

# Additive Cutting-Verfahren für ganzzahlige Programme

## Diplomarbeit

vorgelegt von

**Jonas Ide**

aus

Achim

angefertigt

im Institut für Numerische und Angewandte Mathematik  
der Georg-August-Universität zu Göttingen

2010





In erster Linie möchte ich mich sehr herzlich bei Prof. Dr. Anita Schöbel für ihre große Hilfsbereitschaft, ihre wertvollen Ideen und die Möglichkeit zur freien Themengestaltung bedanken.

Ebenfalls danken möchte ich meinen Kommilitonen und Freunden Rebekka Brink-Spalink, Ulrike Ober und Marco Oesting für ihren Rat bei der Fertigstellung dieser Arbeit.

Und schließlich möchte ich mich bei meinen Eltern dafür bedanken, dass sie mich all die Jahre von ganzem Herzen unterstützt haben und mich in Freiheit meinen Lebensweg finden ließen.



# Inhaltsverzeichnis

<b>0</b>	<b>Notationen und Hilfssätze</b>	<b>1</b>
0.1	Notationen . . . . .	1
0.2	Definitionen . . . . .	2
0.3	Hilfssätze . . . . .	3
<b>1</b>	<b>Einleitung und Motivation</b>	<b>9</b>
<b>2</b>	<b>Die <math>\{1,0,-1\}</math>-Form</b>	<b>11</b>
2.1	Das additive Verfahren . . . . .	12
2.2	Das Verfahren der Binärcodierung . . . . .	17
<b>3</b>	<b>Cutting-Verfahren</b>	<b>23</b>
3.1	Das Abrundungs-Verfahren . . . . .	23
3.2	Das 2-Additive Cutting . . . . .	24
3.2.1	2-Additives Cutting im 2-dimensionalen Raum . . . . .	30
3.2.2	2-Additives Cutting im 3-dimensionalen Raum . . . . .	36
3.2.3	Fragestellungen zum 2-Additiven Cutting . . . . .	43
3.3	Das k-Additive Cutting . . . . .	53
3.3.1	Unterschiede zwischen dem k-Additiven Cutting und anderen Cutting-Verfahren . . . . .	54
<b>4</b>	<b>Fazit</b>	<b>57</b>
	<b>Literaturverzeichnis</b>	<b>i</b>



# 0 Notationen und Hilfssätze

## 0.1 Notationen

In dieser Arbeit werden einige Notationen verwendet, die mehr oder weniger bekannt und verbreitet sind.

Um Missverständnissen vorzubeugen, sind sie an dieser Stelle zusammengefasst.

**Notation 0.1.** Sei

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & \ddots & & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

mit  $m, n \in \mathbb{N}$  eine  $m \times n$ -Matrix.

Wir schreiben für alle  $j \in \{1, \dots, m\}$  kurz  $a_j$  für die Zeile  $(a_{j1}, \dots, a_{jn})$  und für alle  $i \in \{1, \dots, n\}$  kurz  $A_i$  für die Spalte  $(a_{1i}, \dots, a_{mi})^t$ .

**Notation 0.2.** Den Vektor  $x \in \mathbb{R}^n$  mit  $x_i = 0$  für alle  $i \in \{1, \dots, n\}$  schreiben wir als  $\mathbf{0}_n$ .

Den Vektor  $x \in \mathbb{R}^n$  mit  $x_i = 1$  für alle  $i \in \{1, \dots, n\}$  schreiben wir als  $\mathbf{1}_n$ .

**Notation 0.3.** Sei für die Unbekannte  $x \in \mathbb{R}^n$  ein Ungleichungssystem

$$Ax \leq b$$

mit  $A \in \mathbb{R}^{m \times n}$  und  $b \in \mathbb{R}^m$  gegeben.

Wir schreiben für dieses Ungleichungssystem auch  $(A|b)$ , für die Ungleichung

$$a_i x \leq b_i$$

auch  $(a_i|b_i)$  und für die Hyperebene

$$\{x \in \mathbb{R}^n : a_i x = b_i\}$$

auch  $H_i$ .

Der Einfachheit halber betrachten wir in manchen Fällen  $(A|b)$  auch als Matrix und  $(a_i|b_i)$  auch als Vektor.

## 0.2 Definitionen

In der mathematischen Literatur finden sich viele verschiedene Definitionen für dieselben Begriffe.

Deshalb werden hier einige dieser Definitionen gegeben, um klarzustellen, in welchem Sinne diese Begriffe zu verstehen sind.

**Definition 0.4.** Die Funktion  $\text{sgn} : \mathbb{R} \rightarrow \{-1, 1\}$  ist definiert durch:

$$\text{sgn}(x) := \begin{cases} 1 & , \text{ falls } x = 0 \\ \frac{x}{|x|} & \text{sonst} \end{cases}$$

**Definition 0.5.** Die Funktion  $\lfloor \cdot \rfloor : \mathbb{R} \rightarrow \mathbb{Z}$  definieren wir als die untere **Gauss-Klammer**, also für  $x \in \mathbb{R}$  ist  $\lfloor x \rfloor$  die größte Zahl in  $\mathbb{Z}$ , die kleiner oder gleich  $x$  ist. Für  $x = (x_1, \dots, x_n)^t \in \mathbb{R}^n$  schreiben wir  $(\lfloor x_1 \rfloor, \dots, \lfloor x_n \rfloor)^t$  kurz als  $\lfloor x \rfloor$ .

**Definition 0.6.** Wir definieren die Funktionen

$$p : \mathbb{Q} \cong \mathbb{Z} \times \mathbb{N} \rightarrow \mathbb{Z} \\ (a, b) \mapsto \frac{a}{\text{ggT}(a, b)}$$

und

$$q : \mathbb{Q} \cong \mathbb{Z} \times \mathbb{N} \rightarrow \mathbb{N} \\ (a, b) \mapsto \frac{b}{\text{ggT}(a, b)}.$$

**Definition 0.7.** Eine Menge  $\{x \in \mathbb{R}^n : a^t x \leq b\}$  mit  $a \in \mathbb{R}^n \setminus \{0\}$  und  $b \in \mathbb{R}$  heißt **Halbraum**.

**Definition 0.8.** Ein Schnitt endlich vieler Halbräume heißt **Polyeder**.

**Definition 0.9.** Ein Polyeder  $P \subset \mathbb{R}^n$  heißt **volldimensional** genau dann, wenn die maximale Anzahl affin unabhängiger Punkte in  $P$  genau  $n + 1$  ist.

**Definition 0.10.** Sei  $M \subset \mathbb{R}^n$ .

Eine Menge  $X \subset M$  heißt **Extremalmenge von  $M$**  genau dann, wenn für alle  $x_1, x_2 \in M$  und  $\lambda \in (0, 1)$  die Bedingung

$$(\lambda \cdot x_1 + (1 - \lambda) \cdot x_2 \in X) \Rightarrow (x_1, x_2 \in X)$$

gilt.

Eine  $(n - 1)$ -dimensionale Extremalmenge heißt **Facette**.

Besitzt eine Extremalmenge genau ein Element, so heißt dieses **Extrempunkt**.



**Definition 0.11.** Sei  $M \subset \mathbb{R}^n$  eine Menge von Punkten.

Dann heißt

$$\text{conv}(M) := \{x \in \mathbb{R}^n : x = \lambda_1 x_1 + \cdots + \lambda_m x_m$$

$$\text{mit } m \in \mathbb{N}, x_1, \dots, x_m \in M, \lambda_1, \dots, \lambda_m \in \mathbb{R}_{\geq 0} : \sum_{i=1}^m \lambda_i = 1\}$$

die **konvexe Hülle** von  $M$ .

### 0.3 Hilfssätze

Es werden in dieser Arbeit an unterschiedlichen Stellen Hilfssätze verwendet, die in keinem direkten Zusammenhang mit dem restlichen Inhalt stehen.

Da sie für den jeweiligen Beweisverlauf jedoch essentiell sind, sind sie an dieser Stelle aufgeführt.

**Lemma 0.12.** ([1]) Seien  $U, V \subset \mathbb{R}^n$  zwei Mengen.

Dann gilt:

$$\text{conv}(U \times V) = \text{conv}(U) \times \text{conv}(V)$$

*Beweis.* “ $\subset$ “: Sei  $(u, v) \in \text{conv}(U \times V)$ .

Dann existieren nach Definition der konvexen Hülle reelle Koeffizienten

$$\lambda_1, \dots, \lambda_n \in \mathbb{R}_{\geq 0}$$

mit

$$\sum_{i=1}^n \lambda_i = 1$$

und Punkte  $(u_1, v_1), \dots, (u_n, v_n) \in U \times V$  so, dass

$$(u, v) = \lambda_1 \cdot (u_1, v_1) + \cdots + \lambda_n \cdot (u_n, v_n).$$

Daraus ergibt sich

$$\begin{aligned} (u, v) &= \lambda_1 \cdot (u_1, v_1) + \cdots + \lambda_n \cdot (u_n, v_n) \\ &= (\lambda_1 u_1, \lambda_1 v_1) + \cdots + (\lambda_n u_n, \lambda_n v_n) \\ &= \underbrace{(\lambda_1 u_1 + \cdots + \lambda_n u_n)}_{\in \text{conv}(U)}, \underbrace{(\lambda_1 v_1 + \cdots + \lambda_n v_n)}_{\in \text{conv}(V)} \in \text{conv}(U) \times \text{conv}(V), \end{aligned}$$

da  $u_1, \dots, u_n \in U$  und  $v_1, \dots, v_n \in V$  und  $\sum_{i=1}^n \lambda_i = 1$ .

Es folgt

$$\text{conv}(U \times V) \subset \text{conv}(U) \times \text{conv}(V). \quad (0.1)$$

“ $\supset$ “: Sei nun  $(u, v) \in \text{conv}(U) \times \text{conv}(V)$ .

Dann existieren wieder nach Definition der konvexen Hülle reelle Koeffizienten

$$\lambda_1, \dots, \lambda_n, \mu_1, \dots, \mu_m \in \mathbb{R}_{\geq 0}$$

mit

$$\sum_{i=1}^n \lambda_i = 1 = \sum_{j=1}^m \mu_j \quad (0.2)$$

und Punkte  $u_1, \dots, u_n \in U$  und  $v_1, \dots, v_m \in V$  so, dass

$$\sum_{i=1}^n \lambda_i u_i = u$$

und

$$\sum_{j=1}^m \mu_j v_j = v.$$

Nun ist

$$\begin{aligned} (u, v) &= \left( \sum_{i=1}^n \lambda_i u_i, v \right) \\ &\stackrel{(0.2)}{=} \sum_{i=1}^n \lambda_i (u_i, v) \\ &= \sum_{i=1}^n \lambda_i \left( u_i, \sum_{j=1}^m \mu_j v_j \right) \\ &\stackrel{(0.2)}{=} \sum_{i=1}^n \lambda_i \sum_{j=1}^m \mu_j (u_i, v_j) \\ &\stackrel{(0.2)}{=} \sum_{i=1}^n \sum_{j=1}^m \lambda_i \mu_j (u_i, v_j). \end{aligned}$$

Da für alle  $i \in \{1, \dots, n\}$  und  $j \in \{1, \dots, m\}$  gilt, dass

$$\lambda_i \mu_j \geq 0$$

und

$$\begin{aligned} \sum_{i=1}^n \sum_{j=1}^m \lambda_i \mu_j &= \sum_{i=1}^n \lambda_i \sum_{j=1}^m \mu_j \\ &\stackrel{(0.2)}{=} \sum_{i=1}^n \lambda_i \cdot 1 \\ &\stackrel{(0.2)}{=} 1, \end{aligned}$$

folgt

$$(u, v) = \sum_{i=1}^n \sum_{j=1}^m \lambda_i \mu_j (u_i, v_j) \in \text{conv}(U \times V)$$

wegen  $(u_i, v_j) \in U \times V$  für alle  $i \in \{1, \dots, n\}$  und  $j \in \{1, \dots, m\}$ . Somit gilt

$$\text{conv}(U \times V) \supset \text{conv}(U) \times \text{conv}(V)$$

und zusammen mit (0.1) folgt die Behauptung. □

**Lemma 0.13.** ([1]) Seien  $a \in \mathbb{R}^n$  und  $b_1, b_2 \in \mathbb{R}$ .

Existiert ein  $x_0 \in \mathbb{R}^n$  mit

$$\{x \in \mathbb{R}^n : a^t x \leq b_1\} \not\ni x_0 \notin \{x \in \mathbb{R}^n : -a^t x \leq b_2\},$$

so gilt

$$\{x \in \mathbb{R}^n : a^t x \leq b_1\} \cap \{x \in \mathbb{R}^n : -a^t x \leq b_2\} = \emptyset.$$

*Beweis.* Wegen

$$x_0 \notin \{x \in \mathbb{R}^n : a^t x \leq b_1\}$$

gilt

$$a^t x_0 > b_1$$

und wegen

$$x_0 \notin \{x \in \mathbb{R}^n : -a^t x \leq b_2\}$$

gilt ebenso

$$-a^t x_0 > b_2 \Leftrightarrow a^t x_0 < -b_2.$$

Zusammen ergibt sich

$$b_1 < a^t x_0 < -b_2. \tag{0.3}$$

Annahme:

$$\{x \in \mathbb{R}^n : a^t x \leq b_1\} \cap \{x \in \mathbb{R}^n : -a^t x \leq b_2\} \neq \emptyset$$

Dann existiert ein Punkt  $x_1 \in \mathbb{R}^n$  mit

$$a^t x_1 \leq b_1 \text{ und } -a^t x_1 \leq b_2 \Leftrightarrow a^t x_1 \geq -b_2,$$

also zusammen

$$b_1 \geq a^t x_1 \geq -b_2.$$

Dies ist aber ein Widerspruch zu (0.3) und es folgt die Behauptung. □

**Lemma 0.14.** ([1]) Seien  $a \in \mathbb{Q}^2 \setminus \{(0,0)\}$ ,  $b \in \mathbb{R}$ .

Existiert ein  $z \in \mathbb{Z}^2$  mit

$$z \in \{x \in \mathbb{R}^2 : a^t x = b\},$$

so enthält die Gerade  $\{x \in \mathbb{R}^2 : a^t x = b\}$  unendlich viele ganzzahlige Punkte.

*Beweis.* Wir zeigen, dass für jedes  $y_0 \in q(a_1) \cdot p(a_2) \cdot \mathbb{Z}$  der Punkt

$$y := \left( z_1 + y_0, z_2 - \frac{a_1 y_0}{a_2} \right)^t$$

die Bedingung

$$y \in \{x \in \mathbb{Z}^2 : a^t x = b\}$$

erfüllt.

Zunächst gilt wegen  $z \in \mathbb{Z}^2$  und  $y_0 \in q(a_1) \cdot p(a_2) \cdot \mathbb{Z}$

$$y = \left( z_1 + y_0, z_2 - \frac{a_1 y_0}{a_2} \right)^t \in \mathbb{Z}^2.$$

Weiter ist

$$\begin{aligned} a^t y &= a_1 \cdot (z_1 + y_0) + a_2 \cdot \left( z_2 - \frac{a_1 y_0}{a_2} \right) \\ &= a_1 z_1 + a_1 y_0 + a_2 z_2 - a_2 \cdot \frac{a_1 y_0}{a_2} \\ &= \underbrace{a_1 z_1 + a_2 z_2}_{=b \text{ nach Voraussetzung}} + \underbrace{a_1 y_0 - a_1 y_0}_{=0} \\ &= b. \end{aligned}$$

Es folgt  $y \in \{x \in \mathbb{Z}^2 : a^t x = b\}$  und somit auch

$$\left| \{x \in \mathbb{Z}^2 : a^t x = b\} \right| \geq |q(a_1) \cdot p(a_2) \cdot \mathbb{Z}| = \infty,$$

wie behauptet. □

**Lemma 0.15.** (Schöbel [9], Satz 2.23) Sei  $F$  eine Facette eines Polyeders

$$P := \{x \in \mathbb{R}^n : Ax \leq b\}$$

mit  $A \in \mathbb{R}^{m \times n}$ . Dann existiert eine Ungleichung  $(a_i | b_i)$  aus  $(A | b)$  so, dass

$$F = P \cap \{x \in \mathbb{R}^n : a_i x = b_i\}.$$

Die Ungleichung  $(a_i | b_i)$  heißt **beschreibende Ungleichung** der Facette  $F$ .

**Satz 0.16.** ([1]) Sei  $x \in \mathbb{R}^n$  ein Extrempunkt eines Polyeders

$$P := \{x \in \mathbb{R}^n : Ax \leq b\}.$$

Dann existieren  $n$  Ungleichungen

$$(a_1|b_1), \dots, (a_n|b_n),$$

die in  $a_1, \dots, a_n$  linear unabhängig sind, so, dass

$$\{x\} = \bigcap_{i=1}^n \{x \in \mathbb{R}^n : a_i x = b_i\}.$$

*Beweis.* Die Behauptung geht aus Jungnickel [4], Satz 2.8.2 hervor, wonach  $n$  Ungleichungen  $(a_1|b_1), \dots, (a_n|b_n)$  so existieren, dass  $\{x\}$  der eindeutige Schnitt der zugehörigen Hyperebenen  $H_1, \dots, H_n$  ist.

Diese Ungleichungen müssen wegen der Eindeutigkeit des Schnittes in  $a_1, \dots, a_n$  linear unabhängig sein.

□



# 1 Einleitung und Motivation

**Definition 1.1.** *Ein Problem der Form*

$$\begin{array}{ll} \min & c^t x \\ \text{s.d.} & Ax \leq b \\ & x \in \mathbb{R}^n \end{array}$$

mit einer Matrix  $A \in \mathbb{R}^{m \times n}$  und Vektoren  $b \in \mathbb{R}^m$  und  $c \in \mathbb{R}^n$  heißt **lineares Programm** oder kurz (LP).

Ist zusätzlich in den Nebenbedingungen  $x \in \mathbb{Z}^n$  gefordert, so heißt das Problem **ganzzahliges lineares Programm** oder kurz (IP) (für integer program).

Zur Lösung eines linearen Programms existieren effiziente Verfahren (siehe u.a. Schöbel [8]), während die Lösung eines allgemeinen ganzzahligen linearen Programms bisher nur in nicht-polynomialem Zeitaufwand (abhängig von der Anzahl der Variablen  $n$ ) möglich ist.

Gerade im Bereich der Verkehrsplanung (zum Beispiel bei der Gestaltung eines Dispositions-Fahrplans) müssen allerdings sehr viele ganzzahlige Programme in möglichst kurzer Zeit gelöst werden.

Aus diesem Grund ist eine Verbesserung des Rechenaufwandes für ein solches (IP) stets von großer Wichtigkeit.

In dieser Arbeit soll nun der zulässige Bereich eines (IP) untersucht werden.

**Definition 1.2.** *Sei ein lineares Programm (LP) wie oben gegeben.*

*Die Menge  $P := \{x \in \mathbb{R}^n : Ax \leq b\}$  heißt der **zulässige Bereich** von (LP).*

Jede Ungleichung  $a_i x \leq b_i$  beschreibt wie in Definition 0.7 einen Halbraum und somit ist  $P$  stets ein Schnitt endlich vieler Halbräume, also nach Definition 0.8 ein Polyeder.

Bei einem ganzzahligen linearen Programm wird nun zusätzlich als Nebenbedingung  $x \in \mathbb{Z}^n$  gefordert, der zulässige Bereich eines (IP) besitzt also immer eine Struktur der Form

$$\mathbb{Z}^n \cap P$$

mit einem Polyeder  $P$ .

**Definition 1.3.** Sei ein ganzzahliges lineares Programm

$$(IP) \quad \begin{array}{ll} \min & c^t x \\ \text{s.d.} & x \in \{x' \in \mathbb{Z}^n : Ax' \leq b\} \end{array}$$

gegeben.

Dann heißt

$$P := \{x' \in \mathbb{R}^n : Ax' \leq b\}$$

der **relaxierte zulässige Bereich** von  $(IP)$ .

Das Programm

$$(LP) \quad \begin{array}{ll} \min & c^t x \\ \text{s.d.} & x \in P \end{array}$$

heißt **LP-Relaxation** von  $(IP)$ .

Nach dem Hauptsatz der Optimierung (siehe Schöbel [8], geometrische Form von Satz 2.21, S.19), kann für jedes beschränkte  $(LP)$  ein Extrempunkt des zulässigen Bereichs gefunden werden, der eine Optimallösung von  $(LP)$  darstellt.

Wären also alle Extrempunkte des relaxierten zulässigen Bereichs eines  $(IP)$  ganzzahlig, so könnte anstelle des (schweren)  $(IP)$  auch seine (wesentlich leichtere) LP-Relaxation gelöst werden, um eine (ganzzahlige) Optimallösung von  $(IP)$  zu erhalten.

Ziel dieser Arbeit ist es, Verfahren herzuleiten, die die Lösung bestimmter ganzzahliger linearer Programme erleichtern können.

Dazu soll der relaxierte zulässige Bereich eines  $(IP)$  durch sogenannte Cuts, also durch Hinzufügen von Ungleichungen, so beschnitten werden, dass seine Extrempunkte ganzzahlig werden, dabei aber keinerlei ganzzahlige (für das  $(IP)$  zulässige) Lösungen abgeschnitten werden.

Gesucht ist also stets die konvexe Hülle der ganzzahligen Punkte des relaxierten zulässigen Bereichs eines  $(IP)$ .

Es wird zunächst eine besondere Form von Matrizen eingeführt, durch die ein Cutting-Verfahren ermöglicht wird, welches anschließend vorgestellt und untersucht werden soll.



## 2 Die $\{1,0,-1\}$ -Form

Wir betrachten im Folgenden ganzzahlige lineare Optimierungsprobleme der Form

$$\begin{aligned} (GLP) \quad & \min c^t x \\ & s.d. \quad Ax \leq b \\ & \quad \quad x \in \mathbb{Z}^n \end{aligned}$$

mit  $A \in \mathbb{Q}^{m \times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n$ .

Da die beschreibende Matrix rational ist, beschränken wir damit die betrachteten Probleme nicht sehr stark.

Es gibt sicherlich einige lineare Probleme, deren Nebenbedingungen sich nicht mit rationaler Matrix  $A$  beschreiben lassen, allerdings trifft dies auf einen großen Teil der heute relevanten Optimierungsprobleme nicht zu.

Außerdem schränken wir die Anzahl der (aktuell) berechenbaren Probleme gar nicht ein, da Computer (zur Zeit) nur rationale Zahlen darstellen können.

Um das angesprochene Cutting-Verfahren umsetzen zu können, benötigen wir allerdings die ganzzahligen linearen Programme in einer speziellen Form, in der  $\{1,0,-1\}$ -Form.

**Definition 2.1 (Die  $\{1,0,-1\}$ -Form).** Sei

$$\begin{aligned} (GLP) \quad & \min c^t x \\ & s.d. \quad Ax \leq b \\ & \quad \quad x \in \mathbb{Z}^n \end{aligned}$$

mit  $A \in \mathbb{Q}^{m \times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n$  ein ganzzahliges lineares Programm.

Wir sagen,  $(GLP)$  ist in  $\{1,0,-1\}$ -Form genau dann, wenn  $A \in \{1,0,-1\}^{m \times n}$ .

Nun mag der erste Eindruck sein, dass die Forderung nach dieser Form sehr viel einschränkender ist als die, dass  $A$  rational sein soll.

Der folgende Satz zeigt aber, dass dieser erste Eindruck täuscht und die beiden Forderungen tatsächlich äquivalent sind.

**Satz 2.2.** Sei  $(GLP)$  wie oben. Dann existiert ein zu  $(GLP)$  äquivalentes lineares Programm

$$\begin{aligned} (GLP) \quad & \min c^{*t} x \\ & s.d. \quad A^* x \leq b^* \\ & \quad \quad x \in \mathbb{Z}^N \end{aligned}$$

in  $\{1,0,-1\}$ -Form.

Dieser Satz rechtfertigt den Ansatz der  $\{1,0,-1\}$ -Form, da wir durch diese die betrachteten Probleme nicht weiter einschränken.

Es werden nun in den folgenden Abschnitten neben dem Beweis von Satz 2.2 zwei Verfahren vorgestellt, um ein ganzzahliges lineares Programm mit rationaler Matrix  $A$  in  $\{1,0,-1\}$ -Form zu bringen.

Diese Verfahren unterscheiden sich in ihrer Ausgabe unter Umständen sehr stark, siehe dazu auch Beispiel 2.7.

## 2.1 Das additive Verfahren

**Beispiel 2.3.** Um dem konstruktiven Beweis von Satz 2.2, in dem das additive Verfahren verwendet wird, später besser folgen zu können, sei an dieser Stelle bereits ein Beispiel für dieses Verfahren gegeben:

Wir formen das Problem

$$(GLP1) \quad \min \quad (1, -5) \cdot x \\ \text{s. d.} \quad \begin{pmatrix} \frac{3}{5} & 0 \\ -\frac{1}{3} & \frac{1}{15} \end{pmatrix} \cdot x \leq \begin{pmatrix} \frac{1}{10} \\ \frac{1}{5} \end{pmatrix} \\ x \in \mathbb{Z}^2$$

in ein Programm in  $\{1,0,-1\}$ -Form um.

**Schritt 1:** Wir multiplizieren  $A := \begin{pmatrix} \frac{3}{5} & 0 \\ -\frac{1}{3} & \frac{1}{15} \end{pmatrix}$  und  $b := \begin{pmatrix} \frac{1}{10} \\ \frac{1}{5} \end{pmatrix}$  mit dem (positiven)

Hauptnenner der Einträge von  $A \in \mathbb{Q}^{2 \times 2}$ . Also

$$A' := 15 \cdot \begin{pmatrix} \frac{3}{5} & 0 \\ -\frac{1}{3} & \frac{1}{15} \end{pmatrix} = \begin{pmatrix} 9 & 0 \\ -5 & 1 \end{pmatrix} \in \mathbb{Z}^{2 \times 2}$$

und

$$b' := 15 \cdot \begin{pmatrix} \frac{1}{10} \\ \frac{1}{5} \end{pmatrix} = \begin{pmatrix} \frac{3}{2} \\ 3 \end{pmatrix}.$$

**Schritt 2:** Wir definieren

$$N := \sum_{i=1}^2 \max \{ |a'_{ji}| : j \in \{1, 2\} \} = 9 + 1 = 10,$$

$N$  ist also die Summe der betragsmäßigen Spaltenmaxima.

Nun schreiben wir die Ungleichungen  $a_i x \leq b_i$  so um, dass wir die einzelnen Koeffizienten als Summe von Einsen schreiben. Wir schreiben also zum Beispiel:

$$\Leftrightarrow \begin{array}{r} -5x_1 + x_2 \leq 3 \\ -x_1 - x_1 - x_1 - x_1 - x_1 + x_2 \leq 3 \end{array}$$

$$\Leftrightarrow (-1, -1, -1, -1, -1, 0, 0, 0, 0, 1) \cdot \begin{pmatrix} x_1 \\ x_1 \\ x_1 \\ x_1 \\ x_1 \\ x_1 \\ x_1 \\ x_1 \\ x_1 \\ x_2 \end{pmatrix} \leq 3$$

Somit setzen wir

$$A'' := \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \in \{1, 0, -1\}^{2 \times 10}.$$

Es ergibt sich folgendes lineares Programm:

$$\begin{array}{ll} (GLP1') & \min (1, 0, 0, 0, 0, 0, 0, 0, 0, -5) \cdot x \\ & s.d. \quad A''x \leq \begin{pmatrix} 3 \\ 2 \\ 3 \end{pmatrix} \\ & x \in \mathbb{Z}^{10} \end{array}$$

**Schritt 3:** Da am Ende der Umformungen  $(GLP1) \Leftrightarrow (GLP1')$  gelten soll, müssen wir in  $(GLP1')$  noch

$$x'_1 = x'_2 = x'_3 = x'_4 = x'_5 = x'_6 = x'_7 = x'_8 = x'_9$$

fordern.

Mit

$$A''' := \left( \begin{array}{cccccccc|c} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 \\ \hline 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array} \right) \quad \text{und } b'' := \begin{pmatrix} \frac{3}{2} \\ 3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

wobei  $A''' \in \{1,0,-1\}^{18 \times 10}$ , erkennen wir für

$$\begin{aligned} (GLP1'') \quad & \min \quad (1, 0, 0, 0, 0, 0, 0, 0, 0, -5) \cdot x \\ & \text{s.d. } A'''x \leq b'' \\ & x \in \mathbb{Z}^{10} \end{aligned}$$

die Gültigkeit der Aussage  $(GLP1) \Leftrightarrow (GLP1'')$ .

Da wir nun mit dem additiven Verfahren vertraut sind, folgt an dieser Stelle der Beweis von Satz 2.2:

Beweis von Satz 2.2. Sei zunächst  $(A|b) \in \mathbb{Q}^{m \times n} \times \mathbb{R}^m$ .

Sei  $h \in \mathbb{N}$  der (positive) Hauptnenner der Einträge von  $A$ .

Dann ist  $(A'|b') := h \cdot (A|b) \in \mathbb{Z}^{m \times n} \times \mathbb{R}^m$ .

Weiterhin gilt offensichtlich

$$\{x \in \mathbb{R}^n : Ax \leq b\} = \{x \in \mathbb{R}^n : A'x \leq b'\}.$$

Unser Ziel ist nun, die Ungleichungen  $a'_j x \leq b'_j$  derart umzuschreiben, dass diese eine  $\{1,0,-1\}$ -Struktur erhalten.

Dazu definieren wir zunächst für  $i \in \{1, \dots, n\}$  den Wert

$$\text{count}_i := \max \{ |a'_{ji}| : j \in \{1, \dots, m\} \}.$$

$count_i$  ist also als Maximum der Beträge der Einträge von Spalte  $i$ .

Unsere neue Matrix  $A''$  definieren wir nun über ihre Zeilen:

Für  $j \in \{1, \dots, m\}$  setzen wir

$$a''_j := \underbrace{(sgn(a'_{j1}) \cdot \mathbf{1}_{|a'_{j1}|}^t, \mathbf{0}^t_{count_1 - |a'_{j1}|}, \dots, sgn(a'_{jn}) \cdot \mathbf{1}_{|a'_{jn}|}^t, \mathbf{0}^t_{count_n - |a'_{jn}|})}_{count_1 \text{ Einträge}} \cdot \underbrace{\phantom{(sgn(a'_{j1}) \cdot \mathbf{1}_{|a'_{j1}|}^t, \mathbf{0}^t_{count_1 - |a'_{j1}|}, \dots, sgn(a'_{jn}) \cdot \mathbf{1}_{|a'_{jn}|}^t, \mathbf{0}^t_{count_n - |a'_{jn}|})}}_{count_n \text{ Einträge}}.$$

Weiterhin definieren wir

$$N := \sum_{i=1}^n count_i$$

und erhalten  $(A''|b') \in \{1, 0, -1\}^{m \times N} \times \mathbb{R}^m$ .

Nun fordern wir noch, dass für jedes  $l \in \{1, \dots, n\}$  zwei Variablen  $x_i$  und  $x_j$  mit

$$i, j \in \left\{ \sum_{u=1}^{l-1} count_u + 1, \dots, \sum_{u=1}^l count_u \right\}$$

stets denselben Wert annehmen.

Dafür fügen wir für alle  $l \in \{1, \dots, n\}$  der Matrix  $(A''|b')$  die Ungleichungen

$$x_p - x_i \leq 0$$

und

$$x_i - x_p \leq 0$$

mit

$$p := \sum_{u=1}^{l-1} count_u + 1$$

und

$$i = \sum_{u=1}^{l-1} count_u + 2, \dots, \sum_{u=1}^l count_u$$

hinzu.

Offensichtlich verändern diese hinzugefügten Ungleichungen dabei die erwünschte  $\{1, 0, -1\}$ -Struktur unserer Matrix  $A''$  nicht.

Die so entstehende Matrix  $(A'''|b'')$  erfüllt  $(A'''|b'') \in \{1, 0, -1\}^{M \times N} \times \mathbb{R}^M$  mit  $M = 2 \cdot (N - n) + m$ , da für jedes  $i \in \{1, \dots, n\}$  zu den  $m$  vorhandenen Nebenbedingungen  $(count_i - 1)$  Nebenbedingungs-Paare hinzugefügt werden.

Wir definieren abschließend

$$c^{tt} := \left( \underbrace{c_1, \mathbf{0}_{count_1-1}}_{count_1 \text{ Einträge}}, \dots, \underbrace{c_n, \mathbf{0}_{count_n-1}}_{count_n \text{ Einträge}} \right) \in \mathbb{R}^{1 \times N}.$$

Nun setzen wir  $A^* := A'''$ ,  $b^* := b''$  und  $c^* := c'$  und erhalten damit das lineare Programm

$$(GLP^*) \quad \begin{array}{ll} \min & c^{*t}x \\ \text{s.d.} & A^*x \leq b^* \\ & x \in \mathbb{Z}^N \end{array}$$

mit  $A^* \in \{1,0,-1\}^{M \times N}$ ,  $b^* \in \mathbb{R}^M$ ,  $c^* \in \mathbb{R}^N$  in  $\{1,0,-1\}$ -Form.

Es bleibt zu zeigen, dass  $(GLP)$  und  $(GLP^*)$  äquivalent sind.

Dafür wird gezeigt, dass für jedes  $y \in \{x \in \mathbb{R}^n : Ax \leq b\}$  mit Zielfunktionswert  $c^t y$  ein  $y' \in \{x \in \mathbb{R}^N : A^*x \leq b^*\}$  mit  $c^{*t} y' = c^t y$  existiert und umgekehrt.

Sei zunächst  $y \in \{x \in \mathbb{R}^n : Ax \leq b\}$ . Setze  $y' = (y_1 \cdot \mathbf{1}_{count_1}^t, \dots, y_n \cdot \mathbf{1}_{count_n}^t)^t \in \mathbb{R}^N$ . Aus der Konstruktion von  $(A^*|b^*)$  und  $c^*$  ergibt sich  $y' \in \{x \in \mathbb{R}^N : A^*x \leq b^*\}$  und  $c^{*t} y' = c^t y$ .

Sei nun  $y' \in \{x \in \mathbb{R}^N : A^*x \leq b^*\}$ . Setze

$$y = (y'_{count_1}, y'_{count_1+count_2}, \dots, y'_{\sum_{i=1}^n count_i})^t \in \mathbb{R}^n.$$

Wieder aus der Konstruktion von  $(A^*|b^*)$  und  $c^*$  ergibt sich  $y \in \{x \in \mathbb{R}^n : Ax \leq b\}$  und  $c^t y = c^{*t} y'$ .

Somit sind  $(GLP)$  und  $(GLP^*)$  äquivalent und es folgt die Behauptung.  $\square$

**Bemerkung 2.4.** Die Anzahl der Variablen sowie die der Nebenbedingungen in  $(GLP^*)$  ist in den meisten Fällen sehr viel größer als in  $(GLP)$ . Sie hängt von den Einträgen der Matrix  $A$  ab.

Sind diese betragsmäßig sehr groß, erhält man sehr viele neue Dimensionen und davon abhängig linear viele neue Nebenbedingungen. Es gelten dabei

$$N = \sum_{i=1}^n count_i$$

und

$$M = 2 \cdot (N - n) + m.$$

Das Programm  $(GLP^*)$  besitzt somit mehr Dimensionen und mehr Nebenbedingungen als  $(GLP)$ , jedoch hat die beschreibende Matrix in  $(GLP^*)$  eine einfachere Struktur (siehe Definition 2.1), die für das vorgestellte Cutting-Verfahren benötigt wird.

## 2.2 Das Verfahren der Binärkodierung

Wie auch schon in Beispiel 2.3 zu sehen ist, steigt die Anzahl der Variablen (und damit die der Nebenbedingungen) enorm an, falls die Einträge der Matrix  $A$  betragsmäßig groß sind.

Deswegen sei hier noch ein zweites Verfahren angesprochen, welches im Folgenden das Verfahren der Binärkodierung genannt wird. Dieses bringt das ( $GLP$ ) ebenfalls in  $\{1,0,-1\}$ -Form, benötigt dabei aber wesentlich weniger neue Variablen.

Die Grundidee<sup>1</sup> dabei ist, die Einträge von  $A$  in Binärschreibweise zu bringen: So lässt sich zum Beispiel die Zahl 13 folgendermaßen darstellen:

$$13 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = (1101)_2$$

Die  $\{1,0\}$ -Struktur der Binärdarstellung (bzw.  $\{-1,0\}$ -Struktur falls negative Zahlen dargestellt werden sollen) motiviert nun folgenden Ansatz:

Anstatt  $13x$  als

$$(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1) \cdot \mathbf{1}_{13} \cdot x$$

in  $\{1,0,-1\}$ -Form zu bringen, schreiben wir

$$\begin{aligned} 13x &= (1101)_2 \cdot x \\ &= 1 \cdot (2^3x) + 1 \cdot (2^2x) + 0 \cdot (2^1x) + 1 \cdot (2^0x) \end{aligned}$$

Allgemein gehen wir also wie folgt vor:

Wir ersetzen die Variable  $x_i$  durch die Variablen

$$\begin{aligned} x_{i0} &:= x_i \\ x_{i1} &:= 2 \cdot x_i \\ x_{i2} &:= 2^2 \cdot x_i \\ &\vdots \\ x_{i[\log_2(\text{count}_i)]} &:= 2^{[\log_2(\text{count}_i)]} \cdot x_i. \end{aligned}$$

<sup>1</sup>Die Idee, die Einträge von  $A$  binär zu kodieren, entstand in einem persönlichen Gespräch mit Michael Siebert, für die Inspiration dazu sei ihm an dieser Stelle noch einmal herzlich gedankt.

Diese lassen sich einfach rekursiv durch die linearen Gleichungen

$$\begin{aligned} x_{i[\log_2(\text{count}_i)]} &= 2 \cdot x_{i([\log_2(\text{count}_i)]-1)} \\ x_{i([\log_2(\text{count}_i)]-1)} &= 2 \cdot x_{i([\log_2(\text{count}_i)]-2)} \\ &\vdots \\ x_{i1} &= 2 \cdot x_{i0} \end{aligned}$$

darstellen.

Weiter schreiben wir die  $a_{ij}$  in ihrer Binärdarstellung (mit  $\lfloor \log_2(\text{count}_i) \rfloor + 1$  Ziffern).

So erhalten wir zusammen mit den Nebenbedingungen an die neuen Variablen eine Matrix  $A' \in \{2, 1, 0, -1, -2\}^{M' \times N'}$  mit  $M', N' \in \mathbb{N}$  (auf  $M'$  und  $N'$  wird später noch genauer eingegangen).

Dabei ist ein Eintrag von  $A'$  höchstens dann gleich  $\pm 2$ , wenn er in den neu hinzugefügten Gleichungen auftaucht.

Um die Zielfunktionswerte nicht zu verändern setzen wir

$$c^t := \underbrace{\left( \mathbf{0}_{\lfloor \log_2(\text{count}_1) \rfloor}^t, c_1, \dots \right)}_{\substack{\lfloor \log_2(\text{count}_1) \rfloor + 1 \\ \text{Einträge}}}, \dots, \underbrace{\left( \mathbf{0}_{\lfloor \log_2(\text{count}_n) \rfloor}^t, c_n \right)}_{\substack{\lfloor \log_2(\text{count}_n) \rfloor + 1 \\ \text{Einträge}}}.$$

Das so entstandene Programm

$$\begin{aligned} \min \quad & c^t x \\ \text{s.d.} \quad & A' x \leq b \\ & x \in \mathbb{Z}^{N'} \end{aligned}$$

bringen wir nun mit dem im Beweis von Satz 2.2 gegebenen Verfahren auf  $\{1,0,-1\}$ -Form.

**Beispiel 2.5.** Wir formen Beispiel 2.2 nach dem Verfahren der Binärcodierung um:

Für

$$A = \begin{pmatrix} 9 & 0 \\ -5 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} \frac{3}{2} \\ 3 \end{pmatrix}$$

kodieren wir die einzelnen Einträge binär:

Da

$$\begin{aligned} (9)_{10} &= (1001)_2 \\ -(5)_{10} &= -(101)_2 \\ (0)_{10} &= (0)_2 \\ (1)_{10} &= (1)_2, \end{aligned}$$

kodieren wir jeden Eintrag der ersten Spalte vierstellig und jeden Eintrag der zweiten Spalte einstellig.



Auf diese Art entsteht die Matrix

$$A' = \left( \begin{array}{cccc|c} 1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & -1 & 1 \end{array} \right).$$

Mit

$$\begin{aligned} c'^t &:= (\mathbf{0}_{[\log_2(\text{count}_1)]}^t, c_1, \mathbf{0}_{[\log_2(\text{count}_2)]}^t, c_2) \\ &= (\mathbf{0}_3^t, c_1, \mathbf{0}_0^t, c_2) \\ &= (0, 0, 0, 1, -5) \end{aligned}$$

ergibt sich das lineare Programm

$$\begin{aligned} (GLP1^*) \quad &\min \quad (0, 0, 0, 1, -5) \cdot x \\ &s.d. \quad A'x \leq b \\ &\quad \quad x \in \mathbb{Z}^5. \end{aligned}$$

Nun müssen wir noch die Binärkodierung (also das richtige Verhältnis der neuen Variablen) in (GLP') fordern.

Somit soll für jedes zulässige  $x \in \mathbb{Z}^5$

$$x_1 = 2 \cdot x_2$$

$$x_2 = 2 \cdot x_3$$

$$x_3 = 2 \cdot x_4$$

gelten.

Jede dieser Gleichheitsbedingungen wird durch zwei Ungleichungen modelliert und wir erhalten

$$A'' = \left( \begin{array}{ccccc} 1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & -1 & 1 \\ \hline 1 & -2 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 & 0 \\ 0 & 1 & -2 & 0 & 0 \\ 0 & -1 & 2 & 0 & 0 \\ 0 & 0 & 1 & -2 & 0 \\ 0 & 0 & -1 & 2 & 0 \end{array} \right) \quad \text{und } b' := \begin{pmatrix} \frac{3}{2} \\ 3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Diese Matrix bringen wir nun mit dem Verfahren aus Satz 2.2 auf  $\{1, 0, -1\}$ -Form.

So erhalten wir

$$A''' = \left( \begin{array}{cccccc|c} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 & 0 & 1 \\ \hline 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 1 & 0 \\ \hline 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \end{array} \right) \quad \text{und } b'' = \begin{pmatrix} 3 \\ 2 \\ 3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

mit  $A''' \in \{1,0,-1\}^{14 \times 8}$  und  $b'' \in \mathbb{R}^{14}$ .

Insgesamt erhalten wir also durch das Verfahren der Binärkodierung ein zu (GLP1) äquivalentes Programm in  $\{1,0,-1\}$ -Form mit 8 Variablen und 14 Nebenbedingungen. Wie in Beispiel 2.3 zu sehen ist, benötigt man im additiven Verfahren 10 Variablen und 18 Nebenbedingungen.

Betrachten wir die allgemeine Anzahl an benötigten Variablen und Nebenbedingungen im Verfahren der Binärkodierung, so erkennen wir folgende Aussagen:

**Korollar 2.6.** Für das Verfahren der Binärkodierung gelten:

$$N^* = 2 \cdot \left( \sum_{i=1}^n \lfloor \log_2(\text{count}_i) \rfloor \right) + n$$

$$M^* = 2 \cdot (N^* - n) + m$$

*Beweis.* Wie man im Verfahren erkennt, wird im ersten Schritt des Verfahrens jede 'alte' Variable  $x_i$  durch die 'neuen' Variablen  $x_{i0}, \dots, x_{i_{\lfloor \log_2(\text{count}_i) \rfloor}}$  modelliert.

Somit erhält man ganz offensichtlich  $\lfloor \log_2(\text{count}_i) \rfloor + 1$  'neue' Variablen, also insgesamt

$$\sum_{i=1}^n \lfloor \log_2(\text{count}_i) \rfloor$$

zusätzliche Variablen.

Im zweiten Schritt des Verfahrens wird nun jede dieser zusätzlichen Variablen durch zwei 'ganz neue' Variablen modelliert, man erhält also noch

$$\sum_{i=1}^n \lfloor \log_2(\text{count}_i) \rfloor$$

Variablen dazu.

In beiden Schritten zusammen erhält man also insgesamt

$$2 \cdot \sum_{i=1}^n \lfloor \log_2(\text{count}_i) \rfloor$$

zusätzliche Variablen.

Es ergibt sich die Anzahl der neuen Variablen  $N^*$  als Summe aus den zusätzlichen und den bereits vorhandenen Variablen:

$$N^* = 2 \cdot \sum_{i=1}^n \lfloor \log_2(\text{count}_i) \rfloor + n$$

Die Anzahl der zusätzlichen Nebenbedingungen ergibt sich nun wieder aus beiden Schritten zusammen:

Im ersten Schritt erhält man

$$2 \cdot \sum_{i=1}^n \lfloor \log_2(\text{count}_i) \rfloor$$

zusätzliche Nebenbedingungen, da für jede zusätzliche Variable ein Paar von Nebenbedingungen zur Matrix hinzugefügt wird, das die Binärkodierung sichert.

Im zweiten Schritt wird jede neu hinzugefügte Variable durch zwei Variablen und zwei zugehörige Nebenbedingungen modelliert, man erhält also wieder

$$2 \cdot \sum_{i=1}^n \lfloor \log_2(\text{count}_i) \rfloor$$

zusätzliche Nebenbedingungen.

Somit werden

$$2 \cdot (2 \cdot \sum_{i=1}^n \lfloor \log_2(\text{count}_i) \rfloor) = 2 \cdot (N^* - n)$$

Nebenbedingungen hinzugefügt.

Insgesamt ergibt sich also

$$M^* = 2 \cdot (N^* - n) + m$$

und damit die Behauptung. □

Es ist nun leicht ersichtlich, dass das Verfahren der Binärkodierung für große Koeffizienten zu einem äquivalenten Programm mit wesentlich weniger Variablen und Nebenbedingungen führt, als das additive Verfahren.

Das nachfolgende Beispiel soll diesen Unterschied noch einmal sehr deutlich zeigen:

**Beispiel 2.7.** Wir betrachten für  $c, b \in \mathbb{R}^3$  das lineare Programm

$$\begin{aligned}
 (GLP') \quad & \min \quad c^t x \\
 \text{s.d.} \quad & \begin{pmatrix} 2^{32} & 1 & 15 \\ 12 & -1 & 7 \\ -64 & 0 & -36 \end{pmatrix} \cdot x \leq b \\
 & x \in \mathbb{Z}^5.
 \end{aligned}$$

Nach Bemerkung 2.4 und Korollar 2.6 gelten:

$$\begin{aligned}
 N &= 2^{32} + 1 + 36 && \approx 4.3 \cdot 10^9 \\
 M &= 2 \cdot (N - 3) + 3 && \approx 4.3 \cdot 10^9 \\
 N^* &= 2 \cdot (32 + 0 + 5) + 3 &= 77 \\
 M^* &= 2 \cdot (77 - 3) + 3 &= 151
 \end{aligned}$$

Ganz offensichtlich ist  $N^* = 77 \ll 4.3 \cdot 10^9 \approx N$  und  $M^* = 151 \ll 4.3 \cdot 10^9 \approx M$ .

Gerade bei sehr großen Einträgen von  $A$  ist das Verfahren der Binärcodierung zur Umwandlung eines  $(GLP)$  mit rationaler Matrix  $A$  in  $\{1,0,-1\}$ -Form also wesentlich effizienter als das additive Verfahren.

## 3 Cutting-Verfahren

In diesem Kapitel werden nun Cutting-Verfahren vorgestellt, die den relaxierten zulässigen Bereich eines (*GLP*) so einschränken, dass dieser sich der konvexen Hülle des zulässigen Bereichs annähert.

### 3.1 Das Abrundungs-Verfahren

Dieses Verfahren nutzt die leicht zu erkennende Tatsache aus, dass Produkte ganzzahliger Vektoren wieder ganzzahlig sind.

Wir erhalten ein leicht durchzuführendes Cutting-Verfahren:

**Satz 3.1.** (Nemhauser and Wolsey [5], Satz 1.2) *Seien  $A \in \{1, 0, -1\}^{m \times n}$  und  $b \in \mathbb{R}^m$ .*

*Dann gilt:*

$$X_1 := \{x \in \mathbb{Z}^n : Ax \leq b\} = \{x \in \mathbb{Z}^n : Ax \leq \lfloor b \rfloor\} =: X_2$$

*Beweis.* Ganz offensichtlich ist  $X_2 \subseteq X_1$ .

Sei  $x \in X_1$ . Dann gilt zunächst  $Ax \leq b$ .

Da aber  $x \in \mathbb{Z}^n$  und  $A \in \{1, 0, -1\}^{m \times n}$  gilt außerdem  $Ax \in \mathbb{Z}^m$ .

Zusammen folgt  $Ax \leq \lfloor b \rfloor$ , also  $x \in X_2$  und somit  $X_1 \subseteq X_2$ .

□

Das erste Cutting-Verfahren ist also die Abrundung des  $b$ -Vektors.

Da dieses Verfahren alleine jedoch sehr schnell getan ist und keiner Vorarbeit bedarf, betrachten wir ab jetzt nur noch (*GLP*) der Form

$$\begin{array}{ll} (\text{GLP}) & \min \quad c^t x \\ & \text{s.d.} \quad Ax \leq b \\ & \quad \quad x \in \mathbb{Z}^n \end{array}$$

mit  $A \in \{1, 0, -1\}^{m \times n}$ ,  $b \in \mathbb{Z}^m$ ,  $c \in \mathbb{R}^n$ .

Das zweite vorgestellte Cutting-Verfahren ist nun etwas komplexer und nutzt die  $\{1, 0, -1\}$ -Struktur des (*GLP*) aus.

## 3.2 Das 2-Additive Cutting

Wir betrachten in diesem Kapitel Paare von Nebenbedingungen  $(a_i|b_i)$  eines  $(GLP)$  in  $\{1,0,-1\}$ -Form.

Wir betrachten aber nur solche Paare, die linear unabhängig in den  $a_i$  sind, denn ansonsten sind die beiden  $H_i$  parallel und somit sind diese entweder identisch oder besitzen einen leeren Schnitt.

Betrachten wir nun für Ungleichungen  $(a_i|b_i)$  (linear unabhängig in  $a_i$ ) den Schnitt der beiden Hyperebenen  $H_i$ , so erkennen wir, dass dieser (geschnitten mit dem relaxierten zulässigen Bereich  $P$  des  $(GLP)$ ) eine Extremalmenge von  $P$  darstellt. Enthält der Schnitt von  $P$  und den  $H_i$  keinen (für das  $(GLP)$  zulässigen) ganzzahligen Punkt, soll diese Extremalmenge vom zulässigen Polyeder abgeschnitten werden.

Da es in den meisten Fällen aber sehr schwer ist, zu entscheiden, ob eine Menge zulässig für ein  $(GLP)$  ist (also in  $P$  liegt), betrachten wir eine andere hinreichende Bedingung:

Wir stellen uns die Frage, wann der Schnitt zweier solcher Hyperebenen  $H_i$  gar keinen ganzzahligen Punkt enthält:

**Satz 3.2.** *Es seien  $(a_1|b_1)$  und  $(a_2|b_2)$  zwei in  $a_1$  und  $a_2$  linear unabhängige Nebenbedingungen eines  $(GLP)$  in  $\{1,0,-1\}$ -Form. Dann gilt*

$$\{x \in \mathbb{Z}^n : a_1x = b_1\} \cap \{x \in \mathbb{Z}^n : a_2x = b_2\} = \emptyset$$

genau dann, wenn die beiden folgenden Bedingungen erfüllt sind:

$$a_{1i} = 0 \Leftrightarrow a_{2i} = 0 \quad \forall i \in \{1, \dots, n\} \tag{3.1}$$

$$b_1 \not\equiv b_2 \pmod{2} \tag{3.2}$$

*Beweis.* Wir zeigen zunächst “ $\Rightarrow$ “:

Dazu nehmen wir an, dass eine der Bedingungen (1.) und (2.) nicht erfüllt ist und zeigen, dass dann

$$\{x \in \mathbb{Z}^n : a_1x = b_1\} \cap \{x \in \mathbb{Z}^n : a_2x = b_2\} \neq \emptyset$$

gilt.

1. Fall: Sei zunächst Bedingung (3.1) nicht erfüllt, es gelte also für ein  $i \in \{1, \dots, n\}$

$$((a_{1i} = 0) \wedge (a_{2i} \neq 0)) \text{ oder } ((a_{1i} \neq 0) \wedge (a_{2i} = 0)).$$

O.B.d.A. können wir  $((a_{1i} = 0) \wedge (a_{2i} \neq 0))$  annehmen.

Dann existiert ein  $j \in \{1, \dots, n\} \setminus \{i\}$  mit  $a_{1j} \neq 0$ , denn es gilt  $a_1 \neq \mathbf{0}_n^t$ , da  $a_1$  und  $a_2$  linear unabhängig sind.

Wir setzen für  $l = 1, \dots, n$ :

$$x'_l := \begin{cases} -\operatorname{sgn}(a_{1j}) \cdot (a_{11} + \dots + a_{1n} - b_1 - a_{1j}) & , \text{ falls } l = j \\ -\operatorname{sgn}(a_{2i}) \cdot (a_{21} + \dots + a_{2n} - b_2 - a_{2i} - a_{2j} + a_{2j} \cdot x'_j) & , \text{ falls } l = i \\ 1 & \text{sonst} \end{cases}$$

Es gilt  $x' \in \mathbb{Z}^n$ , da  $a_1^t, a_2^t \in \{1, 0, -1\}^n$  und  $b_i \in \mathbb{Z}$ .

Weiter ist

$$\begin{aligned} a_1 x' &= a_{11} x'_1 + a_{12} x'_2 + \dots + a_{1n} x'_n \\ &\stackrel{a_{1i} \equiv 0}{=} (a_{11} + \dots + a_{1n} - a_{1j}) \cdot 1 \\ &\quad + a_{1j} \cdot (-\operatorname{sgn}(a_{1j}) \cdot (a_{11} + \dots + a_{1n} - b_1 - a_{1j})) \\ &\stackrel{a_{1j} \equiv \pm 1}{=} (a_{11} + \dots + a_{1n} - a_{1j}) - (a_{11} + \dots + a_{1n} - b_1 - a_{1j}) \\ &= b_1 \end{aligned}$$

und somit  $x' \in \{x \in \mathbb{Z}^n : a_1 x = b_1\}$ .

Ebenso gilt

$$\begin{aligned} a_2 x' &= a_{21} x'_1 + a_{22} x'_2 + \dots + a_{2n} x'_n \\ &= (a_{21} + \dots + a_{2n} - a_{2j} - a_{2i}) \cdot 1 + a_{2j} \cdot x'_j \\ &\quad + a_{2i} \cdot (-\operatorname{sgn}(a_{2i}) \cdot (a_{21} + \dots + a_{2n} - b_2 - a_{2i} - a_{2j} + a_{2j} \cdot x'_j)) \\ &\stackrel{a_{2i} \equiv \pm 1}{=} (a_{21} + \dots + a_{2n} - a_{2j} - a_{2i}) + a_{2j} \cdot x'_j \\ &\quad - (a_{21} + \dots + a_{2n} - b_2 - a_{2i} - a_{2j} + a_{2j} \cdot x'_j) \\ &= b_2 \end{aligned}$$

und deshalb  $x' \in \{x \in \mathbb{Z}^n : a_2 x = b_2\}$ .

Es folgt  $\{x \in \mathbb{Z}^n : a_1 x = b_1\} \cap \{x \in \mathbb{Z}^n : a_2 x = b_2\} \supset \{x'\} \neq \emptyset$ .

2. Fall: Sei nun Bedingung (3.1) erfüllt, Bedingung (3.2) jedoch nicht.

Es gelte also für alle  $i \in \{1, \dots, n\}$ :

$$(a_{1i} = 0 \Leftrightarrow a_{2i} = 0) \text{ aber } b_1 \equiv b_2 \pmod{2}$$

Seien nun  $i$  und  $j$  so gewählt, dass  $0 \neq a_{1i} = a_{2i}$  und  $0 \neq a_{1j} = -a_{2j}$ .

Diese  $i$  und  $j$  existieren wegen der  $\{1, 0, -1\}$ -Form von  $A$ , denn ansonsten wären  $a_1$  und  $a_2$  linear abhängig.

Wir setzen:

$$x_l'' := \begin{cases} \operatorname{sgn}(a_{1i}) \cdot \frac{b_1+b_2}{2} & , \text{ falls } l = i \\ \operatorname{sgn}(a_{1j}) \cdot (b_1 - \frac{b_1+b_2}{2}) & , \text{ falls } l = j \\ 0 & \text{sonst} \end{cases}$$

Es gilt  $\frac{b_1+b_2}{2} \in \mathbb{Z}$ , da  $b_1 \equiv b_2 \pmod{2}$  und somit folgt  $x'' \in \mathbb{Z}^n$ .

Außerdem ist

$$\begin{aligned} a_1 x'' &= a_{11} x_1'' + a_{12} x_2'' + \cdots + a_{1n} x_n'' \\ &= a_{1i} \cdot (\operatorname{sgn}(a_{1i}) \cdot \frac{b_1+b_2}{2}) \\ &\quad + a_{1j} \cdot (\operatorname{sgn}(a_{1j}) \cdot (b_1 - \frac{b_1+b_2}{2})) \\ &\stackrel{a_{1i}, a_{1j} = \pm 1}{=} \frac{b_1+b_2}{2} + (b_1 - \frac{b_1+b_2}{2}) \\ &= b_1 \end{aligned}$$

und somit  $x'' \in \{x \in \mathbb{Z}^n : a_1 x = b_1\}$ .

Ebenso gilt

$$\begin{aligned} a_2 x'' &= a_{21} x_1'' + a_{22} x_2'' + \cdots + a_{2n} x_n'' \\ &= a_{2i} \cdot (\operatorname{sgn}(a_{1i}) \cdot \frac{b_1+b_2}{2}) \\ &\quad + a_{2j} \cdot (\operatorname{sgn}(a_{1j}) \cdot (b_1 - \frac{b_1+b_2}{2})) \\ &\stackrel{a_{1i} = a_{2i}, a_{1j} = -a_{2j}}{=} a_{1i} \cdot (\operatorname{sgn}(a_{1i}) \cdot \frac{b_1+b_2}{2}) \\ &\quad - a_{1j} \cdot (\operatorname{sgn}(a_{1j}) \cdot (b_1 - \frac{b_1+b_2}{2})) \\ &\stackrel{a_{1i}, a_{1j} = \pm 1}{=} \frac{b_1+b_2}{2} - (b_1 - \frac{b_1+b_2}{2}) \\ &= b_2 \end{aligned}$$

und deshalb  $x'' \in \{x \in \mathbb{Z}^n : a_2 x = b_2\}$ .

Daraus erkennen wir wie auch in Fall 1

$$\{x \in \mathbb{Z}^n : a_1 x = b_1\} \cap \{x \in \mathbb{Z}^n : a_2 x = b_2\} \supset \{x'\} \neq \emptyset,$$

und somit haben wir den ersten Teil der Behauptung bewiesen.

Weiter zeigen wir nun " $\Leftarrow$ ":

Es gelte also für alle  $i \in \{1, \dots, n\}$ :

$$(a_{1i} = 0 \Leftrightarrow a_{2i} = 0) \text{ und } b_1 \not\equiv b_2 \pmod{2}$$



Dann ist zunächst, da das (*GLP*) in  $\{1,0,-1\}$ -Form ist,  $(a_1 - a_2)^t \in \{2, 0, -2\}^n$  und somit

$$\left(\frac{a_1 - a_2}{2}\right)^t \in \mathbb{Z}^n.$$

Weiter folgt aus  $b_1 \not\equiv b_2 \pmod{2}$

$$\frac{b_1 - b_2}{2} \notin \mathbb{Z}.$$

Nun gilt aber

$$\begin{aligned} \{x \in \mathbb{Z}^n : a_1 x = b_1\} \cap \{x \in \mathbb{Z}^n : a_2 x = b_2\} &\subset \{x \in \mathbb{Z}^n : a_1 x - b_1 = a_2 x - b_2\} \\ &= \{x \in \mathbb{Z}^n : \frac{(a_1 - a_2)}{2} x = \frac{b_1 - b_2}{2}\}. \end{aligned}$$

Da  $\frac{b_1 - b_2}{2} \notin \mathbb{Z}$ , andererseits aber  $\left(\frac{a_1 - a_2}{2}\right)^t \in \mathbb{Z}^n$  und somit

$$\frac{a_1 - a_2}{2} \cdot x \in \mathbb{Z}$$

für  $x \in \mathbb{Z}^n$ , folgt nun

$$\left\{x \in \mathbb{Z}^n : \frac{a_1 - a_2}{2} \cdot x = \frac{b_1 - b_2}{2}\right\} = \emptyset$$

und somit

$$\{x \in \mathbb{Z}^n : a_1 x = b_1\} \cap \{x \in \mathbb{Z}^n : a_2 x = b_2\} = \emptyset.$$

Dies ist der zweite Teil der Behauptung. □

Satz 3.2 liefert also ein notwendiges und hinreichendes Kriterium um festzustellen, wann der Schnitt zweier Hyperebenen  $H_1$  und  $H_2$  ganzzahlige Punkte enthält und wann nicht.

Im zweiten Fall wollen wir den Schnitt dieser beiden Gleichungen für das (*GLP*) verbieten.

Dies wird durch das später vorgestellte Cutting-Verfahren geschehen, für welches der folgende Hilfssatz die Grundlage liefert:

**Lemma 3.3.** *Es seien  $(a_1|b_1)$  und  $(a_2|b_2)$  zwei in  $a_1$  und  $a_2$  linear unabhängige Ungleichungen eines (GLP), die die Bedingungen*

$$a_{1i} = 0 \Leftrightarrow a_{2i} = 0 \quad \forall i \in \{1, \dots, n\} \quad (3.1)$$

und

$$b_1 \not\equiv b_2 \pmod{2} \quad (3.2)$$

erfüllen.

Dann gelten für  $a_{neu} := \frac{a_1+a_2}{2}$  und  $b_{neu} := \lfloor \frac{b_1+b_2}{2} \rfloor$ :

1. Es ist  $a_{neu} \in \{1, 0, -1\}^n$ .
2. Die Anzahl der Nicht-Null-Einträge in  $a_{neu}$  ist echt kleiner als die der Nicht-Null-Einträge in  $a_1$  bzw.  $a_2$ .
3. Die Ungleichung  $(a_{neu}|b_{neu})$  ist eine gültige Ungleichung für das (GLP).

*Beweis.* zu 1.: Aus Bedingung (3.1) und der  $\{1,0,-1\}$ -Struktur von  $a_1$  und  $a_2$  folgt, dass

$$(a_1 + a_2) \in \{2, 0, -2\}^n$$

und somit

$$\frac{a_1 + a_2}{2} \in \{1, 0, -1\}^n.$$

zu 2.: Ebenfalls aus (3.1) folgt

$$\begin{aligned} & (a_{1i} = 0 \vee a_{2i} = 0) \\ \Rightarrow & (a_{1i} = 0 \wedge a_{2i} = 0) \\ \Rightarrow & a_{neu\ i} = 0. \end{aligned}$$

Es übertragen sich also bei der Addition alle bereits in  $a_1$  oder  $a_2$  vorhandenen Null-Einträge nach  $a_{neu}$ .

Nun sind  $a_1$  und  $a_2$  linear unabhängig.

Daraus folgt, dass es mindestens ein  $i \in \{1, \dots, n\}$  so gibt, dass  $a_{1i} \neq a_{2i}$ .

Da  $a_1$  und  $a_2$  (3.1) erfüllen, ergibt sich, dass  $a_{1i} \neq 0$  und  $a_{2i} \neq 0$ .

Weiter gilt  $a_1^t, a_2^t \in \{1, 0, -1\}^n$  und somit  $a_{1i} = \pm 1$  und  $a_{2i} = \mp 1$ , also insbesondere  $a_{neu\ i} = a_{1i} + a_{2i} = 0$ .

Mindestens eine (nämlich die  $i$ -te) Komponente wird also bei der Addition Null.

Da alle Null-Einträge erhalten bleiben und mindestens ein weiterer Null-Eintrag entsteht, folgt Behauptung 2.

zu 3.: Wir zeigen, dass jeder zulässige Punkt des (GLP) auch die Ungleichung  $(a_{neu}|b_{neu})$  erfüllt:

Sei  $y$  ein zulässiger Punkt des (GLP), gelte also

$$y \in \{x \in \mathbb{Z}^n : Ax \leq b\}.$$

Dann gilt insbesondere  $a_1 y \leq b_1$  und  $a_2 y \leq b_2$ , also auch

$$(a_1 + a_2) \cdot y \leq b_1 + b_2$$

und somit

$$\frac{a_1 + a_2}{2} \cdot y \leq \frac{b_1 + b_2}{2}.$$

Nach 1. folgt nun aber  $(\frac{a_1+a_2}{2})^t \in \{1, 0, -1\}^n$  und damit

$$\underbrace{\frac{a_1 + a_2}{2}}_{\in \{1, 0, -1\}^{1 \times n}} \cdot \underbrace{y}_{\in \mathbb{Z}^n} \in \mathbb{Z}.$$

Folglich gilt auch

$$\frac{a_1 + a_2}{2} \cdot y \leq \left\lfloor \frac{b_1 + b_2}{2} \right\rfloor,$$

also

$$y \in \left\{ x \in \mathbb{Z}^n : \frac{a_1 + a_2}{2} \cdot x \leq \left\lfloor \frac{b_1 + b_2}{2} \right\rfloor \right\}.$$

Es folgt die Behauptung. □

**Bemerkung 3.4.** In Lemma 3.3 gilt außerdem wegen Bedingung (3.2) auch

$$\frac{b_1 + b_2}{2} \notin \mathbb{Z}.$$

Deshalb können wir feststellen, dass sich der zulässige Bereich des (GLP) genau dann durch Hinzufügen der Ungleichung  $(a_{neu}|b_{neu})$  verkleinert, wenn

$$\{x \in \mathbb{R}^n : Ax \leq b\} \cap \left\{ x \in \mathbb{R}^n : \frac{a_1 + a_2}{2} \cdot x > \left\lfloor \frac{b_1 + b_2}{2} \right\rfloor \right\} \neq \emptyset, \quad (3.3)$$

denn genau dieser Schnitt wird durch Hinzufügen der Ungleichung verboten.

Wir geben nun noch die Definition zweier Mengen an, welche wir im Cutting-Verfahren benötigen werden:

**Definition 3.5.** Seien ein (GLP) und ein  $k \in \mathbb{N}$  gegeben. Wir definieren für  $v = 1, 2$  die Mengen

$$M_v^k = \{j \in \{1, \dots, m\} : a_j \text{ enthält genau } k \text{ Nicht-Null-Einträge, } b_j \equiv v \pmod{2}\}.$$

Wir können nun folgendes Cutting-Verfahren für ein (GLP) in  $\{1, 0, -1\}$ -Form einführen:

**Algorithmus 3.6. (2-Additives Cutting)**

- *Schritt 0:*  
setze  $b := \lfloor b \rfloor$
- *Schritt 1:*  
setze  $k = n$
- *Schritt 2:*  
füge für je zwei Ungleichungen  $(a_{j_1} | b_{j_1})$  und  $(a_{j_2} | b_{j_2})$  mit

$$j_1 \in M_1^k \text{ und } j_2 \in M_2^k,$$

die linear unabhängig in  $a_{j_1}$  und  $a_{j_2}$  sind und für die

$$(a_{j_1 i} = 0 \Leftrightarrow a_{j_2 i} = 0)$$

gilt, die Ungleichung  $\left( \frac{a_{j_1} + a_{j_2}}{2} \mid \left\lfloor \frac{b_{j_1} + b_{j_2}}{2} \right\rfloor \right)$  zu  $A$  hinzu

- *Schritt 3:*  
setze  $k = k - 1$ , falls nun  $k = 1$ : STOP  
sonst gehe zu 2

**Bemerkung 3.7.** In Algorithmus 3.6 brauchen wir den Fall  $k = 1$  nicht zu betrachten, da Paare von Ungleichungen mit genau einem Nicht-Null-Eintrag (und diesem in derselben Komponente) stets linear abhängig sind.

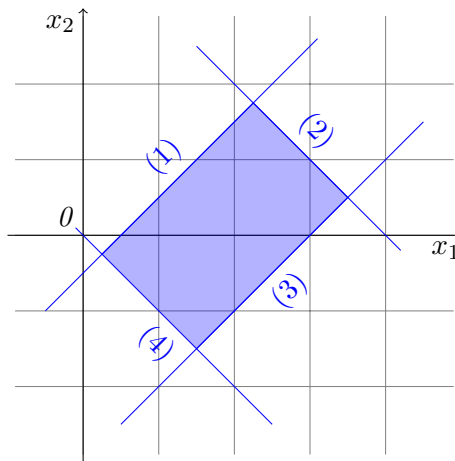
### 3.2.1 2-Additives Cutting im 2-dimensionalen Raum

Zunächst betrachten wir ein Beispiel für das 2-Additive Cutting im 2-dimensionalen Raum:

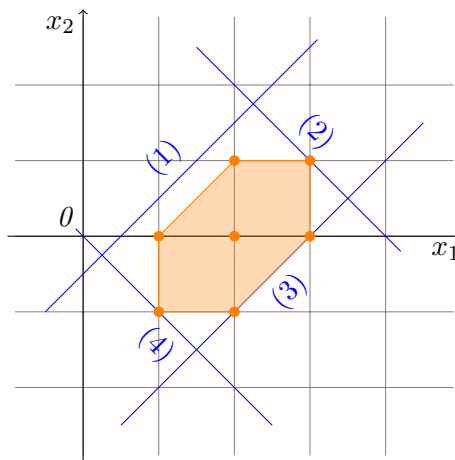
**Beispiel 3.8.** Gegeben sei das lineare Programm

$$\begin{aligned}
 (GLP2) \quad & \min \quad c^t x \\
 \text{s.d.} \quad & \begin{pmatrix} -1 & 1 \\ 1 & 1 \\ 1 & -1 \\ -1 & -1 \end{pmatrix} \cdot x \leq \begin{pmatrix} -0.5 \\ 4 \\ 3 \\ 0 \end{pmatrix} \quad \begin{matrix} (1) \\ (2) \\ (3) \\ (4) \end{matrix} \\
 & x \in \mathbb{Z}^2.
 \end{aligned}$$

Eine grafische Darstellung des relaxierten zulässigen Bereichs von (GLP2) ergibt:



Als konvexe Hülle des zulässigen Bereichs von (GLP2) erkennen wir also folgendes Polyeder:



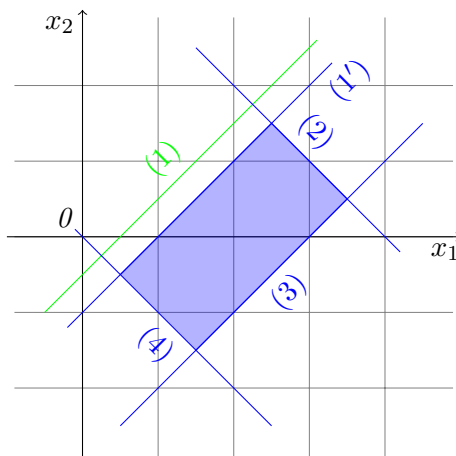
**Schritt 1:** Im ersten Schritt des Algorithmus setzen wir:

$$b' := \left[ \begin{array}{c} \left( \begin{array}{c} -0.5 \\ 4 \\ 3 \\ 0 \end{array} \right) \end{array} \right] = \begin{pmatrix} -1 \\ 4 \\ 3 \\ 0 \end{pmatrix}$$

Wir erhalten

$$\begin{aligned} (GLP2') \quad & \min \quad c^t x \\ \text{s.d.} \quad & \begin{pmatrix} -1 & 1 \\ 1 & 1 \\ 1 & -1 \\ -1 & -1 \end{pmatrix} \cdot x \leq \begin{pmatrix} -1 \\ 4 \\ 3 \\ 0 \end{pmatrix} \quad \begin{array}{l} (1') \\ (2) \\ (3) \\ (4) \end{array} \\ & x \in \mathbb{Z}^2 \end{aligned}$$

mit dem folgenden (verkleinerten) relaxierten zulässigen Bereich:



**Schritt 2:** Im zweiten Schritt suchen wir nun Paare von Ungleichungen, deren  $k = n = 2$  Nicht-Null-Einträge in denselben Komponenten stehen, die in ihren  $A$ -Anteilen linear unabhängig sind und deren  $b'$ -Anteile modulo 2 nicht in derselben Restklasse sind.

Betrachten wir die Matrix  $(A|b')$ , so sehen wir leicht, dass diese Bedingungen genau auf die Ungleichungs-Paare  $\{(1'), (2)\}$ ,  $\{(1'), (4)\}$ ,  $\{(2), (3)\}$  und  $\{(3), (4)\}$  zutreffen.

Nun fügen wir der Matrix  $(A|b')$  die Ungleichungen entsprechend Schritt 3 des Algorithmus 3.6 hinzu:

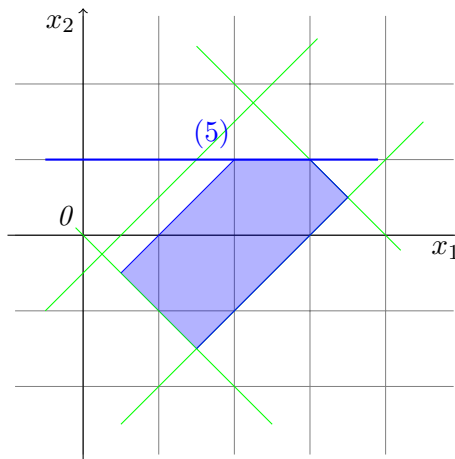
Wegen des Paares  $\{(1'), (2)\}$  fügen wir die Ungleichung

$$\left( \frac{(-1, 1) + (1, 1)}{2} \left\| \left\lfloor \frac{-1 + 4}{2} \right\rfloor \right. \right) = ((0, 1)|1)$$

hinzu und erhalten

$$\begin{aligned}
 (GLP2'') \quad & \min \quad c^t x \\
 \text{s.d.} \quad & \begin{pmatrix} -1 & 1 \\ 1 & 1 \\ 1 & -1 \\ -1 & -1 \\ 0 & 1 \end{pmatrix} \cdot x \leq \begin{pmatrix} -1 \\ 4 \\ 3 \\ 0 \\ 1 \end{pmatrix} \\
 & x \in \mathbb{Z}^2.
 \end{aligned} \tag{5}$$

Wie wir sehen, verkleinert sich der relaxierte zulässige Bereich weiter:



Wir fügen weiterhin wegen der Paare  $\{(1'), (4)\}$ ,  $\{(2), (3)\}$  und  $\{(3), (4)\}$  die folgenden Ungleichungen hinzu:

$$\left( \frac{(-1, 1) + (-1, -1)}{2} \left\| \left\lfloor \frac{-1+0}{2} \right\rfloor \right. \right) = ((-1, 0) | -1)$$

$$\left( \frac{(1, 1) + (1, -1)}{2} \left\| \left\lfloor \frac{4+3}{2} \right\rfloor \right. \right) = ((1, 0) | 3)$$

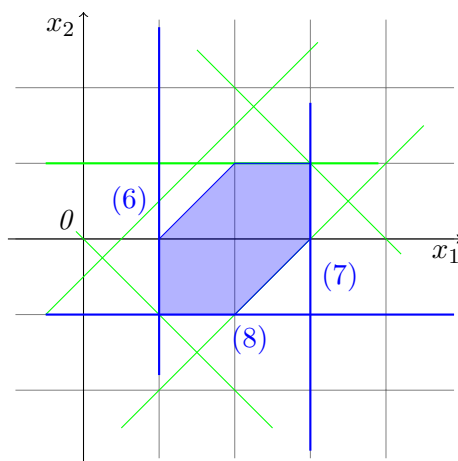
$$\left( \frac{(1, -1) + (-1, -1)}{2} \left\| \left\lfloor \frac{3+0}{2} \right\rfloor \right. \right) = ((0, -1) | 1)$$

Es ergibt sich:

$$\begin{aligned}
 & (GLP2''') \quad \min \quad c^t x \\
 & \text{s.d.} \quad \begin{pmatrix} -1 & 1 \\ 1 & 1 \\ 1 & -1 \\ -1 & -1 \\ 0 & 1 \\ -1 & 0 \\ 1 & 0 \\ 0 & -1 \end{pmatrix} \cdot x \leq \begin{pmatrix} -1 \\ 4 \\ 3 \\ 0 \\ 1 \\ -1 \\ 3 \\ 1 \end{pmatrix} \quad \begin{matrix} (6) \\ (7) \\ (8) \end{matrix} \\
 & \quad \quad \quad x \in \mathbb{Z}^2
 \end{aligned}$$

Nun bricht der Algorithmus ab, da der Fall  $k = 2$  vollständig behandelt wurde und die Fälle  $k = 1, 0$  nicht betrachtet werden.

Schauen wir uns nun den relaxierten zulässigen Bereich an, so ist dieser gerade die konvexe Hülle des zulässigen Bereichs von (GLP2):



Das 2-Additive Cutting liefert uns in diesem Beispiel also die konvexe Hülle des zulässigen Bereichs des betrachteten ganzzahligen linearen Programms.

Im 2-dimensionalen Raum lässt sich diese Beobachtung sogar zu folgendem Satz verallgemeinern:

**Satz 3.9.** *Wendet man das 2-Additive Cutting auf ein (GLP) mit zwei Variablen an, so ist nach Beendigung des Algorithmus die Menge*

$$P' := \{x \in \mathbb{R}^2 : A'x \leq b'\}$$

*gleich der konvexen Hülle des zulässigen Bereichs des (GLP).*



*Beweis.* Ist  $P'$  beschränkt, so reicht es aus, zu zeigen, dass jeder Extrempunkt von  $P'$  ganzzahlig ist.

Ist  $P'$  unbeschränkt, so muss außerdem gezeigt werden, dass jede unbeschränkte Facette  $F$  von  $P'$  unendlich viele ganzzahlige Punkte enthält.

Es ist aber  $F$  nach Definition 0.10 1-dimensional, also eine Gerade bzw. eine Halbgerade.

Weiter existiert nach Lemma 0.15 eine beschreibende Ungleichung  $(a_j|b_j)$  von  $F$ .

Außerdem existiert ein  $i \in \{1, 2\}$  so, dass  $a_{ji} \neq 0$ , da  $F$  sonst keine Facette wäre.

Wir können nun leicht sehen, dass  $x' \in \mathbb{Z}^2$  mit

$$x'_l := \begin{cases} 0 & , \text{ falls } l \neq i \\ a_{ji} \cdot b_i & , \text{ falls } l = i \end{cases}$$

wegen der  $\{1, 0, -1\}$ -Struktur von  $A$

$$a_j x'_j = a_{ji} \cdot x'_i = a_{ji} \cdot a_{ji} \cdot b_j \stackrel{a_{ji} = \pm 1}{=} b_j$$

erfüllt.

Somit enthält die Gerade  $\{x \in \mathbb{R}^2 : a_j x = b_j\}$  einen ganzzahligen Punkt und daher nach Lemma 0.14 auch unendlich viele ganzzahlige Punkte.

Ist  $F$  eine Gerade, also in beide Richtungen unbeschränkt, so gilt

$$F = \{x \in \mathbb{R}^2 : a_j x = b_j\}$$

und  $F$  enthält unendlich viele ganzzahlige Punkte.

Ist  $F$  eine Halbgerade, so ist  $F$  nur in eine Richtung beschränkt (von einem beliebigen Punkt auf  $F$  aus gesehen), da es nur zwei Richtungen gibt.

Es befinden sich aber von diesem beliebigen Punkt auf  $F$  aus gesehen in der nicht beschränkten Richtung ebenfalls unendlich viele ganzzahlige Punkte in  $F$ .

Damit ist  $F$  im unbeschränkten oder halb-beschränkten Fall stets eine Facette des zulässigen Bereichs des (GLP).

Nun zeigen wir noch, dass jeder Extrempunkt von  $P'$  ganzzahlig ist:

Sei  $y \in \mathbb{R}^2$  ein Extrempunkt von  $P'$ .

Nach Satz 0.16 existieren zwei Ungleichungen  $(a_1|b_1)$  und  $(a_2|b_2)$  (linear unabhängig in  $a_1$  und  $a_2$ ) aus  $(A|b)$  so, dass

$$\{y\} = \{x \in \mathbb{R}^n : a_1 x = b_1\} \cap \{x \in \mathbb{R}^n : a_2 x = b_2\}.$$

Es bleibt zu zeigen, dass  $y \in \mathbb{Z}^2$ .

Wir nehmen an  $y \notin \mathbb{Z}^2$ .

Dann folgen wegen der Struktur des (*GLP*) und nach Satz 3.2

$$a_{1i} = 0 \Leftrightarrow a_{2i} = 0 \quad (3.1)$$

und

$$b_1 \not\equiv b_2 \pmod{2} \quad (3.2)$$

Da nun aber Algorithmus 3.6 durchgeführt wurde, existiert nach Konstruktion des Algorithmus in den Nebenbedingungen des (*GLP*) auch die Ungleichung

$$\left( \frac{a_1 + a_2}{2} \left\| \left\lfloor \frac{b_1 + b_2}{2} \right\rfloor \right\| \right).$$

Nun ist wegen (3.1)

$$\left( \frac{a_1 + a_2}{2} \right)^t \in \mathbb{Z}^n$$

und wegen (3.2)

$$\frac{b_1 + b_2}{2} \notin \mathbb{Z}.$$

Somit ist der Punkt

$$\begin{aligned} y &\in \{x \in \mathbb{R}^n : a_1 x = b_1\} \cap \{x \in \mathbb{R}^n : a_2 x = b_2\} \\ &\subset \left\{ x \in \mathbb{R}^n : \frac{a_1 + a_2}{2} \cdot x = \frac{b_1 + b_2}{2} \right\} \end{aligned}$$

unzulässig für das (*GLP*).

Dies ist nun aber ein Widerspruch dazu, dass  $y$  ein Extrempunkt von  $P'$  ist. □

Das 2-Additive Cutting überführt ein 2-dimensionales (*GLP*) also in ein Problem, dessen relaxierter zulässiger Bereich die konvexe Hülle des zulässigen Bereichs des (*GLP*) ist.

### 3.2.2 2-Additives Cutting im 3-dimensionalen Raum

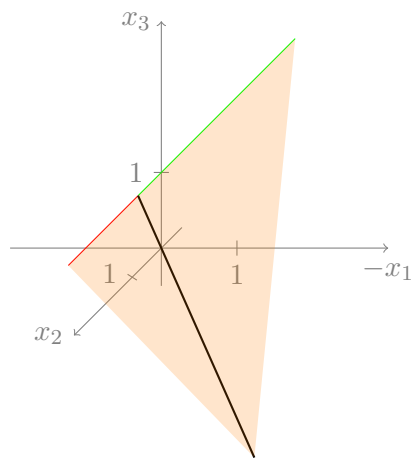
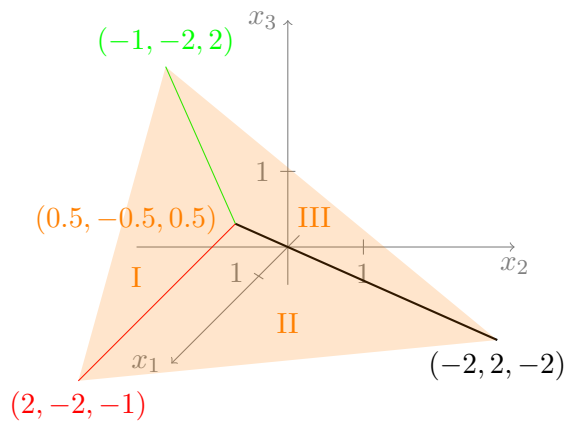
Wie im vorherigen Abschnitt gezeigt wurde, erhalten wir durch das 2-Additive Cutting im 2-dimensionalen Raum stets die konvexe Hülle des zulässigen Bereichs eines betrachteten (*GLP*).

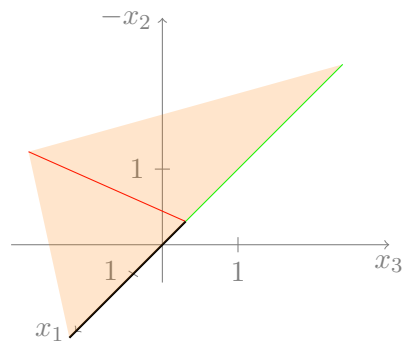
Leider erkennt man an folgendem Beispiel, dass dies in höheren Dimensionen zu-  
meist nicht gilt:

**Beispiel 3.10.** Wir betrachten das Problem

$$\begin{aligned}
 (GLP3) \quad & \min \quad c^t x \\
 \text{s.d.} \quad & \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \cdot x \leq \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \begin{matrix} \text{(I)} \\ \text{(II)} \\ \text{(III)} \end{matrix} \\
 & x \in \mathbb{Z}^3.
 \end{aligned}$$

Grafisch ergibt sich der relaxierte zulässige Bereich von (GLP3) folgendermaßen (der Anschaulichkeit halber aus drei Perspektiven):





Wir erkennen dabei, dass das Polyeder genau einen (nicht-ganzzahligen) Extrempunkt besitzt.

Algorithmus 3.6 liefert nun jedoch keinerlei zusätzliche Cuts, da schon Bedingung (3.1) für kein Paar von Ungleichungen erfüllt ist.

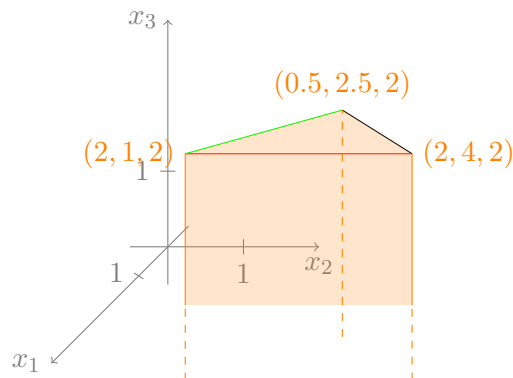
Somit findet der Algorithmus in höheren Dimensionen die konvexe Hülle des zulässigen Bereichs eines (GLP) zumeist nicht.

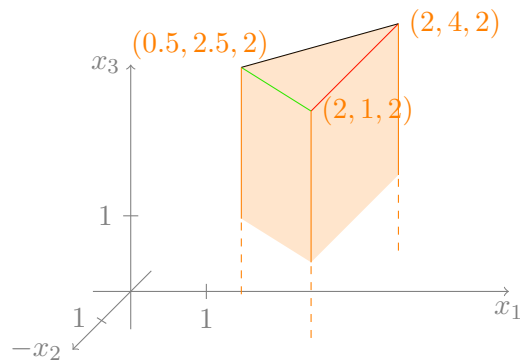
Allerdings kann man auch in diesen Dimensionen Beispiele finden, in denen das 2-Additive Cutting die konvexe Hülle des zulässigen Bereichs des (GLP) liefert.

**Beispiel 3.11.** Gegeben sei das lineare Programm

$$\begin{aligned}
 (GLP4) \quad & \min \quad c^t x \\
 \text{s.d.} \quad & \begin{pmatrix} 1 & 0 & 0 \\ -1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot x \leq \begin{pmatrix} 2 \\ -3 \\ 2 \\ 2 \end{pmatrix} \\
 & x \in \mathbb{Z}^3.
 \end{aligned}$$

Wir stellen den relaxierten zulässigen Bereich grafisch dar:





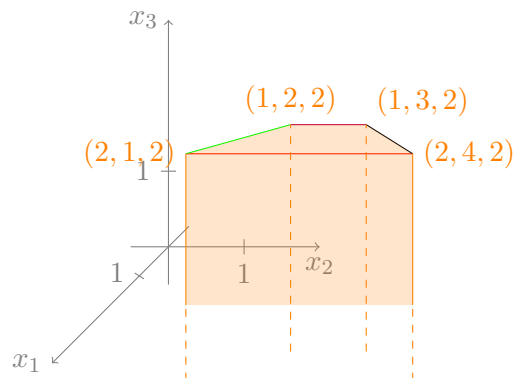
Wie wir sehen, ist nicht jeder Extrempunkt des relaxierten zulässigen Bereichs ganzzahlig. Bei der Durchführung des 2-Additiven Cuttings wird lediglich die Ungleichung

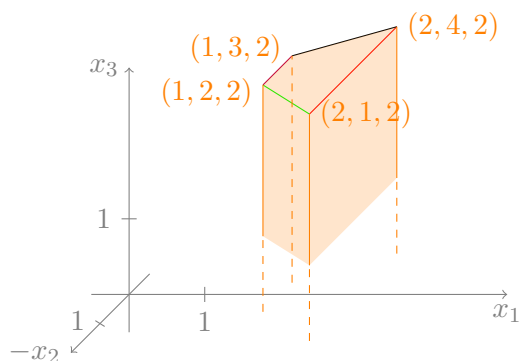
$$\left( \frac{(-1, -1, 0) + (-1, 1, 0)}{2} \mid \left\lfloor \frac{-3 + 2}{2} \right\rfloor \right) = ((-1, 0, 0) \mid -1)$$

hinzugefügt und wir erhalten das lineare Programm

$$\begin{aligned}
 (GLP4^*) \quad & \min \quad c^t x \\
 \text{s.d.} \quad & \begin{pmatrix} 1 & 0 & 0 \\ -1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{pmatrix} \cdot x \leq \begin{pmatrix} 2 \\ -3 \\ 2 \\ 2 \\ -1 \end{pmatrix} \\
 & x \in \mathbb{Z}^3
 \end{aligned}$$

mit relaxiertem zulässigem Bereich





Wie wir sehen, ist dieser die konvexe Hülle des zulässigen Bereichs von (GLP4).

Es ist sogar möglich, eine Spezialform von Matrizen anzugeben, für die das 2-Additive Cutting auch im 3-dimensionalen Raum stets die konvexe Hülle des beschriebenen Bereichs findet.

**Satz 3.12.** Sei ein (GLP) mit  $A \in \{1, 0, -1\}^{m \times 3}$ ,  $b \in \mathbb{Z}^m$  und es existiere ein  $i \in \{1, 2, 3\}$  so, dass gelte:

$$\forall j \in \{1, \dots, m\} : a_{ji} \neq 0 \Rightarrow a_{jk} = 0 \quad \forall k \neq i \tag{3.4}$$

Dann findet das 2-Additive Cutting die konvexe Hülle des zulässigen Bereichs des (GLP).

*Beweis.* Nach Umsortierung der Spalten können wir o.B.d.A.  $i = 3$  annehmen. Wir definieren  $J := \{j \in \{1, \dots, m\} : a_{j3} \neq 0\}$ .

(3.4) bedeutet nun, dass für jedes  $j \in J$  die Ungleichung  $(a_j|b_j)$  die Form

$$\pm x_3 \leq b_j$$

besitzt.

Es gilt somit

$$\begin{aligned}
\{x \in \mathbb{R}^3 : Ax \leq b\} &= \{x \in \mathbb{R}^3 : a_j x \leq b_j \forall j \in \{1, \dots, m\} \setminus J\} \\
&\quad \cap \{x \in \mathbb{R}^3 : a_j x \leq b_j \forall j \in J\} \\
&\stackrel{(3.4)}{=} \{x \in \mathbb{R}^3 : a_{j1}x_1 + a_{j2}x_2 \leq b_j \forall j \in \{1, \dots, m\} \setminus J\} \\
&\quad \cap \{x \in \mathbb{R}^3 : a_{j3}x_3 \leq b_j \forall j \in J\} \\
&= \{x \in \mathbb{R}^2 : a_{j1}x_1 + a_{j2}x_2 \leq b_j \forall j \in \{1, \dots, m\} \setminus J\} \times \mathbb{R} \\
&\quad \cap \mathbb{R}^2 \times \{x \in \mathbb{R} : a_{j3}x \leq b_j \forall j \in J\} \\
&= \{x \in \mathbb{R}^2 : a_{j1}x_1 + a_{j2}x_2 \leq b_j \forall j \in \{1, \dots, m\} \setminus J\} \\
&\quad \times \{x \in \mathbb{R} : a_{j3}x \leq b_j \forall j \in J\}.
\end{aligned}$$

Nach Durchführung des 2-Additiven Cuttings erhalten wir eine Matrix

$$(A'|b') \in \{1, 0, -1\}^{m' \times 3} \times \mathbb{Z}^{m'}.$$

Nun besitzt jede Ungleichung  $(a_j|b_j)$  mit  $a_{j3} \neq 0$  wegen (3.4) im  $a_j$ -Anteil genau eine Nicht-Null-Komponente (nämlich die dritte).

Diese Ungleichungen werden im 2-Additiven Cutting nach Konstruktion des Algorithmus demnach gar nicht betrachtet (siehe Schritt 3 des Algorithmus).

Jede neu hinzugefügte Ungleichung  $(a_{neu}|b_{neu})$  erfüllt also

$$a_{neu\ 3} = 0$$

und es gilt

$$J' := \{j \in \{1, \dots, m'\} : a'_{j3} \neq 0\} = J.$$

Wie oben folgt

$$\begin{aligned}
\{x \in \mathbb{R}^3 : A'x \leq b'\} &= \{x \in \mathbb{R}^2 : a'_{j1}x_1 + a'_{j2}x_2 \leq b'_j \forall j \in \{1, \dots, m'\} \setminus J'\} \\
&\quad \times \{x \in \mathbb{R} : a'_{j3}x \leq b'_j \forall j \in J'\} \\
&= \{x \in \mathbb{R}^2 : a'_{j1}x_1 + a'_{j2}x_2 \leq b'_j \forall j \in \{1, \dots, m'\} \setminus J\} \\
&\quad \times \{x \in \mathbb{R} : a_{j3}x \leq b_j \forall j \in J\}.
\end{aligned}$$

Es geht aber die Matrix

$$\left( \begin{array}{cc|c} a'_{11} & a'_{12} & b'_j \\ \vdots & \vdots & \vdots \\ a'_{m'1} & a'_{m'2} & b'_j \end{array} \right)$$

gerade durch das 2-Additive Cutting aus der Matrix

$$\left( \begin{array}{cc|c} a_{11} & a_{12} & b_j \\ \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & b_j \end{array} \right)$$

hervor (0-Zeilen -bis auf den  $b$ -Anteil- sind in beiden Matrizen zu streichen).

Somit ist nach Satz 3.9

$$M'_1 := \{x \in \mathbb{R}^2 : a'_{j1}x_1 + a'_{j2}x_2 \leq b'_j \forall j \in \{1, \dots, m'\} \setminus J\}$$

gerade die konvexe Hülle der ganzzahligen Punkte von

$$M_1 := \{x \in \mathbb{R}^2 : a_{j1}x_1 + a_{j2}x_2 \leq b_j \forall j \in \{1, \dots, m\} \setminus J\}.$$

Ebenso ist wegen

$$a_{j3} = \pm 1 \text{ und } b \in \mathbb{Z}^m$$

(und damit nach Konstruktion des 2-Additiven Cuttings auch  $b' \in \mathbb{Z}^{m'}$ ) die Menge

$$\{x \in \mathbb{R} : a_{j3}x \leq b_j \forall j \in J\}$$

die konvexe Hülle der in ihr enthaltenen ganzzahligen Punkte.

Zusammen folgt wegen Lemma 0.12, dass

$$\begin{aligned} \{x \in \mathbb{R}^3 : A'x \leq b'\} &= \{x \in \mathbb{R}^2 : a'_{j1}x_1 + a'_{j2}x_2 \leq b'_j \forall j \in \{1, \dots, m'\} \setminus J\} \\ &\quad \times \{x \in \mathbb{R} : a_{j3}x \leq b_j \forall j \in J\} \end{aligned}$$

die konvexe Hülle der ganzzahligen Punkte von

$$\begin{aligned} \{x \in \mathbb{R}^3 : Ax \leq b\} &= \{x \in \mathbb{R}^2 : a_{j1}x_1 + a_{j2}x_2 \leq b_j \forall j \in \{1, \dots, m\} \setminus J\} \\ &\quad \times \{x \in \mathbb{R} : a_{j3}x \leq b_j \forall j \in J\} \end{aligned}$$

ist.

□



Für diesen Spezialfall erhalten wir durch das 2-Additive Cutting also die konvexe Hülle des zulässigen Bereichs eines (*GLP*).

Dazu sei allerdings angemerkt, dass jedes derartige Problem äquivalent zu einem 2-dimensionalen Problem ist.

Denn setzen wir für  $j = 1, \dots, m$

$$b'_j := \begin{cases} b_j & , \text{ falls } j \notin J \\ 0 & \text{sonst} \end{cases}$$

und

$$b''_j := \begin{cases} 0 & , \text{ falls } j \notin J \\ b_j & \text{sonst} \end{cases}$$

(wobei  $J$  wie im Beweis von Satz 3.12 definiert ist), so ist der Punkt  $(x_1, x_2, x_3) \in \mathbb{R}^3$  genau dann eine Optimallösung des (*GLP*), wenn  $(x_1, x_2)$  eine Optimallösung von

$$\begin{array}{ll} \min & c_1x_1 + c_2x_2 \\ \text{s.d.} & (A_1|A_2) \cdot x \leq b' \\ & x \in \mathbb{Z}^2 \end{array}$$

und  $x_3$  eine Optimallösung von

$$\begin{array}{ll} \min & c_3x_3 \\ \text{s.d.} & A_3x_3 \leq b'' \\ & x_3 \in \mathbb{Z} \end{array}$$

ist.

Da beide Probleme voneinander unabhängig sind, lösen wir also so keine “echten“ 3-dimensionalen Probleme.

### 3.2.3 Fragestellungen zum 2-Additiven Cutting

Wie in den vorherigen Abschnitten gezeigt wurde, bietet das 2-Additive Cutting für bestimmte (*GLP*) eine gute Möglichkeit, die konvexe Hülle des zulässigen Bereichs zu finden.

Einige Fragen sollen in diesem Abschnitt nun noch angesprochen und teilweise geklärt werden.

Es ist zum Beispiel sehr interessant, zu betrachten, wie viele von den neuen Ungleichungen, die der Algorithmus erstellt, eine echte Verbesserung des relaxierten zulässigen Bereichs bewirken.

Da diese Fragestellung jedoch leider der Probe nach Bedingung (3.3), also ob

$$\{x \in \mathbb{R}^n : Ax \leq b\} \cap \left\{x \in \mathbb{R}^n : \frac{a_1 + a_2}{2} \cdot x > \left\lfloor \frac{b_1 + b_2}{2} \right\rfloor\right\} \neq \emptyset,$$

gleichkommt, ist die Antwort zumeist sehr schwer zu geben.

Ganz offensichtlich ist eine neue Ungleichung

$$\left( \frac{a_1 + a_2}{2} \left\lfloor \frac{b_1 + b_2}{2} \right\rfloor \right)$$

jedoch mindestens dann unnötig, wenn bereits eine Ungleichung  $(a_3|b_3)$  mit

$$a_3 = \frac{a_1 + a_2}{2} \text{ und } b_3 \leq \left\lfloor \frac{b_1 + b_2}{2} \right\rfloor$$

existiert, da diese schärfer (oder genauso scharf) ist.

Der Algorithmus könnte also durch einen entsprechenden Test noch verbessert werden.

Nun ist für die Effizienz des 2-Additiven Cuttings ebenfalls die Fragestellung interessant, ob die Reihenfolge des Hinzufügens der Ungleichungen dafür eine Rolle spielt, ob die Ungleichungen den relaxierten zulässigen Bereich im Moment des Hinzufügens echt verkleinern.

Um die Problematik zu verdeutlichen betrachten wir folgendes Beispiel:

**Beispiel 3.13.** Gegeben sei das lineare Programm

$$\begin{aligned} (GLP5) \quad & \min \quad c^t x \\ & s.d. \quad \begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & -1 \end{pmatrix} x \leq \begin{pmatrix} 3 \\ 0 \\ 1 \\ 0 \end{pmatrix} \\ & x \in \mathbb{Z}^3. \end{aligned}$$

Wir sehen, dass das 2-Additive Cutting zwei Ungleichungen zur Matrix hinzufügt:

$$\left( \frac{(1, 1, 0) + (1, -1, 0)}{2} \left\lfloor \frac{3+0}{2} \right\rfloor \right) = ((1, 0, 0)|1) \quad (3.5)$$

und

$$\left( \frac{(1, 0, 1) + (1, 0, -1)}{2} \left\lfloor \frac{1+0}{2} \right\rfloor \right) = ((1, 0, 0)|0) \quad (3.6)$$

Es ist leicht ersichtlich, dass (3.6) schärfer ist als (3.5), der Test nach Bedingung (3.3), also ob

$$\{x \in \mathbb{R}^n : Ax \leq b\} \cap \left\{x \in \mathbb{R}^n : \frac{a_{i_1} + a_{i_2}}{2} \cdot x > \left\lfloor \frac{b_{i_1} + b_{i_2}}{2} \right\rfloor\right\} \neq \emptyset,$$

fällt somit je nachdem, welche Ungleichung wir zuerst hinzufügen, entweder zweimal positiv (zuerst (3.5) und dann (3.6)) oder nur einmal positiv (zuerst (3.6) und dann (3.5)) aus.

Im ersten Fall müssten wir zwei Ungleichungen hinzufügen, im zweiten Fall nur eine.

Wird in jedem Schritt des Algorithmus Bedingung (3.3) getestet, so ist die Reihenfolge des Hinzufügens der Ungleichungen also zumeist von Wichtigkeit, da unter Umständen durch eine geschickte Wahl dieser Reihenfolge weniger Ungleichungen hinzugefügt werden müssen.

Dies gilt jedoch nicht für jedes Problem, wie der folgende Satz zeigt:

**Satz 3.14.** *Sei ein 2-dimensionales (GLP) mit nicht-leerem zulässigem Bereich gegeben, bei dem keine zwei Zeilen der Matrix  $A$  identisch und keine zwei Zeilen der Matrix  $(A|b)$  Vielfache voneinander seien.*

*Dann ist die Reihenfolge des Hinzufügens neuer Ungleichungen im 2-Additiven Cutting für den Test von Bedingung (3.3) unerheblich.*

*Beweis.* Es genügt, zu zeigen, dass die Reihenfolge des Hinzufügens zweier beliebiger Ungleichungen vertauscht werden kann.  
Seien

$$(a_{neu}|b_{neu}) := \left( \frac{a_1 + a_2}{2} \left| \left\lfloor \frac{b_1 + b_2}{2} \right\rfloor \right. \right)$$

und

$$(a'_{neu}|b'_{neu}) := \left( \frac{a'_1 + a'_2}{2} \left| \left\lfloor \frac{b'_1 + b'_2}{2} \right\rfloor \right. \right)$$

mit

$$\{(a_1|b_1), (a_2|b_2)\} \neq \{(a'_1|b'_1), (a'_2|b'_2)\}$$

zwei Ungleichungen, die im Verlauf des Algorithmus hinzugefügt werden, die also die entsprechenden Voraussetzungen (3.1) und (3.2) und

$$a_{11}, a_{12}, a_{21}, a_{22}, a'_{11}, a'_{12}, a'_{21}, a'_{22} \neq 0$$

erfüllen.

Außerdem gelte

$$\{x \in \mathbb{R}^2 : a_{neu}x > b_{neu}\} \cap \{x \in \mathbb{R}^2 : Ax \leq b\} \neq \emptyset.$$

Wir zeigen nun

$$\{x \in \mathbb{R}^2 : a'_{neu}x \leq b'_{neu}\} \cap \{x \in \mathbb{R}^2 : a_{neu}x > b_{neu}\} \cap \{x \in \mathbb{R}^2 : Ax \leq b\} \neq \emptyset.$$

Daraus folgt die Behauptung, da die Reihenfolge des Hinzufügens für den Test von (3.3) somit keine Rolle spielt.

Sei

$$x^* \in \{x \in \mathbb{R}^2 : a_{neu}x > b_{neu}\} \cap \{x \in \mathbb{R}^2 : Ax \leq b\} \neq \emptyset.$$

Angenommen

$$\{x \in \mathbb{R}^2 : a'_{neu}x \leq b'_{neu}\} \cap \{x \in \mathbb{R}^2 : a_{neu}x > b_{neu}\} \cap \{x \in \mathbb{R}^2 : Ax \leq b\} = \emptyset,$$

dann gilt insbesondere

$$x^* \notin \{x \in \mathbb{R}^2 : a'_{neu}x \leq b'_{neu}\}$$

und also

$$a'_{neu}x^* > b'_{neu}.$$

Somit verbietet das Hinzufügen von  $(a'_{neu}|b'_{neu})$  den Punkt  $x^*$  ebenso wie das Hinzufügen von  $(a_{neu}|b_{neu})$ .

Nun besitzt jede der beiden Ungleichungen  $(a_{neu}|b_{neu})$  und  $(a'_{neu}|b'_{neu})$  wegen Lemma 3.3.2 die Struktur

$$\pm x_1 \leq b_{neu} \text{ bzw. } b'_{neu}$$

oder

$$\pm x_2 \leq b_{neu} \text{ bzw. } b'_{neu}.$$

Es ergeben sich drei Fälle:

*Fall 1:*  $a_{neu} = a'_{neu}$

Da  $a_1$  und  $a_2$  linear unabhängig sind, können wir o.B.d.A.  $a_{neu 1} \neq 0$  annehmen, ansonsten sortieren wir die Spalten der Matrix um.

Dann ist wegen  $a_{neu} = a'_{neu}$  auch

$$a_{11} = a_{21} = a'_{11} = a'_{21}.$$

Weiter muss wegen Lemma 3.3.2

$$a_{neu 2} = a'_{neu 2} = 0$$

gelten. Weil  $a_1$  und  $a_2$  bzw.  $a'_1$  und  $a'_2$  linear unabhängig sind, ergeben sich

$$0 \neq a_{12} = -a_{22} \neq 0$$

und

$$0 \neq a'_{12} = -a'_{22} \neq 0.$$

Es folgt dann aber wegen der  $\{1,0,-1\}$ -Struktur von  $A$

$$\{a_1, a_2\} = \{a'_1, a'_2\}$$

und da keine zwei Zeilen von  $A$  identisch sind ergibt sich

$$\{(a_1|b_1), (a_2|b_2)\} = \{(a'_1|b'_1), (a'_2|b'_2)\}.$$

Dies ist ein Widerspruch zur Wahl von  $a_{neu}$  und  $a'_{neu}$ .

Fall 2:  $a_{neu} = -a'_{neu}$

Die beiden Ungleichungen induzieren Halbräume

$$\{x \in \mathbb{R}^2 : a_{neu}x \leq b_{neu}\} \text{ und } \{x \in \mathbb{R}^2 : a'_{neu}x \leq b'_{neu}\}.$$

Da die beiden Geraden

$$\{x \in \mathbb{R}^2 : a_{neu}x = b_{neu}\} \text{ und } \{x \in \mathbb{R}^2 : a'_{neu}x = b'_{neu}\}$$

parallel sind und der Punkt  $x^*$  in keinem der beiden Halbräume liegt, muss der Schnitt der Halbräume nach Lemma 0.13 leer sein.

Weil aber beide Ungleichungen nach Lemma 3.3 gültig für den zulässigen Bereich des  $(GLP)$  sind, muss dieser zulässige Bereich leer sein.

Dies ist ein Widerspruch zu den Voraussetzungen des Satzes.

Fall 3:  $a_{neu} \neq \pm a'_{neu}$

Dann folgt wegen der Struktur von  $a_{neu} \neq (0,0)$  und  $a'_{neu} \neq (0,0)$

$$(a_{neu\ i} = \pm 1) \Leftrightarrow (a'_{neu\ i} = 0) \quad \forall i = 1, 2$$

und

$$(a'_{neu\ i} = \pm 1) \Leftrightarrow (a_{neu\ i} = 0) \quad \forall i = 1, 2$$

Somit verschwindet das Skalarprodukt von  $a_{neu}$  und  $a'_{neu}$  und also stehen die beiden Geraden

$$g_{neu} := \{x \in \mathbb{R}^n : a_{neu}x = b_{neu}\}$$

und

$$g'_{neu} := \{x \in \mathbb{R}^n : a'_{neu}x = b'_{neu}\}$$

senkrecht aufeinander.

Außerdem schneiden sie sich in einem ganzzahligen Punkt, nämlich (je nachdem ob  $a_{neu1} \neq 0$  oder  $a_{neu1} = 0$ ) in

$$s_1 := (a_{neu1}b_{neu}, a'_{neu2}b'_{neu})$$

oder

$$s_2 := (a'_{neu1}b'_{neu}, a_{neu2}b_{neu})$$

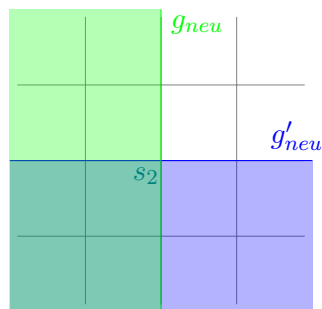
Es ergeben sich nun mehrere Kombinationsmöglichkeit für die Struktur von  $a_{neu}$  und  $a'_{neu}$ .

Hier wird nur der Fall

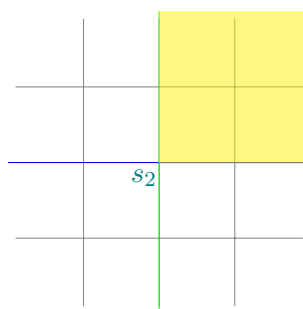
$$\begin{aligned} a_{neu} &= (0, 1) \\ \wedge \quad a'_{neu} &= (1, 0) \end{aligned}$$

betrachtet, für alle anderen Fälle folgt die Behauptung analog zu diesem.

Grafisch veranschaulicht stehen die Ungleichungen also folgendermaßen zueinander:



Da beide Ungleichungen den Punkt  $x^*$  verbieten, muss dieser in dem gelb eingefärbten Bereich liegen:



Nun muss aber wegen des Verfahrens des Algorithmus gelten:

$$a_1 = (\pm 1, 1)$$

und

$$a_2 = (\mp 1, 1)$$

O.B.d.A. setzen wir

$$a_1 = (1, 1)$$

und

$$a_2 = (-1, 1)$$

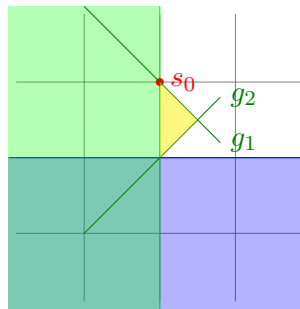
Wir fügen die Geraden

$$g_1 := \{x \in \mathbb{R}^n : a_1 x = b_1\}$$

und

$$g_2 := \{x \in \mathbb{R}^n : a_2 x = b_2\}$$

der Zeichnung hinzu und benennen den Schnittpunkt von  $g_1$  und  $g_{neu}$  mit  $s_0$ :



Die genaue Einordnung von  $g_1$  und  $g_2$  an  $g_{neu}$  ergibt sich dabei aus dem Verhältnis von  $b_1$  und  $b_2$  zu  $b_{neu}$ , denn der Schnittpunkt  $x_g$  der beiden Geraden  $g_1$  und  $g_2$  erfüllt

$$a_1 x_g = b_1 \text{ und } a_2 x_g = b_2$$

und somit

$$a_{neu} x_g = \frac{b_1 + b_2}{2} = b_{neu} + 0.5,$$

während alle Punkte  $x_{g'}$  auf der Geraden  $g_{neu}$  die Bedingung

$$a_{neu} x_{g'} = b_{neu}$$

erfüllen.

Die Zeichnung vereinfacht hier aber insofern, als dass  $g'_{neu}$  durchaus einen größeren Abstand zu  $g_1$  und  $g_2$  besitzen kann. Allerdings kann  $g'_{neu}$  lediglich weiter nach unten verschoben sein, da die Ungleichung ( $a'_{neu} | b'_{neu} \in \mathbb{Z}$ ) den Punkt  $x^*$  verbietet und dieser nach seiner Definition in dem gelb eingefärbten Bereich liegen muss.

Da nun

$$U := \{x \in \mathbb{R}^2 : a_{neu} x > b_{neu}\} \cap \{x \in \mathbb{R}^2 : Ax \leq b\} \neq \emptyset \quad (3.7)$$

gilt, erkennen wir, dass der Punkt  $s_0$  ebenfalls für das (GLP) zulässig sein muss:

Wäre  $s_0$  nicht zulässig und trotzdem (3.7) erfüllt (also Teile des gelb eingefärbten Bereichs für die LP-Relaxation des (GLP) zulässig), würde eine Ungleichung  $(a^*|b^*)$  so existieren, dass die Gerade

$$g^* := \{x \in \mathbb{R}^2 : a^*x = b^*\}$$

- mindestens eine der Geraden  $g_{neu}$  und  $g'_{neu}$  in einem nicht-ganzzahligen Punkt schneiden würde.

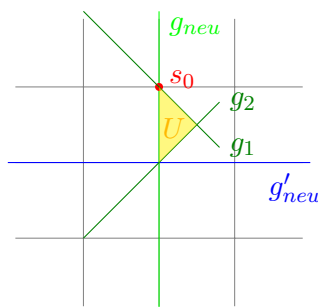
Dies ist aber nicht möglich, da wegen (3.7)  $g^*$  nicht identisch mit  $g_{neu}$  bzw.  $g'_{neu}$  sein kann, und wegen  $a^* \in \{1, 0, -1\}^{1 \times 2}$  und  $b^* \in \mathbb{Z}$  die Gerade  $g^*$  einen ganzzahligen Schnittpunkt mit  $g_{neu}$  bzw.  $g'_{neu}$  besitzt, nämlich

$$(-a_1^* \cdot (a_2^* \cdot b_{neu} - b^*), b_{neu})$$

bzw.

$$(b'_{neu}, -a_2^* \cdot (a_1^* \cdot b'_{neu} - b^*)).$$

- zu der Geraden  $g_1$  bzw.  $g_2$  parallel wäre.  
Da  $g^*$  die Gerade  $g'_{neu}$  bzw.  $g_{neu}$  in einem ganzzahligen Punkt schneidet, kann (3.7) nur gelten, wenn  $g^* = g_1$  bzw.  $g_2 = g^*$ , da aber nach Voraussetzung  $(a_2|b_2) \neq \lambda(a^*|b^*)$  für jedes  $\lambda \in \mathbb{R}$ , gilt auch  $g_2 \neq g^*$ .



Grafik zur Veranschaulichung

Andere Möglichkeiten,  $s_0$  zu verbieten und gleichzeitig (3.7) zu erfüllen gibt es wegen der  $\{1,0,-1\}$ -Struktur des (GLP) nicht.

Da keine der Möglichkeiten zutrifft, muss also der Punkt  $s_0$  zulässig für das (GLP) sein, da (3.7) erfüllt ist.

Dies ist nun ein Widerspruch zu der Annahme, dass die Ungleichung  $(a'_{neu}|b'_{neu})$  im 2-Additiven Cutting entsteht, da sie nach Lemma 3.3.3 für das (GLP) gültig wäre. Sie verbietet aber den Punkt  $s_0$  und ist somit keine gültige Ungleichung, da dieser zulässig für das (GLP) ist.

In allen drei Fällen erhalten wir einen Widerspruch, unsere Annahme muss also falsch sein.

Es folgt die Behauptung. □



Es sei angemerkt, dass die Voraussetzungen von Satz 3.14 unterschiedlich einschränkend sind.

Die Voraussetzung, dass keine zwei Zeilen von  $A$  identisch sein dürfen, verbietet lediglich einige überflüssige Ungleichungen.

Existieren zwei Ungleichungen mit demselben  $A$ -Anteil, so ist die Ungleichung mit dem größeren  $b$ -Anteil stets redundant.

Dagegen ist die Voraussetzung, dass keine zwei Zeilen von  $(A|b)$  Vielfache voneinander sein dürfen, stärker einschränkend, denn sie verbietet die Modellierung von Gleichheitsbedingungen.

Zumeist wird die Nebenbedingung  $ax = b$  durch die beiden Ungleichungen  $(a|b)$  und  $(-a| -b)$  modelliert.

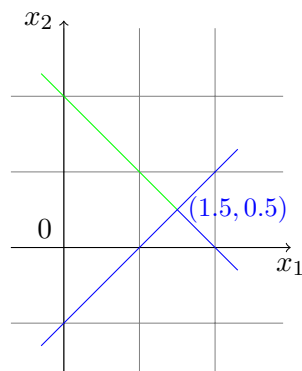
Die Voraussetzung, dass keine zwei Zeilen Vielfache voneinander sein dürfen, verbietet nun aber diese Modellierung.

Dass diese Voraussetzung allerdings wirklich notwendig ist, zeigt das folgende Beispiel:

**Beispiel 3.15.** Gegeben sei das lineare Programm

$$\begin{aligned}
 (GLP6) \quad & \min \quad c^t x \\
 & s.d. \quad \begin{pmatrix} 1 & 1 \\ -1 & -1 \\ 1 & -1 \end{pmatrix} \cdot x \leq \begin{pmatrix} 2 \\ -2 \\ 1 \end{pmatrix} \\
 & \quad \quad \quad x \in \mathbb{Z}^2.
 \end{aligned}$$

Als relaxierter zulässiger Bereich ergibt sich der hier grün eingefärbte Bereich:



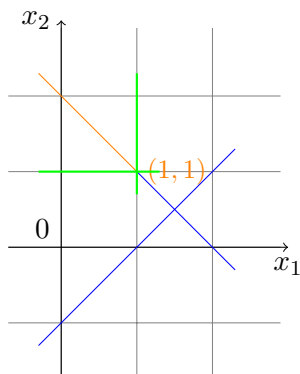
Das 2-Additive Cutting fügt nun zwei Nebenbedingungen hinzu:

$$\left( \frac{(1, 1) + (1, -1)}{2} \left| \left\lfloor \frac{2+1}{2} \right\rfloor \right. \right) = ((1, 0)|1)$$

und

$$\left( \frac{(-1, -1) + (1, -1)}{2} \left| \left\lfloor \frac{-2+1}{2} \right\rfloor \right. \right) = ((0, -1)|-1)$$

Der neue relaxierte zulässige Bereich ist der orange eingefärbte Bereich, dieser ist die konvexe Hülle des zulässigen Bereichs von (GLP6):



Wie an der Zeichnung leicht erkennbar ist, muss nur eine von den beiden neuen Nebenbedingungen zur Matrix  $(A|b)$  hinzugefügt werden, um die konvexe Hülle des zulässigen Bereichs zu beschreiben.

In Beispiel 3.15 betrachten wir ein nicht-volldimensionales Polyeder. Dies wirft die Frage auf, ob es lediglich bei nicht-volldimensionalen Polyedern dazu kommen kann, dass die Voraussetzungen von Satz 3.14 nicht erfüllt sind.

**Satz 3.16.** *Im 2-dimensionalen Raum erfüllt die beschreibende Matrix eines voll-dimensionalen Polyeders die Bedingung, dass keine zwei Zeilen von  $(A|b)$  linear abhängig sind.*

*Beweis.* O.B.d.A. sind keine zwei Zeilen von  $(A|b)$  identisch, ansonsten ist eine von beiden redundant.

Sind nun zwei Zeilen  $(a_1|b_1)$  und  $(a_2|b_2)$  aus  $(A|b)$  linear abhängig, so gilt

$$(a_1|b_1) = \lambda \cdot (a_2|b_2)$$

für ein  $\lambda \in \mathbb{R} \setminus \{0, 1\}$ .

Wegen  $a_1, a_2 \in \{1, 0, -1\}^{1 \times 2} \setminus \{(0, 0)\}$  folgt direkt

$$\lambda = -1$$

und somit modellieren  $(a_1|b_1)$  und  $(a_2|b_2)$  eine Gleichheits-Nebenbedingung, das Polyeder kann also nicht volldimensional sein. □

Die Sätze 3.14 und 3.16 führen zusammen zu dem Ergebnis, dass im 2-dimensionalen Raum die Reihenfolge des Hinzufügens neuer Ungleichungen im 2-Additiven Cutting für jedes volldimensionale Polyeder unerheblich ist.

### 3.3 Das $k$ -Additive Cutting

Betrachtet man das Abrundungs-Verfahren und das 2-Additive Cutting, so ist ein Muster erkennbar:

Um zu sichern, dass sich in jeder zulässigen  $(n - k + 1)$ -dimensionalen Extremalmenge ganzzahlige Punkte befinden, werden  $k$  Ungleichungen ausgewählt, addiert, das Ergebnis durch  $k$  dividiert und abgerundet.

Im 2-Additiven-Cutting ist dabei  $k = 2$  und im Abrundungs-Verfahren  $k = 1$ , wir könnten dieses Verfahren also auch 1-Additives Cutting nennen.

Nun wäre es möglich, diese Methode zu verallgemeinern. So erhielte man ein  $k$ -Additives Cutting.

Allerdings entstehen dabei schon für  $k = 3$  Ungleichungen, die auch ganzzahlige Punkte abschneiden, also nicht zur konvexen Hülle des zulässigen Bereichs führen. Das folgende Beispiel zeigt einen solchen Fall:

**Beispiel 3.17.** Gegeben sei das Problem

$$(GLP7) \quad \min \quad c^t x$$

$$s.d. \quad \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & -1 & 0 \end{pmatrix} \cdot x \leq \begin{pmatrix} 0 \\ 5 \\ 0 \end{pmatrix}$$

$$x \in \mathbb{Z}^3.$$

Wir sehen, dass die Zeilen von  $A$  linear unabhängig sind.

Die Ungleichung

$$\left( \left\lfloor \frac{(1, 0, 1) + (1, 1, 0) + (1, -1, 0)}{3} \right\rfloor \left\| \left\lfloor \frac{0 + 5 + 0}{3} \right\rfloor \right\| \right) = ((1, 0, 0) | 1)$$

ist nun aber keine gültige Ungleichung, da sie den Punkt  $(2, 2, -2)$  verbietet, dieser aber für das (GLP) zulässig ist.

An diesem Beispiel wird ersichtlich, dass eine einfache Übertragung des erkannten Musters ( $k$  Ungleichungen addieren, durch  $k$  dividieren und dann abrunden) zu keinem funktionierenden Verfahren führt.

Allerdings war diese einfache Übertragung auch beim 2-Additiven Cutting nicht möglich. Hierbei dürfen (und müssen) nur solche Ungleichungspaare betrachtet werden, die die Bedingungen (3.1) und (3.2) erfüllen.

Ähnliche (oder gänzlich andersartige) Forderungen an die Auswahl der zu addierenden Ungleichungen könnten also auch für k-Additives Cutting zu funktionierenden Verfahren führen.

Dies ist nun lediglich eine persönliche Vermutung, die noch nicht genauer betrachtet werden konnte.

Sie eröffnet aber Möglichkeiten und eventuelle Ansätze für spätere Forschungen.

### 3.3.1 Unterschiede zwischen dem k-Additiven Cutting und anderen Cutting-Verfahren

In der Literatur finden sich viele verschiedene Cutting-Verfahren (vergleiche dazu z.B. Dück [3], Nemhauser and Wolsey [5], Parker [6] und Plane [7]).

Wir wollen in diesem Abschnitt eines davon, das Gomory-Schnittebenenverfahren (siehe Nemhauser and Wolsey [5], Kapitel 2) näher beschreiben und Unterschiede zwischen diesem Verfahren und dem 2-Additiven Cutting aufzeigen.

Das Gomory-Schnittebenenverfahren eignet sich für diese Zwecke besonders, da, wie später noch genauer ausgeführt, die meisten anderen Cutting-Verfahren demselben strukturellen Ansatz folgen und gerade das Gomory-Schnittebenenverfahren dem 2-Additiven Cutting ähnlich erscheinen mag.

Beim Gomory-Schnittebenenverfahren wird stets eine optimale Lösung der LP-Relaxation des betrachteten Problems berechnet und diese, falls sie nicht ganzzahlig ist, anschließend durch Hinzufügen einer Ungleichung abgeschnitten.

Die Erstellung der neuen Ungleichung geschieht dabei durch Auswahl und Modifizierung einer bestimmten Zeile von  $(A|b)$ , abhängig von der zuvor berechneten Optimallösung der LP-Relaxation.

Bei der Modifizierung spielt auch das Runden eine Rolle, weswegen dieses Verfahren und das 2-Additive Cutting zunächst ähnlich erscheinen mögen.

Die Unterschiede sind nun aber direkt ersichtlich:

Im Gomory-Schnittebenenverfahren wird stets zunächst eine Optimallösung der LP-Relaxation berechnet und anschließend eine einzelne neue Ungleichung hinzugefügt, die diese Lösung verbietet.

Die meisten anderen Cutting-Verfahren wenden ebenfalls dieses Prinzip an, während das 2-Additive Cutting hier einem anderen Ansatz folgt. Hier muss keine Lösung berechnet werden, denn es wird bereits an der Struktur der Matrix erkannt, welche Cuts hinzugefügt werden müssen.

Das Gomory-Schnittebenenverfahren berücksichtigt also auch die Zielfunktion des aktuell betrachteten Problems (also den Vektor  $c$ ), während das 2-Additive Cutting sich auf die Struktur der Matrix  $(A|b)$  beschränkt und das Problem daher unabhängig von der Zielfunktion bearbeitet.

Dadurch ist das 2-Additive Cutting auf mehrere Arten einsetzbar. Zum Einen kann es als *Pre-Processing-Verfahren* (also zur Modifizierung des Problems vor dessen eigentlicher Lösung) eingesetzt werden.

Zum Anderen ist natürlich auch folgendes Vorgehen möglich:

Zunächst wird eine Optimallösung der LP-Relaxation des Problems berechnet. Nun kann leicht überprüft werden, in welchen Facetten des relaxierten zulässigen Bereichs diese Lösung liegt, denn da jede Facette durch eine Ungleichung  $(a_i|b_i)$  repräsentiert wird (Satz 0.15), liegt die berechnete (zulässige) Lösung genau dann in der von  $(a_i|b_i)$  beschriebenen Facette, wenn sie diese Ungleichung mit Gleichheit erfüllt.

Schließlich kann die berechnete (nicht-ganzzahlige) Lösung abgeschnitten werden, indem aus den von ihr mit Gleichheit erfüllten Nebenbedingungen zwei ausgewählt werden (unter Berücksichtigung der Bedingungen (3.1) und (3.2)) und entsprechend Lemma 3.3 eine neue Ungleichung hinzugefügt wird. Diese wird die vorher berechnete Lösung verbieten (siehe Bemerkung 3.4).

Diese Variation des 2-Additiven Cuttings, deren grobe Struktur der des Gomory-Schnittebenenverfahrens gleich ist, bietet unter Umständen eine Möglichkeit, die benötigte Rechenzeit zu verkürzen, da hierbei nicht zunächst alle möglichen Paare von Nebenbedingungen kombiniert werden müssen.

Leider funktioniert das 2-Additive Cutting in beiden Variationen nur für Spezialfälle eines allgemeinen ( $GLP$ ). In den meisten Fällen wird es lediglich den zulässigen relaxierten Bereich verkleinern.

Außerdem ist zu erwähnen, dass das 2-Additive Cutting noch numerisch auf seine Effizienz und Laufzeit zu prüfen ist, da diese beiden Aspekte bis jetzt nicht näher betrachtet werden konnten.

Es ist bekannt, dass sich jede für ein ( $GLP$ ) gültige Ungleichung durch iterierte Linearkombination der Zeilen der beschreibenden Matrix mit anschließendem Runden darstellen lässt (vergleiche auch Chvátal [2]).

Dabei ist die Auswahl der Nebenbedingungen, die miteinander linear kombiniert werden sollten, bisher jedoch rein zufällig.

Es kann also jede gültige Ungleichung als entsprechende Linearkombination dargestellt werden, welche ausgewählten Ungleichungen eine vielversprechende Linearkombination besitzen, ist jedoch a priori nicht klar.

Das 2-Additive Cutting (bzw. Satz 3.2) liefert nun aber Bedingungen, wann eine Linearkombination von genau zwei solchen Nebenbedingungen überhaupt nötig ist. Für jede Linearkombination von zwei Ungleichungen kann demnach mit den Bedingungen von Satz 3.2 leicht geprüft werden, ob eine solche Linearkombination vielversprechend ist oder nicht.

Es wäre sicherlich wünschenswert, auch für eine Auswahl von  $k$  Ungleichungen solche Bedingungen zu erhalten, da hierdurch leichter entschieden werden könnte, welche Linearkombinationen von Nebenbedingungen für ein gegebenes Problem überhaupt von Interesse sind (also Ungleichungen liefern, die den relaxierten zulässigen Bereich echt verkleinern).

Diese Bedingungen könnten dann auch zur Herleitung eines  $k$ -Additiven Cuttings dienen, welches die konvexe Hülle des zulässigen Bereichs schon vor der Lösung einer Relaxation berechnen könnte.

Außerdem könnten diese Auswahlbedingungen helfen, bereits existierende Cutting-Verfahren zu beschleunigen.

## 4 Fazit

In dieser Arbeit wurden allgemeine ganzzahlige lineare Programme mit rationaler Matrix  $A$  betrachtet.

Es wurde eine bestimmte Formulierung solcher Programme, die  $\{1,0,-1\}$ -Form, eingeführt und gezeigt, dass die Forderung nach dieser Formulierung keine weitere Einschränkung der betrachteten Probleme bedeutet.

Zwei Verfahren sind entwickelt worden, die ein allgemeines ganzzahliges lineares Programm mit rationaler Matrix  $A$  in  $\{1,0,-1\}$ -Form überführen. Diese Verfahren wurden in ihrer Effizienz verglichen, wobei sich zeigte, dass das Verfahren der Binärcodierung eine wesentlich bessere Möglichkeit der ( $GLP$ )-Umformung darstellt als das additive Verfahren.

Durch Ausnutzen der  $\{1,0,-1\}$ -Form war es schließlich möglich, das 2-Additive Cutting zu entwickeln, welches zum Ziel hat, den relaxierten zulässigen Bereich eines ( $GLP$ ) zu verkleinern.

Im 2-dimensionalen Fall führt dieses Verfahren, wie bewiesen wurde, sogar zu der konvexen Hülle des zulässigen Bereichs.

Dies gilt auch für höher-dimensionale Spezialfälle, Beispiele dazu sind näher betrachtet worden.

Für das 2-Additive Cutting sind einfache Kriterien entwickelt worden, nach denen die zur Erstellung der Cuts verwendeten Ungleichungspaare ausgewählt werden.

Diese Kriterien erweitern die bisher bekannte Aussage, dass jede für ein ( $GLP$ ) zulässige Ungleichung durch iterierte Linearkombination der Zeilen der beschreibenden Matrix darstellbar ist.

Ausgehend von dem betrachteten 2-Additiven Cutting wurde ein allgemeines  $k$ -Additives Cutting angesprochen.

Dieses ist noch nicht realisiert, allerdings bietet sich eine nähere Betrachtung, motiviert durch die Erkenntnisse des 2-Additiven Cuttings, an.

Dazu wurde aufgezeigt, dass sich das 2-Additive Cutting in seinem Ansatz von vielen anderen Cutting-Verfahren unterscheidet.

Nach diesem Ansatz wird nicht nur ein bestimmtes Optimierungsproblem gelöst, sondern es werden der zulässige Bereich eines Problems und somit alle Optimierungsprobleme mit demselben zulässigen Bereich vereinfacht.

Dieser Lösungsweg ist gerade im Bereich der Robusten Optimierung (vergleiche z.B. Schöbel [10]) von Interesse. Dort gilt es häufig, nacheinander viele Probleme zu lösen, die denselben zulässigen Bereich besitzen.

Gleichzeitig ergeben sich durch die Ergebnisse des 2-Additiven Cuttings Möglichkeiten, bereits bestehende Cutting-Verfahren zu modifizieren und eventuell zu beschleunigen.

Hierfür bieten sich die in Kapitel 3.2 gefundenen Bedingungen zur Auswahl von linear zu kombinierenden Nebenbedingungen an.



# Literaturverzeichnis

- [1] Dieser Satz wurde dem Autor entweder irgendwann während seines Studiums vermittelt oder ergab sich ihm intuitiv. Da eine entsprechende Textstelle nicht gefunden werden konnte, ist hier ein Beweis angegeben. Der Satz wurde jedoch nicht vom Autor entwickelt.
- [2] V. Chvátal. „Edmonds polytopes and a hierarchy of combinatorial problems“. *Discrete Mathematics* 4, 1973.
- [3] W. Dück. *Diskrete Optimierung*. Akademie-Verlag Berlin, 1977.
- [4] D. Jungnickel. *Optimierungsmethoden*. Springer, 2008.
- [5] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.
- [6] R.G. Parker and R.L. Rardin. *Discrete Optimization*. Academic Press, 1988.
- [7] D.R. Plane and C. McMillan, Jr. *Discrete Optimization*. Prentice Hall, 1971.
- [8] A. Schöbel. *Einführung in die Optimierung*. Skript zur Vorlesung an der Universität Göttingen, 2007.
- [9] A. Schöbel. *Ganzzahlige Programmierung*. Skript zur Vorlesung an der Universität Göttingen, 2008.
- [10] A. Schöbel. *Robuste Optimierung*. Skript zur Vorlesung an der Universität Göttingen, 2009.