Technische Universität München
Zentrum Mathematik

Diskrete Optimierung: Fallstudien aus der Praxis

Barbara Langfeld | Michael Ritter | Barbara Wilhelm

Schnittebenen: Grundlagen und Gomory-Schnitte

1 Grundidee

1.1 Problemstellung

Gegeben ist ein ganzzahliges lineares Optimierungsproblem der Form

$$\begin{aligned} \max\ & c^T x \\ & Ax \le b \\ & x \in \mathbb{Z}^n. \end{aligned}$$

Mit dem *Branch and Bound-Verfahren* haben wir bereits eine Möglichkeit kennengelernt, solche Probleme zu lösen. Heute und in den folgenden Stunden beschäftigen wir uns mit einem grundlegend anderen Ansatz. Branch and Bound versucht, über die Betrachtung von Teilproblemen alle Lösungen systematisch aufzuzählen. Dabei werden Kriterien verwendet, die ganze Klassen von Teilproblemen möglichst früh ausschließen sollen, um die Anzahl betrachteter Lösungen gering zu halten. Im Gegensatz dazu stellen Schnittebenen-Verfahren einen geometrisch motivierten Ansatz dar.

1.2 Bezeichnungen

Wir benötigen zunächst ein paar Bezeichnungen.

Definition 1
Seien $A \in \mathbb{Z}^{m\times n}$, $b \in \mathbb{Z}^m$ und $c \in \mathbb{Z}^n$ gegeben. Dann bezeichnen wir mit $P(A, b)$ den zulässigen Bereich des obigen LPs, d. h.,

$$P(A, b) := \{x \in \mathbb{R}^n : Ax \le b\},$$

Die *ganzzahlige Hülle* $P_I(A, b)$ von $P(A, b)$ ist definiert als

$$P_I(A, b) := \operatorname{conv}\Big(P(A, b) \cap \mathbb{Z}^n\Big) = \operatorname{conv}\{x \in \mathbb{Z}^n : Ax \le b\}.$$

Wenn keine Verwechslungsgefahr besteht, schreiben wir auch einfach P und P_I, vgl. Abbildung.

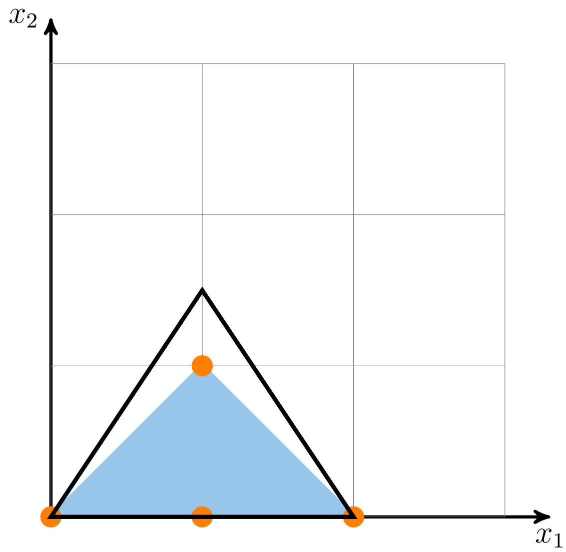

Abbildung 1: P und P_I

Definition 2
Sei P ein Polyeder, $a \in \mathbb{R}^n$ und $\beta \in \mathbb{R}$. Dann heißt die Ungleichung

$$a^T x \leq \beta$$

gültig für P, falls $a^T x \leq \beta$ für alle $x \in P$. Eine für P_I gültige Ungleichung heißt auch *Schnittebene* für P_I. Eine Schnittebene heißt *uneigentlich*, wenn sie sowohl für P als auch für P_I gültig ist, und *eigentlich* sonst.

Definition 3
Sei $p \in \mathbb{R}$, dann heißt

$$\langle p \rangle := p - \lfloor p \rfloor$$

der fraktionelle Anteil von p. Für einen Vektor $v \in \mathbb{R}^n$ ist $\langle v \rangle$ entsprechend komponentenweise definiert.

1.3 Grundidee

Die Idee eines Schnittebenen-Verfahrens besteht darin, aus der $\mathcal{H}$-Darstellung für P sukzessive eine $\mathcal{H}$-Darstellung für P_I zu bestimmen. Um das zu erreichen, werden Ungleichungen gesucht, die für P_I gültig sind und möglichst große Teile von P „abschneiden“. Diese Ungleichungen bezeichnet man als *Schnittebenen*. Gewinnt man so eine $\mathcal{H}$-Darstellung für P_I, so kann man das ganzzahlige Optimierungsproblem mit einem Simplex-Algorithmus lösen, weil alle Ecken von P_I ganzzahlig sind. Der Simplex-Algorithmus findet immer eine optimale Ecke und diese wäre in P_I dann automatisch ganzzahlig.

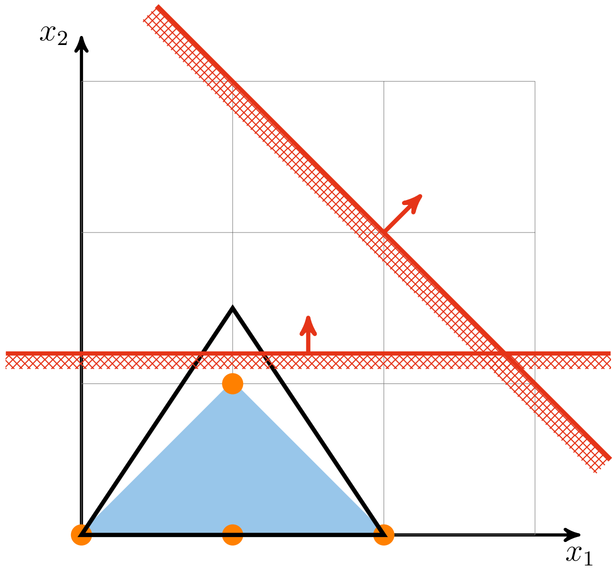

Abbildung 2: Gültige Ungleichungen für P_I und eine Schnittebene.

Meist lässt man sich bei der Suche nach Schnittebenen von einer bekannten Ecke in P leiten, die aber nicht ganzzahlig ist. Hat man so einen Punkt, dann sucht man eine Schnittebene, die diesen Punkt abschneidet, aber für alle ganzzahligen Punkte in P (und damit für P_I) gültig ist.

2 Wichtige Fragestellungen

Die Suche nach Schnittebenen wirft ein paar Fragestellungen auf, die wir kurz zusammenstellen wollen:

- Wie findet man überhaupt eine Schnittebene?
- Wenn man eine nicht ganzzahlige (*fraktionelle*) Ecke in P kennt, wie findet man eine Schnittebene, die diesen Punkt abschneidet?
- Wie „gut" ist eine bestimmte Schnittebene? Wie misst man überhaupt die Güte eines Schnitts? Kann man so etwas wie „bestmögliche" Schnittebenen bestimmen?
- Wie vergleicht man verschiedene Schnittebenen miteinander? Geht das überhaupt? Welche Schnitte sind „besser" als andere?
- Wie viele Schnittebenen gibt es?
- Wie viele Schnittebenen braucht man für gegebenes P, um eine H-Darstellung für P_I zu bestimmen?
- Gibt es überhaupt ein endliches oder wenigstens ein konvergentes Verfahren, um P_I mittels Schnittebenen zu bestimmen?
- Wie effizient sind Schnittebenen-Verfahren?

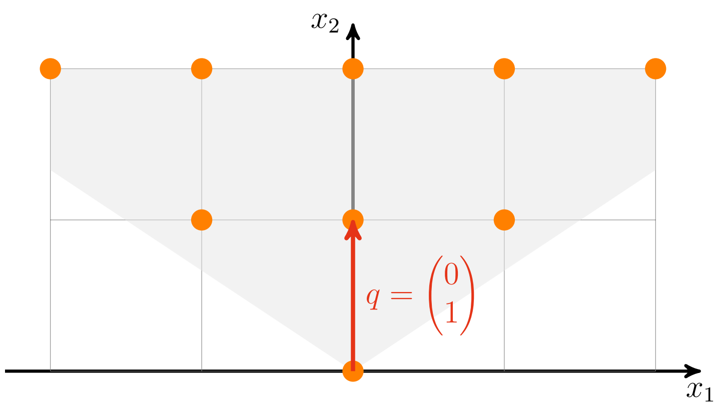

Abbildung 3: Kegel der äußeren Normalen $N_P(x^*)$

3 Rundungsschnitte

Wir betrachten eine Ecke $x^* \in P$ des Polytops P (wir gehen im Folgenden von einem Polytop aus). Wenn x^* nicht ganzzahlig ist, können wir eine Schnittebene finden, die x^* abschneidet, wenn wir als Schnittvektor ein $q \in \mathbb{Z}^n$ aus dem Kegel der äußeren Normalen nehmen. Die Hyperebene $q^T x = q^T x^*$ ist dann eine Stützhyperebene an P in x^*. Da die linke Seite $q^T x$ für alle Ecken von P_I ganzzahlig ist, gilt weiter

$$\text{für alle } x \in P_I\colon \quad q^T x \leq q^T x^* \Leftrightarrow q^T x \leq \left\lfloor q^T x^* \right\rfloor .$$

Also ist

$$q^T x \leq \left\lfloor q^T x^* \right\rfloor$$

eine für P_I gültige Ungleichung. Wenn q jetzt noch so gewählt ist, dass $q^T x^* \notin \mathbb{Z}$, so haben wir eine Schnittebene, die tatsächlich etwas von P abschneidet, also einen eigentlichen Schnitt. Wir brauchen also einen Vektor q mit folgenden Eigenschaften:

- $q \in \mathbb{Z}^n$
- q liegt im Kegel der äußeren Normalen an x^*
- $q^T x^* \notin \mathbb{Z}$

Definition 4
Sei P ein Polyeder, x^* eine Ecke von P und $q \in N_P(x^*)$ so, dass $q \in \mathbb{Z}^n$ und $q^T x^* \notin \mathbb{Z}$. Dann heißt die Ungleichung

$$q^T x \leq \left\lfloor q^T x^* \right\rfloor$$

der *Rundungsschnitt* zum Schnittvektor q.

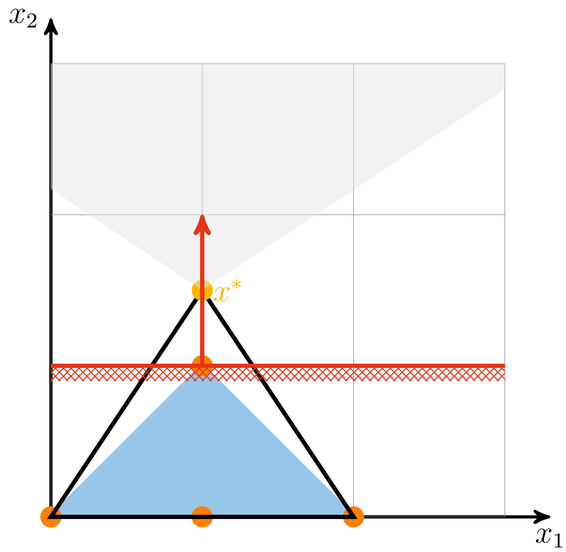

Abbildung 4: Rundungsschnitt

4 Separation von Rundungsschnitten: Gomory-Schnitte

Wir wissen jetzt, wie ein möglicher Schnitt aussehen könnte. Was wir noch nicht wissen, ist, wie wir zu gegebener fraktioneller Ecke $x^* \in P$ einen passenden Rundungsschnitt bestimmen können. Die Bestimmung eines konkreten Schnitts bezeichnet man auch als *Separation* eines Schnitts.

Wie kann man also zu gegebener Ecke x^* konkret ein passendes q bestimmen? Da $q \in N_P(x^*)$ sein soll, wäre es naheliegend, q als Positivkombination der äußeren Normalen darzustellen. Sei dazu $B \subseteq \{1, \dots, n\}$ eine Basis zur Ecke x^* und $N := \{1, \dots, m\} \setminus B$ (die „Nichtbasis"). Wir suchen also ein $y \in \mathbb{R}^m$ mit

- $y \geq 0$ und $y_N = 0$,
- $q^T = y^T A \in \mathbb{Z}^n$,
- $q^T x^* \notin \mathbb{Z} \Rightarrow q^T x^* = y^T A x^* = y_B^T A_B x^* = y_B b_B = y^T b \notin \mathbb{Z}$.

Wir zeigen zunächst, dass wir die Suche noch etwas weiter einschränken können: Es genügt, $y \in [0,1]^m$ zu betrachten. Damit erhalten wir q-Vektoren, die sich in der Form

$$q^T = \sum_{i \in B} y_i a_i^T$$

mit $0 \leq y_i \leq 1$ darstellen lassen, wobei a_i^T die i-te Zeile der Matrix A ist.

Satz 5
Sei x^* eine Ecke des Polytops P und $B \subseteq \{1, \dots, n\}$ eine zugehörige Basis, $N := \{1, \dots, m\} \setminus B$. Weiter sei $q \in N_P(x^*)$ mit $q \in \mathbb{Z}^n$ und $q^T x^* \notin \mathbb{Z}$ gegeben und $y \in \mathbb{R}^m$ mit $y \geq 0$ und $y_N = 0$ so, dass $q^T = y^T A$. Dann gilt für $(q^*)^T := \langle y \rangle^T A$:

- *$q^* \in \mathbb{Z}^n$*
- *q^* liegt im Kegel der äußeren Normalen an x^**

- $(q^*)^T x^* \notin \mathbb{Z}$

Beweis. Sei $N_P(x^*) = \operatorname{pos}\{a_i : i \in B\}$, dann gilt

$$q^T = y^T A = \sum_{i\in B} y_i a_i^T = \sum_{i\in B} \lfloor y_i \rfloor a_i^T + \sum_{i\in B} \langle y_i \rangle a_i^T$$
$$\text{und} \quad (q^*)^T = \sum_{i\in B} \langle y_i \rangle a_i^T.$$

Da $\langle y_i \rangle \geq 0$, ist $q^* \in N_P(x^*)$ und es gilt $q \in \mathbb{Z} \Leftrightarrow q^* \in \mathbb{Z}$.

Wir zeigen weiter, dass $q^T x^* \notin \mathbb{Z} \Leftrightarrow (q^*)^T x^* \notin \mathbb{Z}$. Es gilt

$$q^T x^* = y^T A x^* = y_B^T A_B x^* = y_B^T b_B = y^T b = \lfloor y \rfloor^T b + \langle y \rangle^T b$$
$$(q^*)^T x^* = \langle y \rangle^T A x^* = \langle y_B \rangle^T A_B x^* = \langle y_B \rangle^T b_B = \langle y \rangle^T b.$$

Damit erhalten wir sogar

$$\langle q^T x^* \rangle = \langle \langle y \rangle^T b \rangle = \langle (q^*)^T x^* \rangle,$$

insbesondere folgt die Behauptung. $\square$

Wenn man sich den Beweis noch einmal vergegenwärtigt, fällt auf, dass $y \geq 0$ eigentlich keine Rolle spielt. Entscheidend ist, dass sich q^T in der Form $y_B^T A_B$ darstellen lässt, dass $\sum_{i\in B} \lfloor y_i \rfloor a_i$ ganzzahlig ist, und dass $\langle y \rangle \geq 0$ gilt. Da A_B aus n linear unabhängigen Zeilen besteht, lässt sich aber grundsätzlich jeder Vektor $v \in \mathbb{Z}^n$ als Linearkombination dieser Zeilen, also in der Form $y_B^T A_B$, darstellen. Wir können den Satz daher etwas allgemeiner formulieren:

Satz 6
Sei x^* eine Ecke des Polytops $P = \{x \in \mathbb{R}^n : Ax \leq b\}$, B eine zugehörige Basis und $N := \{1, \ldots, m\} \setminus B$. Für einen Vektor $v \in \mathbb{Z}^n$ mit $v^T x^* \notin \mathbb{Z}$ sei $y \in \mathbb{R}^m$ definiert durch

$$y_B := \langle A_B^{-T} v \rangle,$$
$$y_N := 0$$

und $q^* := A^T y$. Dann gilt:

- $q^* \in \mathbb{Z}$
- $q^* \in N_P(x^*)$
- $(q^*)^T x^* = y^T b \notin \mathbb{Z}$ *und* $\langle (q^*)^T x^* \rangle = \langle v^T x^* \rangle$

Insbesondere ist der Rundungsschnitt

$$(q^*)^T x \leq \left\lfloor (q^*)^T x^* \right\rfloor$$

eine gültige Ungleichung für P_I und ein eigentlicher Schnitt für P (der Punkt x^* wird abgeschnitten).

Definition 7
Der oben definierte Rundungsschnitt heißt *Gomory-Schnitt* zum Vektor v. Für $v = u_i$ spricht man auch vom Gomory-Schnitt zur i-ten Komponente, für $v = c$ vom Gomory-Schnitt zur Zielfunktion.

Algorithm 1: Gomory-Schnittebenen

Input: Ein LP der Form $\max\left\{c^T x : Ax \leq b\right\}$.
Output: Eine eigentliche Schnittebene für das LP oder eine ganzzahlige Optimallösung.

Berechne eine optimale Ecke x^* von $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ sowie eine zugehörige Basis B.
Ist x^* ganzzahlig, brich ab mit Lösung x^*.
Bestimme einen Vektor $v \in \{v, u_1, \ldots, u_n\}$ mit $v^T x^* \notin \mathbb{Z}$.
Setze $y_B := \langle A_B^{-T} v \rangle$ und $y_N := 0$.
Setze $q^* := A^T y$.
Gib die Schnittebene $(q^*)^T x \leq \left\lfloor (q^*)^T x^* \right\rfloor$ aus.

Man kann zeigen, dass sich der Algorithmus so implementieren lässt, dass nach endlichen vielen Schritten tatsächlich P_I bestimmt ist, die Gomory-Schnitte liefern also auch alle „relevanten“ Schnittebenen. Für die Implementierung ist zu beachten, dass der Simplex-Algorithmus mit lexikographischer Pivot-Regel eingesetzt wird, und dass die Schnitte der Reihe nach zur Zielfunktion, dann (wenn kein Zielfunktions-Schnitt möglich ist) zur den Komponenten in aufsteigender Reihenfolge durchgeführt werden. Der Beweis der Endlichkeit baut auf diesen Details auf, ist allerdings ein bisschen technisch. Vom theoretischen Standpunkt ist der Beweis aber sehr wichtig: Damit war zum ersten Mal klar, dass sich ILPs überhaupt durch Schnittebenenverfahren in endlicher Zeit lösen lassen. Für MILPs (in denen nicht alle Variablen ganzzahlig sein müssen) war diese Frage übrigens noch bis vor kurzem ungeklärt (und ist es teilweise immer noch), näheres dazu in [Jör08].

In der Praxis fanden Gomory-Schnitte lange Zeit nicht die Beachtung, die sie verdient haben. Bei der Implementierung kam es häufig zu numerischen Problemen, außerdem benötigt man oft sehr, sehr viele Schnitte, bis ein ganzzahliges Optimum bestimmt ist. Die numerischen Schwierigkeiten hat man in den letzten Jahren allerdings recht gut in den Griff bekommen. Hier sei nur darauf hingewiesen, dass man sich bei der Implementierung tunlichst um solche Fragen kümmern sollte! Häufig kombiniert man das Gomory-Verfahren mit anderen Ideen, um ILPs zu lösen. Die meisten LP-Solver benutzen zu Beginn eine Anzahl von Gomory-Schnitten und weiteren, problemspezifischen Schnitten, um das Polytop (und damit den Suchraum für folgende Verfahren) deutlich zu verkleinern. Oft wird dann mit einem Branch and Bound-Algorithmus auf dem kleineren Polytop weitergearbeitet, häufig kombiniert man das auch mit problemspezifischen Schnitten für die Teilprobleme (dann spricht man von *Branch and Cut*-Ansätzen).

Außerdem spielen in praktischen Problemen spezielle, auf das jeweilige Problem angepasste Schnitte eine wichtige Rolle. Ein allgemeines Verfahren wie der Gomory-Algorithmus kann zwar sehr hilfreich sein (und funktioniert immer), effektiver kann man aber schneiden, wenn man die bekannte Struktur eines speziellen Problems ausnutzt. Die Idee ist dabei meistens, etwas über die Geometrie des Polytops P_I in Erfahrung zu bringen (z. B. das Aussehen einiger Facetten zu beschreiben) und dieses Wissen in Schnittebenen zu verarbeiten. Diese sind dann häufig sehr effektiv, manchmal aber schwer zu separieren. Oft fehlt es auch einfach an Wissen, in vielen Fällen kennt man zwar einige Klassen von Facetten von P_I, aber leider nicht alle (oder man weiß nicht einmal, ob es noch weitere gibt). Beispiele für problemspezifische Schnitte und die damit verbundenen Fragestellungen und Herausforderungen werden wir in der nächsten

Stunde noch kennenlernen.

Referenzen

[Jör08] Markus Jörg. „k-disjunctive cuts and cutting plane algorithms for general mixed integer linear programs“. Dissertation. München: Technische Universität München, 2008.