D_t.$$

Angenommen, der Vektor $v-u$ sei gerade. Dann erhält man über die alternative Charakterisierung (5) von geraden Vektoren, dass $\frac{1}{2}(v-u) \in \Lambda$ gilt. Dann ist aber auch $\frac{1}{2}(v+u) = \frac{1}{2}(v-u)+u \in \Lambda$ ein Gittervektor.

Nach der Dreiecksungleichung für Normen gilt:

$$\|v-t\|_p + \|u-t\|_p \geq \|(v-t)+(u-t)\|_p. \tag{20}$$

Einsetzen von D_t für $\|u-t\|_p$ und $\|v-t\|_p$ und umformen liefert:

$$D_t \geq \frac{1}{2}\|v+u-2t\|_p = \left\|\frac{1}{2}(v+u)-t\right\|_p. \tag{21}$$

Als Gittervektor hat $\frac{1}{2}(v+u)$ mindestens Abstand D_t zu t. Also ist die Abschätzung (21) mit Gleichheit erfüllt und es gilt

$$D_t = \left\|\frac{1}{2}(v+u)-t\right\|_p = \frac{1}{2}\|(v-t)+(u-t)\|_p.$$

Damit ist also auch die Dreiecksungleichung (20) mit Gleichheit erfüllt.

Da die l_p-Normen für $p \in (1,\infty)$ streng konvex sind, gilt

$$\forall x,y \in \mathbb{R}^m : \ \Big(\|x\|_p = \|y\|_p \text{ und } \|x\|_p + \|y\|_p = \|x+y\|_p \Big) \implies x = y.$$

Es gilt nun $\|u-t\|_p = \|v-t\|_p$, und (20) ist mit Gleichheit erfüllt. Deshalb folgt $u-t = v-t$. Damit gilt $u = v$, im Widerspruch zur Voraussetzung, dass u und v verschiedene Vektoren sind. □

In Lemma 4.1.2 wurde für die kürzesten Vektoren gezeigt, dass sie primitiv und damit ungerade sind. Dies lässt sich im Allgemeinen für nächste Gittervektoren nicht sagen. Wenn beispielsweise ein gerader Gittervektor als Zielvektor t gewählt wird, so ist er offensichtlich der nächste Gittervektor. Es kann im Allgemeinen also durchaus nächste Gittervektoren geben, die gerade sind. Ohne großen Aufwand lässt sich jedoch zeigen, dass es höchstens einen geraden nächsten Gittervektor geben kann:

Lemma 5.1.3. *Sei Λ ein Gitter der Dimension m, und $t \in \mathbb{R}^m$ ein Zielvektor. Dann enthält Λ höchstens einen geraden Gittervektor mit minimalem Abstand zu t bezüglich jeder l_p-Norm mit $p \in (1,\infty)$.*

Beweis. Man erhält diese Aussage als Folgerung aus Lemma 5.1.2.

Denn der Differenzvektor zweier gerader Gittervektoren ist immer ein gerader Gittervektor. Nach Lemma 5.1.2 ist die Differenz zweier nächster Vektoren jedoch immer ein ungerader Gittervektor. Folglich kann es keine zwei geraden Vektoren mit minimalem Abstand zu t geben. □

Der folgende Satz liefert eine Eindeutigkeitsaussage, die ein Schlüsselargument im Beweis des Theorems sein wird. In Satz 4.1.5 wurde eine entsprechende Eindeutigkeitsaussage für Mengen von Äquivalenzklassen bewiesen. In diesem Fall wird eine Menge von Vektoren betrachtet, da UNIQUE-CVP anders als UNIQUE-SVP Eindeutigkeit von Vektoren verlangt, und nicht Eindeutigkeit bezüglich einer Äquivalenzrelation.

Satz 5.1.4. *Sei $p \in (1,\infty)$ und $T_0,\ldots,T_{(c+1)n}$ ein Mengensystem, welches über den Konstruktionsprozess des Gittersystems* (19) *definiert ist wie folgt:*

$$T_k := \Big\{ v \ : \ v \in L_k,\ v \text{ ungerade in } L,\ \|v-t\|_p = D_t \Big\} \tag{22}$$

mit L_k aus (19)*, und $D_t = \min\Big\{ \|u-t\|_p : u \in L \Big\}$.*

Enthält L mindestens einen nächsten Vektor, der ungerade ist, so wird mit Wahrscheinlichkeit von mindestens $\frac{2}{3} - 2^{-n}$ bei der Konstruktion der L_k ein Gitter erzeugt, so dass die zugehörige Menge des Systems (22) *genau ein Element enthält.*

Beweis. Man erhält diese Aussage direkt durch Anwenden von Satz 3.2.3. Es müssen lediglich alle Voraussetzungen des Satzes geprüft werden. Sie waren gegeben wie folgt:

(a) $0 < |T_0| \leq 2^{cn}$

(b) $T_0 \supseteq T_1 \supseteq \cdots \supseteq T_{(c+1)n}$

(c) $\forall k \in \mathbb{N}, 0 < k \leq (c+1)n \; \forall v \in T_0$: $Pr[v \in T_k \mid v \in T_{k-1}] = \frac{1}{2}$

(d) $\forall k \in \mathbb{N}, 0 < k \leq (c+1)n \; \forall u, v \in T_{k-1}, u \neq v$:
Die Ereignisse $[u \in T_k]$ und $[v \in T_k]$ sind unabhängig.

(e) $\forall v \in T_0 \; \forall k, l \in \mathbb{N}, 0 < k < l \leq (c+1)n$:
Die Ereignisse $[v \in T_k \mid v \in T_{k-1}]$ und $[v \in T_l \mid v \in T_{l-1}]$ sind unabhängig.

Da das Eingabegitter L nach Vorgabe durch FEW-CVP$_c$ höchstens 2^{cn} viele Vektoren mit minimalem Abstand D_t zu t hat, und T_0 nur Vektoren aus L mit minimalem Abstand D_t enthält, folgt $|T_0| \leq 2^{cn}$. Laut Voraussetzung enthält L mindestens einen ungeraden nächsten Gittervektor. Damit gilt $|T_0| > 0$. Zusammen erhält man

$$0 < |T_0| \leq 2^{cn}.$$

Die Bedingung (a) ist also erfüllt.

Da die L_k nach Algorithmus LATTICEGENERATOR konstruiert werden, erfüllen sie

$$L_0 \supseteq L_1 \supseteq \cdots \supseteq L_{(c+1)n}.$$

Sei nun $k \in \mathbb{N}$ mit $0 < k \leq (c+1)n$. Für jeden Vektor $v \in T_k$ gilt nach Definition auch $v \in L_k$. Wegen $L_k \subseteq L_{k-1}$ folgt $v \in L_{k-1}$, und damit $v \in T_{k-1}$. Es gilt also $T_{k-1} \supseteq T_k$ für alle $k \in \mathbb{N}$ mit $0 < k \leq (c+1)n$. Es ergibt sich also die Inklusionskette

$$T_0 \supseteq T_1 \supseteq \cdots \supseteq T_{(c+1)n},$$

was zeigt, dass Bedingung (b) gilt.

Sei im Folgenden $k \in \mathbb{N}$ beliebig gewählt mit $0 < k \leq (c+1)n$.

Gelte $v \in T_{k-1}$. Nach Definition von T_{k-1} bedeutet dies, dass v ein ungerader Gittervektor in L ist, und $v \in L_{k-1}$ gilt. Als ungerader Gittervektor von L ist v auch ungerade in L_{k-1}. Es lässt sich also Lemma 3.2.1 anwenden, und man erhält

$$Pr[v \in L_k] = \frac{1}{2}.$$

Der Vektor v ist genau dann in T_k enthalten, wenn $v \in L_k$ gilt. Damit folgt:

$$Pr[v \in T_k \mid v \in T_{k-1}] = \frac{1}{2}.$$

Die Eigenschaft (c) wird also erfüllt.

Seien $u, v \in T_{k-1}$ zwei Vektoren. Dann handelt es sich nach Definition der T_k um nächste Vektoren in L, und damit auch um nächste Vektoren in L_{k-1}. Nach Lemma 5.1.2 ist ihr Differenzvektor also ein ungerader Vektor, und Lemma 3.2.2 liefert, dass die Ereignisse $[u \in L_k]$ und $[v \in L_k]$ unabhängig sind. Da diese Ereignisse genau dann eintreten, wenn $[u \in T_k]$ und $[v \in T_k]$ dies tun, sind auch letztere Ereignisse unabhängig. Damit ist auch die Eigenschaft (d) erfüllt.

Für beliebige $k, l \in \mathbb{N}$ mit $0 < k < l \leq (c+1)n$ und $v \in L$ sind die Ereignisse $[v \in L_k \mid v \in L_{k-1}]$ und $[v \in L_l \mid v \in L_{l-1}]$ unabhängig, da in jedem Konstruktionsschritt der L_k die Menge W unabhängig neu gewählt wird. Analog zur Argumentation zur Eigenschaft (d) erhält man, dass dann auch die Ereignisse $[v \in T_k \mid v \in T_{k-1}]$ und $[v \in T_l \mid v \in T_{l-1}]$ unabhängig sind für $k \neq l$. Es gilt also auch die Eigenschaft (e).

Das Mengensystem (22) erfüllt also alle Voraussetzungen von Satz 3.2.3. Anwenden des Satzes liefert die Behauptung. □

Es folgt der Beweis des Theorems:

Beweis. [von Theorem 5.1.1] Man hat drei Fälle zu unterscheiden. Im ersten Fall gelte $t \in L$. Dann gibt es genau einen Gittervektor in L mit minimalem Abstand zu t, nämlich t selbst. Das bedeutet, dass (mindestens) $L_0 = L$ die gewünschte Eigenschaft erfüllt. Die Wahrscheinlichkeit, dass L_0 unter allen $(c+1)n+2$ Gittern gewählt wird, ist $\frac{1}{(c+1)n+2} = \Omega(n^{-1})$. Im ersten Fall erfüllt h_c also die Anforderung.

Im zweiten Fall gelte $t \notin L$, und es gebe einen geraden nächsten Gittervektor v zu t in L. Nach Lemma 5.1.3 gibt es höchstens einen solchen Vektor. Alle anderen nächsten Vektoren zu t in L sind ungerade. Das Gitter L' ist so definiert, dass nur die geraden Vektoren von L in L' enthalten sind. Demnach enthält L' genau einen nächsten Gittervektor zu t, nämlich v. Mindestens L' erfüllt also die gewünschte Eigenschaft. Auch hier gilt, dass die Wahrscheinlichkeit, dass L' unter allen $(c+1)n+2$ Gittern gewählt wird, $\frac{1}{(c+1)n+2} = \Omega(n^{-1})$ ist. Auch in diesem Fall erfüllt h_c also die Anforderung.

Im dritten und letzten Fall gelte $t \notin L$, und alle nächsten Gittervektoren sind ungerade. Nach Satz 5.1.4 hat eine der $(c+1)n+1$ Mengen T_k mit Wahrscheinlichkeit mindestens $\frac{2}{3} - 2^{-n}$ genau ein Element. Ist dies der Fall, so enthält das zugehörige Gitter L_k genau einen ungeraden nächsten Vektor. Da es in diesem Fall keinen nächsten Gittervektor gibt, der gerade ist, enthält

L_k dann also genau einen nächsten Vektor. Dieses Gitter wird mit Wahrscheinlichkeit $\frac{1}{(c+1)n+2}$ gewählt. Man erhält also eine untere Schranke für den Erfolg von h_c durch

$$Pr[\text{Ein Gitter hat genau einen Vektor mit minimalem Abstand und wird gewählt}] \geq (\frac{2}{3} - 2^{-n}) \cdot \frac{1}{(c+1)n+2} = \Omega(n^{-1})$$

Auch in diesem Fall erfüllt h_c die Anforderung. Damit ist die Behauptung gezeigt. □

Damit ist gezeigt, dass für l_p-Normen mit $p \in (1,\infty)$ die Probleme FEW-CVP$_c$ und UNIQUE-CVP gleichschwierig sind.

5.2 Untersuchung der Normen l_1 und l_∞

Die l_p-Norm ist für $p = 1$ und $p = \infty$ nicht streng konvex. Genau das war aber das Kernargument im Beweis von Lemma 5.1.2, welches aussagte, dass der Differenzvektor zweier nächster Gittervektoren ein ungerader Vektor ist. Auf diesem Lemma basierte auch der Beweis dafür, dass es höchstens einen nächsten Gittervektor gibt, der gerade ist.

Dass diese Aussagen im Allgemeinen nicht richtig sind für die l_1-Norm und l_∞-Norm, macht man sich leicht an folgenden zwei Beispielen klar:

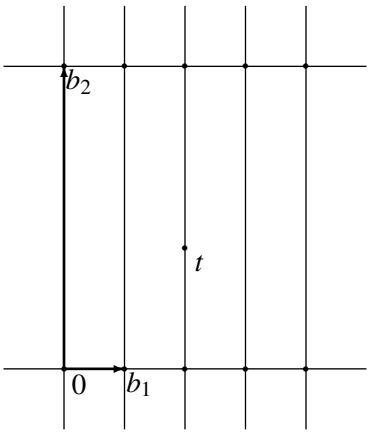

Abbildung 4: Gegenbeispiel für die l_∞-Norm

Beispiel 1: Gegenbeispiel für die l_∞-Norm Seien

$$b_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ und } b_2 = \begin{pmatrix} 0 \\ 5 \end{pmatrix}$$

eine Gitterbasis von $\Lambda := \mathcal{L}(b_1, b_2)$, und $t = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \in \mathbb{Q}^2$ ein Vektor. Das Gitter Λ (siehe Abbildung 4) enthält dann fünf nächste Vektoren zu t bezüglich der l_∞-Norm, nämlich

$$v_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix},\ v_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix},\ v_3 = \begin{pmatrix} 2 \\ 0 \end{pmatrix},\ v_4 = \begin{pmatrix} 3 \\ 0 \end{pmatrix} \text{ und } v_5 = \begin{pmatrix} 4 \\ 0 \end{pmatrix},$$

welche allesamt den Abstand 2 haben.

Nun sind sowohl $v_1 = 0 \cdot b_1 + 0 \cdot b_2$ als auch $v_3 = 2 \cdot b_1 + 0 \cdot b_2$ und $v_5 = 4 \cdot b_1 + 0 \cdot b_2$ gerade Gittervektoren. Es gibt also mehr als einen nächsten Vektor in Λ, der gerade ist.

Außerdem ist $v_4 - v_2 = \begin{pmatrix} 2 \\ 0 \end{pmatrix} = 2 \cdot b_1 + 0 \cdot b_2$ ein gerader Vektor in Λ. Der Differenzvektor von zwei nächsten Vektoren ist also nicht immer ungerade. Sogar für den Differenzvektor von zwei nächsten Vektoren, die selbst ungerade sind, ist dies nicht der Fall.

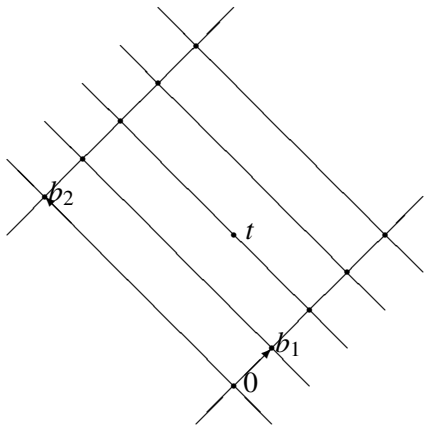

Abbildung 5: Gegenbeispiel für die l_1-Norm

Beispiel 2: Gegenbeispiel für die l_1-Norm Seien

$$b_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{ und } b_2 = \begin{pmatrix} -5 \\ 5 \end{pmatrix}$$

eine Gitterbasis von $\Lambda := \mathcal{L}(b_1, b_2)$, und $t = \begin{pmatrix} 0 \\ 4 \end{pmatrix} \in \mathbb{Q}^2$ ein Vektor. Das Gitter Λ (siehe Abbildung 5) enthält dann fünf nächste Vektoren zu t bezüglich der l_1-Norm, nämlich

$$v_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix},\ v_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix},\ v_3 = \begin{pmatrix} 2 \\ 2 \end{pmatrix},\ v_4 = \begin{pmatrix} 3 \\ 3 \end{pmatrix} \text{ und } v_5 = \begin{pmatrix} 4 \\ 4 \end{pmatrix},$$

welche allesamt den Abstand 4 haben.

Sowohl $v_1 = 0 \cdot b_1 + 0 \cdot b_2$, als auch $v_3 = 2 \cdot b_1 + 0 \cdot b_2$ und $v_5 = 4 \cdot b_1 + 0 \cdot b_2$ sind gerade Gittervektoren. Es gibt also mehr als einen nächsten Vektor in Λ, der gerade ist.

Außerdem ist $v_4 - v_2 = \begin{pmatrix} 2 \\ 2 \end{pmatrix} = 2 \cdot b_1 + 0 \cdot b_2$ ein gerader Vektor in Λ. Der Differenzvektor von zwei nächsten Vektoren ist also nicht immer ungerade. Sogar für den Differenzvektor von zwei nächsten Vektoren, die selbst ungerade sind, ist dies nicht der Fall.

Unter diesen Umständen ist es nicht möglich, den Korrektheitsbeweis für die Reduktion h_c wie in 5.1 für die l_1- und l_∞-Normen zu führen. Da es nun mehr als einen geraden nächsten Gittervektor geben kann, müsste man sie nun bei der Analyse des Konstruktionsprozesses für die Gitter in (19) beachten. Gerade Gittervektoren bleiben bei der Konstruktion eines Untergitters in der Form (9) jedoch immer erhalten. Es kann aber passieren, dass sie im Untergitter nicht mehr gerade sind. Man müsste diesen Sachverhalt genauer untersuchen, um Aussagen über die „Überlebenswahrscheinlichkeit“ der geraden Gittervektoren im gesamten Konstruktionsprozess zu erhalten.

Da der Differenzvektor zweier nächster Vektoren nicht mehr unbedingt ungerade ist, ist auch nicht mehr gegeben, dass für zwei nächste Vektoren u und v die Ereignisse $[u \in L_{k+1} \mid u \in L_k]$ und $[v \in L_{k+1} \mid v \in L_k]$ unabhängig sind. Zwei Vektoren, für die diese Ereignisse abhängig sind, „überleben“ entweder beide den Konstruktionsschritt, oder keiner von beiden. Deshalb müssen alle diese Vektoren im Untergitterkonstruktionsprozess entfernt werden, damit die Reduktion erfolgreich sein kann. Man müsste also untersuchen, wie groß der Anteil solcher Vektoren an der Gesamtzahl aller nächsten Vektoren maximal sein kann, um schließlich Aussagen über die Wahrscheinlichkeit dafür treffen zu können, dass alle diese Vektoren entfernt werden.

Es kann durchaus sein, dass sich nach derartigen Untersuchungen herausstellt, dass die Reduktion h_c auch für die l_1- und l_∞-Normen korrekt funktioniert. Jedoch würde eine weitere Untersuchung dieses Sachverhaltes den Rahmen dieser Arbeit übersteigen.

5.3 Das Verhältnis zwischen CVP und FEW-CVP$_c$

In diesem Abschnitt wird das Verhältnis zwischen CVP und FEW-CVP$_c$ untersucht werden. Es lässt sich zeigen, dass beide Probleme für die l_p-Normen mit $p \in (1, \infty)$ übereinstimmen für alle $c \in \mathbb{N}$. Der Beweis orientiert sich am Beweis von Lemma 4.1.4, in welchem gezeigt wurde, dass es höchstens 2^n kürzeste Vektoren gibt.

Satz 5.3.1. *Sei Λ ein Gitter vom Rang n und Dimension m, und sei die zugrundeliegende Norm eine l_p-Norm mit $p \in (1, \infty)$.*

Dann enthält Λ für jeden Zielvektor $t \in \mathbb{Q}^m$ höchstens 2^n nächste Vektoren. Insbesondere sind die Probleme CVP *und* FEW-CVP$_c$ *identisch für jedes $c \in \mathbb{N}$.*

Beweis. Sei B eine Basis von Λ, und v ein Gittervektor mit Koeffizientenvektor x bezüglich B. Man erinnere sich an die Definition des Paritätsvektors von v, wie sie im Beweis von Lemma

4.1.4 eingeführt wurde: Der Paritätsvektor $p(v) = (p_1, p_2, \ldots, p_n)^T \in \mathbb{Z}_2^n$ eines Gittervektors v ist komponentenweise definiert wie folgt:

$$p_i := \begin{cases} 0, \text{ falls } x_i \equiv 0 \mod 2 \\ 1, \text{ sonst} \end{cases}$$

Sei nun $t \in \mathbb{Q}^m$ ein Zielvektor, und $u = Bx$ und $v = By$ zwei nächste Vektoren mit Koeffizientenvektoren x und y.

Angenommen, u und v hätten den gleichen Paritätsvektor. Dann wäre jede Komponente x_i eine gerade Zahl genau dann, wenn y_i dies wäre. Das bedeutet aber, dass der Vektor $y - x$ nur gerade Komponenten enthält. $y - x$ ist der Koeffizientenvektor von $v - u$. Damit ist $v - u$ ein gerader Gittervektor im Widerspruch zur Aussage von Lemma 5.1.2, nach der dieser Gittervektor ungerade ist für l_p-Normen mit $p \in (1, \infty)$.

Je zwei nächste Vektoren besitzen also nie den gleichen Paritätsvektor. Damit kann es höchstens so viele verschiedene nächste Gittervektoren geben, wie es verschiedene Paritätsvektoren gibt. Die Anzahl an verschiedenen Paritätsvektoren wiederum ist gegeben durch die Anzahl an Vektoren in $\mathbb{Z}_2^n$, nämlich 2^n. □

Da für die Probleme FEW-CVP$_c$ und UNIQUE-CVP gezeigt wurde, dass sie gleichschwierig zu lösen sind, folgt dies nun auch für die Probleme CVP und UNIQUE-CVP. Diese Betrachtungen gelten aber nur für l_p-Normen mit $p \in (1, \infty)$.

Für die l_1- und l_∞-Normen hingegen kann man Beispiele finden, so dass für jede Dimension $n > 1$ und jede Konstante $c \in \mathbb{N}$ ein Tupel (Λ, t) existiert, bestehend aus einem Gitter Λ vom Rang n und einem Zielvektor $t \in \mathbb{Q}^n$, so dass es mehr als 2^{cn} viele nächste Vektoren in Λ gibt.

Für die l_∞-Norm betrachte man das Gitter $\Lambda_\infty = \mathcal{L}(e_1, e_2, \ldots, e_{n-1}, 2d \cdot e_n)$ (siehe Abbildung 6), wobei e_i den i-ten Einheitsvektor des $\mathbb{Q}^n$ bezeichne und $d \in \mathbb{Z}$ gelte. Setzt man $t := (0, \cdots, 0, d)^T$, so haben alle Gittervektoren von Λ_∞ der Menge

$$\left\{ v \in \Lambda_\infty : v = \sum_{i=1}^{n-1} x_i e_i + x_n \cdot (2de_n),\ |x_i| \leq d \text{ für alle } i \in \{1, 2, \ldots, n-1\},\ x_n \in \{0, 1\} \right\}$$

minimalen Abstand d zu t, sind also nächste Gittervektoren. Diese Menge hat $2(2d+1)^{n-1}$ Elemente. Zu jedem $c \in \mathbb{N}$ kann man d so wählen, dass $2(2d+1)^{n-1} > 2^{cn}$ gilt.

Für die l_1-Norm betrachte man das Gitter $\Lambda_1 = \mathcal{L}(b_1, b_2, \ldots, b_n)$ (siehe Abbildung 7) mit

$$b_i := \begin{cases} e_1 + e_{i+1}, & \text{falls } i < n \\ 2d\left((\sum_{j=2}^{n} e_j) - e_1\right), & \text{falls } i = n \end{cases},$$

wobei e_i den i-ten Einheitsvektor des $\mathbb{Q}^n$ bezeichne und $d \in \mathbb{Z}$ gelte. Setzt man

$$t := (-d, d, d, \cdots, d)^T,$$

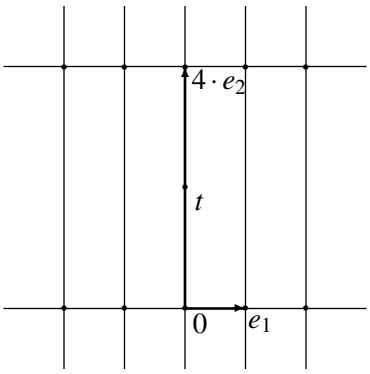

Abbildung 6: Das Gitter Λ_∞ für $n = 2$ und $d = 2$

so haben mindestens die Gittervektoren aus der Menge

$$\left\{ v \in \Lambda_1 : v = \sum_{i=1}^{n-1} x_i \cdot b_i,\ 0 \leq x_i \leq d \text{ für alle } i \in \{1,2,\ldots,n-1\} \right\}$$

minimalen Abstand nd zu t, sind also nächste Gittervektoren. Diese Menge hat $(d+1)^{n-1}$ Elemente. Zu jedem $c \in \mathbb{N}$ kann man d so wählen, dass $(d+1)^{n-1} > 2^{cn}$ gilt.

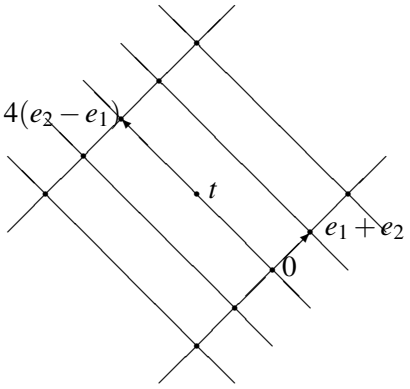

Abbildung 7: Das Gitter Λ_1 für $n = 2$ und $d = 2$

Man sieht also, dass die Probleme FEW-CVP$_c$ und CVP für die l_1- und l_∞-Normen verschieden sind. Für alle l_p-Normen mit $p \in (1,\infty)$ hingegen wurde gezeigt, dass die Suchvarianten der Probleme CVP, FEW-CVP$_c$ und UNIQUE-CVP gleichschwierig zu lösen sind. Man kann jedes der drei Probleme mit höchstens polynomiellem Mehraufwand mit Hilfe der anderen Probleme lösen.

A Anhang

A.1 Beweis der strengen Konvexität der l_p-Normen für $p \in (1,\infty)$

Um zu zeigen, dass die l_p-Norm für $p \in (1,\infty)$ streng konvex ist, ist zu beweisen, dass alle Vektoren gleicher Länge, die die Dreiecksungleichung mit Gleichheit erfüllen, selbst gleich sind:

$$x,y \in \mathbb{Q}^n,\ \|x\|_p = \|y\|_p,\ \|x+y\|_p = \|x\|_p + \|y\|_p \implies x = y. \tag{23}$$

Dazu betrachte man einen Beweis dafür, dass die l_p-Norm die Dreiecksungleichung erfüllt. Er erfolgt in mehreren Schritten. Zunächst zeigt man die *Youngsche Ungleichung*:

$$\forall x,y \in \mathbb{R},\ x,y \geq 0 :\ \ xy \leq \frac{x^p}{p} + \frac{y^q}{q}.$$

Daraus folgert man die *Höldersche Ungleichung*:

$$\forall x,y \in \mathbb{R}^n :\ \ \sum_{i=1}^{n} |x_i y_i| \leq \|x\|_p \|y\|_p .$$

Mit Hilfe der Hölderschen Ungleichung kann man schließlich die *Minkowskische Ungleichung* beweisen, welche direkt die Dreiecksungleichung impliziert:

$$\forall x,y \in \mathbb{R}^n :\ \ \|x+y\|_p \leq \|x\|_p + \|y\|_p .$$

Diese Beweisschritte werden im Folgenden im Detail nachvollzogen. Ein ähnlicher Beweis ist auch in [Kön00] nachzulesen. Da jedoch weniger der Beweis der Dreiecksungleichung selbst, sondern eher das Aussehen der Vektoren, die sie mit Gleichheit erfüllen, von Interesse ist, werden in den Beweisen die Fälle, in denen Abschätzungen mit Gleichheit erfüllt sind, gesondert betrachtet.

Lemma A.1.1. *Seien $p,q \in (1,\infty)$ und gelte $\frac{1}{p} + \frac{1}{q} = 1$. Dann gilt für alle $x,y \in \mathbb{R}$ mit $x,y \geq 0$ die* Youngsche Ungleichung*:*

$$xy \leq \frac{x^p}{p} + \frac{y^q}{q}.$$

Zudem ist diese Abschätzung genau dann mit Gleichheit erfüllt, falls $x^p = y^q$ gilt.

Beweis. Falls $x = 0$ oder $y = 0$ gilt, so folgt die Behauptung sofort. Gelte also $x \neq 0$ und $y \neq 0$. Der natürliche Logarithmus

$$\ln : (0,\infty) \longrightarrow \mathbb{R}$$

ist eine streng konkave Funktion, da $\ln''(x) = -\frac{1}{x^2} < 0$ gilt für alle $x \in (0,\infty)$. Es gilt also für alle $a,b \in (0,\infty)$ und alle $d \in (a,b)$:

$$\ln(d) > \ln(a) + \frac{\ln(a) - \ln(b)}{b-a}(d-a).$$

Falls $x^p \neq y^q$, so folgt wegen $\frac{1}{p}+\frac{1}{q}=1$, dass $\frac{1}{p}x^p+\frac{1}{q}y^q \in (x^p, y^q)$ gilt. Man erhält:

$$\begin{aligned}
\ln\left(\tfrac{1}{p}x^p+\tfrac{1}{q}y^q\right) &> \ln(x^p)+\frac{\ln(x^p)-\ln(y^q)}{x^p-y^q}\left(\frac{1}{p}x^p+\frac{1}{q}y^q-x^p\right)\\
&= \ln(x^p)+\frac{\ln(x^p)-\ln(y^q)}{x^p-y^q}\left(\frac{1}{q}y^q-\frac{1}{q}x^p\right)\\
&= \ln(x^p)-\frac{1}{q}(\ln(x^p)-\ln(y^q))\\
&= \frac{1}{p}\ln(x^p)+\frac{1}{q}\ln(y^q)=\ln x+\ln y\\
\Leftrightarrow \quad e^{\ln\left(\frac{1}{p}x^p+\frac{1}{q}y^q\right)} &> e^{\ln x+\ln y}\\
\Leftrightarrow \quad \tfrac{1}{p}x^p+\tfrac{1}{q}y^q &> xy
\end{aligned}$$

Falls $x^p = y^q$, so gilt $y = x^{\frac{p}{q}}$. Außerdem gilt $\frac{1}{p}+\frac{1}{q}=1 \Leftrightarrow \frac{p}{q}=p-1$. Zusammen erhält man:

$$y = x^{p-1}.$$

Also gilt:

$$\begin{aligned}
x\cdot y = x\cdot x^{p-1} &= x^p\\
&= 1\cdot x^p\\
&= \left(\frac{1}{p}+\frac{1}{q}\right)x^p\\
&= \frac{x^p}{p}+\frac{x^p}{q}\\
&= \frac{x^p}{p}+\frac{y^q}{q}
\end{aligned}$$

In diesem Fall gilt also Gleichheit. Damit ist alles gezeigt. □

Lemma A.1.2. *Seien $p,q \in (1,\infty)$ und gelte $\frac{1}{p}+\frac{1}{q}=1$. Dann gilt für alle $x,y \in \mathbb{R}^n$ die* Hölder*sche Ungleichung:*

$$\sum_{i=1}^{n} |x_i y_i| \leq \|x\|_p \|y\|_p .$$

Falls die Abschätzung mit Gleichheit erfüllt ist, so gilt $\left(\frac{|x_i|}{\|x\|_p}\right)^p = \left(\frac{|y_i|}{\|y\|_q}\right)^q$ gilt für alle $i \in \{1,\ldots,n\}$.

Beweis. Falls $x = 0$ oder $y = 0$ gilt, so folgt die Behauptung direkt. Gelte also $x \neq 0$ und $y \neq 0$. Die Youngsche Ungleichung liefert für alle $i \in \{1,\dots,n\}$:

$$\begin{aligned}\frac{|x_i|}{\|x\|_p} \cdot \frac{|y_i|}{\|y\|_q} &\leq \frac{1}{p}\left(\frac{|x_i|}{\|x\|_p}\right)^p + \frac{1}{q}\left(\frac{|y_i|}{\|y\|_q}\right)^q \\ &= \frac{|x_i|^p}{p\,\|x\|_p^p} + \frac{|y_i|^q}{q\,\|y\|_q^q}\end{aligned} \tag{24}$$

Aufsummieren dieser n Ungleichungen liefert:

$$\begin{aligned}\frac{\sum_{i=1}^n |x_i||y_i|}{\|x\|_p \|y\|_q} &\leq \frac{\sum_{i=1}^n |x_i|^p}{p\,\|x\|_p^p} + \frac{\sum_{i=1}^n |y_i|^q}{q\,\|y\|_q^q} \\ &= \frac{\|x\|_p^p}{p\,\|x\|_p^p} + \frac{\|y\|_q^q}{q\,\|y\|_q^q} = \frac{1}{p} + \frac{1}{q} = 1\end{aligned}$$

Es folgt:

$$\sum_{i=1}^n |x_i||y_i| \leq \|x\|_p \|y\|_q,$$

was zu zeigen war.

Die Gleichheit dieser Abschätzung kann nur dann gelten, wenn alle n aufsummierten Ungleichungen (24) mit Gleichheit erfüllt sind. Dies ist nach Lemma A.1.1 genau dann der Fall, wenn $\left(\frac{|x_i|}{\|x\|_p}\right)^p = \left(\frac{|y_i|}{\|y\|_q}\right)^q$ gilt für alle $i \in \{1,\dots,n\}$. □

Lemma A.1.3. *Sei $p \in (1,\infty)$. Dann gilt für alle $x, y \in \mathbb{R}^n$ die* Minkowskische Ungleichung*:*

$$\|x+y\|_p \leq \|x\|_p + \|y\|_p.$$

Ist sie mit Gleichheit erfüllt und gilt zusätzlich $\|x\|_p = \|y\|_p$, so gilt $x = y$.

Beweis. Seien x und y beliebige Vektoren des $\mathbb{R}^n$, und sei $q := \frac{p}{p-1}$. Es gilt dann $\frac{1}{p} + \frac{1}{q} = 1$.

Es folgt:

$$\begin{aligned}
\|x+y\|_p^p &= \sum_{i=1}^n |x_i+y_i|^p \\
&= \sum_{i=1}^n |x_i+y_i| \cdot |x_i+y_i|^{p-1} \\
&\leq \sum_{i=1}^n |x_i| \cdot |x_i+y_i|^{p-1} + \sum_{i=1}^n |y_i| \cdot |x_i+y_i|^{p-1} \qquad (25) \\
&\overset{\text{Hölder}}{\leq} \|x\|_p \cdot \left(\sum_{i=1}^n |x_i+y_i|^{(p-1)q} \right)^{\frac{1}{q}} + \|y\|_p \cdot \left(\sum_{i=1}^n |x_i+y_i|^{(p-1)q} \right)^{\frac{1}{q}} \qquad (26) \\
&= (\|x\|_p + \|y\|_p) \cdot \left(\sum_{i=1}^n |x_i+y_i|^{(p-1)q} \right)^{\frac{1}{q}} \\
&= (\|x\|_p + \|y\|_p) \cdot \left(\sum_{i=1}^n |x_i+y_i|^{(p-1)\frac{p}{p-1}} \right)^{\frac{p-1}{p}} \\
&= (\|x\|_p + \|y\|_p) \cdot \left(\sum_{i=1}^n |x_i+y_i|^{p} \right)^{\frac{p-1}{p}} \\
&= (\|x\|_p + \|y\|_p) \cdot \|x+y\|_p^{p-1}
\end{aligned}$$

Damit gilt:

$$\|x+y\|_p \leq \|x\|_p + \|y\|_p .$$

Es bleibt der Fall zu untersuchen, in dem die Minkowskische Ungleichung mit Gleichheit erfüllt ist. In obigen Beweis wurden zwei Abschätzungen verwendet, die in diesem Fall also mit Gleichheit erfüllt sind. Da (26) mit Gleichheit erfüllt ist, folgt nach Lemma A.1.2, dass sowohl

$$\left(\frac{|x_i|}{\|x\|_p} \right)^p = \left(\frac{|x_i+y_i|^{p-1}}{\left(\sum_{i=1}^n |x_i+y_i|^{(p-1)q} \right)^{\frac{1}{q}}} \right)^q ,$$

als auch

$$\left(\frac{|y_i|}{\|y\|_p} \right)^p = \left(\frac{|x_i+y_i|^{p-1}}{\left(\sum_{i=1}^n |x_i+y_i|^{(p-1)q} \right)^{\frac{1}{q}}} \right)^q$$

für jedes $i \in \{1,\ldots,n\}$ gilt. Zusammen erhält man:

$$\left(\frac{|x_i|}{\|x\|_p} \right)^p = \left(\frac{|y_i|}{\|y\|_p} \right)^p$$

für jedes $i \in \{1,\ldots,n\}$. Falls zusätzlich $\|x\|_p = \|y\|_p$ haben, folgt also

$$|x_i| = |y_i|$$

für jedes $i \in \{1,\ldots,n\}$.

Da (25) mit Gleichheit erfüllt ist, muss $x_i = 0$ oder $y_i = 0$ oder $\operatorname{sgn}(x_i) = \operatorname{sgn}(y_i)$ gelten für jedes $i \in \{1,\ldots,n\}$.

Beide Überlegungen zusammen liefern $x_i = y_i$ für jedes $i \in \{1,\ldots,n\}$, also

$$x = y.$$

□

Damit wurde gezeigt, dass $\|.\|_p$ die Dreiecksungleichung erfüllt. Zudem wurde gezeigt, dass die Aussage (23) erfüllt ist. Die l_p-Normen sind also für $p \in (1,\infty)$ streng konvex.

B Literatur

[Ajt98] M. Ajtai. The shortest vector problem in l_2 is NP-hard for randomized reductions. In *Proceedings of the 30th annual ACM symposium on Theory of computing*, pages 10–19, 1998.

[AKS01] M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proceedings of the 33th ACM Symposium on Theory of Computing*, pages 601–610, 2001.

[Bab86] L. Babai. On lovasz'lattice reduction and the nearest lattice point problem. *Combinatorica*, pages 6(1):1–13, 1986.

[Blö00] J. Blömer. Closest vectors, successive minima, and dual HKZ-bases of lattices. In *Proceedings of the 27th ICALP, Lecture Notes in Computer Science 1853*, pages 248 – 259. Springer Verlag, 2000.

[BN07] J. Blömer and S. Naewe. Sampling methods for shortest vectors, closest vectors and successive minima. In *Proceedings of the 34th ICALP, Lecture notes in Computer Science*. Springer Verlag, 2007.

[Che52] H. Chernoff. A measure of the asymptotic efficiency for tests of a hypothesis based on the sum of observables. *Annals of Mathematical Statistics 23*, pages 493–507, 1952.

[For84] O. Forster. *Analysis 3 - Integralrechung im $\mathbb{R}^n$ mit Anwendungen*. 3. Auflage. Vieweg Verlag, 1984.

[Hil11] D. Hilbert. *Gesammelte Abhandlungen von Hermann Minkowski*. Chelsea Publishing Company, 1911.

[Kön00] K. Königsberger. *Analysis 1*. 5. Auflage. Springer Verlag, 2000.

[KS99] R. Kumar and D. Sivakumar. A note on the shortest lattice vector problem. In *IEEE Conference on Computational Complexity*, pages 200–204, 1999.

[Len83] H. W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, pages 8:538–548, 1983.

[LLL82] A. Lenstra, H. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, pages 261:515–534, 1982.

[MG02] D. Micciancio and S. Goldwasser. *Complexity of Lattice Problems - A Cryptographic Perspective*. Kluwer Academic Publishers, 2002.

[Pap94] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[Reg04] O. Regev. Lecture note on lattice in computer science, lecture 1, 2004.

[vEB81] P. van Emde Boas. Another NP-complete problem and the complexity of computing short vectors in a lattice. Technical Report 81-04, Mathematische Instituut, University of Amsterdam, 1981.

[VV86] L. Valiant and V. Vazirani. NP ist as easy as detecting unique solutions. *Theoretical Computer Science*, pages 47:85–93, 1986.