Die Größe eines Linearen Programms

Sei $k \in \mathbf{Z}$, $x \in Z^m$, $M \in \mathbf{Z}^{m \times n}$.

Wir definieren die *Größe einer Kodierung* als

$$size(k) := 1 + \lceil \log_2(|k| + 1) \rceil$$

$$size(x) := \sum_{i=1}^{m} size(x_i)$$

$$size(M) := \sum_{i=1}^{m} \sum_{j=1}^{n} size(M_{ij})$$

Die Größe eines Linearen Programms

Sei $A \in \mathbf{Z}^{m \times n}$, $b \in \mathbf{Z}^m$, $c \in \mathbf{Z}^n$.

$$\begin{aligned} \text{Minimiere } & c^T x \\ \text{unter } & Ax = b \\ & x \geq 0 \end{aligned} \qquad (LP)$$

Größe dieses linearen Programms:

$$\mathit{size}(LP) := \mathit{size}(A) + \mathit{size}(b) + \mathit{size}(c)$$

(Durch Skalieren mit dem lcp läßt sich jedes LP in diese Form bringen.)

Die Größe eines Linearen Programms

Statt mit $size(LP)$ rechnet es sich oft günstiger mit einer Größe L, die so definiert ist:

$$L := size(det_{\max}) + size(b_{\max}) + size(c_{\max}) + m + n,$$

wobei

$$\begin{aligned} det_{\max} &= \max_{A'} |det A'| \quad (A' \text{ Untermatrix von } A) \\ b_{\max} &= \max_{i} |b_i| \\ c_{\max} &= \max_{i} |c_i| \end{aligned}$$

Behauptung:

$$L < size(LP)$$

$\Rightarrow$ Ist die Laufzeit polynomiell in L, dann auch in $size(LP)$.

Größe der Ausgabe

Sei x eine Ecke von $\{ x \mid Ax = b,\, x \geq 0 \}$.

Dann ist

$$x^T = \begin{pmatrix} \frac{p_1}{q} & \frac{p_2}{q} & \cdots & \frac{p_n}{q} \end{pmatrix},$$

mit

$$p_i, q \in \mathbf{N},\ 0 \leq p_i < 2^L,\ 1 \leq q < 2^L.$$

Die Ausgabe ist also polynomiell repräsentierbar.

Interior Point Methods

Theorem

Seien x_1 und x_2 Ecken von $\{ x \mid Ax = b,\ x \geq 0 \}$.

Falls $c^T x_1 \neq c^T x_2$, dann ist $|c^T x_1 = c^T x_2| > 2^{-2L}$.

Korollar

Sei $z = \min\{ c^T x \mid Ax = b,\ x \geq 0 \}$ und sei x eine zulässige Lösung mit $c^T x \leq z + 2^{-2L}$.

Dann ist jede Ecke x' mit $c^T x' \leq c^T x$ eine optimale Lösung.

$\Rightarrow$ Es genügt nahe an eine Lösung heranzukommen und man muß nur mit einer beschränkten Genauigkeit rechnen.

Interior Point Methods

Wir lösen gleichzeitig dieses primale und sein duales Problem:

$$\begin{aligned} \text{Minimiere } z &= c^T x \\ \text{unter } Ax &= b \\ x &\geq 0 \end{aligned} \qquad (P)$$

$$\begin{aligned} \text{Maximiere } w &= b^T y \\ \text{unter } A^T y + s &= c \\ s &\geq 0 \end{aligned} \qquad (D)$$

Der Algorithmus arbeitet mit einer primalen Lösung $\bar{x} > 0$ und einer Belegung $\bar{s} > 0$ von s, zu welcher es ein $\bar{y}$ mit $A^T \bar{y} + \bar{s} = c$ gibt.

Interior Point Methods

Eine Grundidee des Algorithmus ist es im Inneren des Polyeders zu bleiben. Keine Komponente von $\bar{x} > 0$ und $\bar{s} > 0$ soll also klein werden.

Folgende Abbildung transformiert das Problem in einen Bildraum:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \mapsto x' = \begin{pmatrix} x_1/\bar{x}_1 \\ x_2/\bar{x}_n \\ \vdots \\ x_n/\bar{x}_n \end{pmatrix}$$

Diese Abbildung bildet $\bar{x}$ auf e (den Einsvektor) ab.

Skalierung

Wir können diese Abbildung auch so beschreiben:

$$x \mapsto x' = \bar{X}^{-1}x$$

mit

$$\bar{X} = \begin{pmatrix} \bar{x}_1 & 0 & 0 & \cdots & 0 \\ 0 & \bar{x}_2 & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & \bar{x}_{n-1} & 0 \\ 0 & 0 & \cdots & 0 & \bar{x}_{n-1} \end{pmatrix}.$$

Skalierung

Im Bildraum sehen die Probleme so aus:

$$\begin{aligned} \text{Minimiere } z &= \bar{c}^T x' \\ \text{unter } \bar{A}x' &= b \\ x' &\geq 0 \end{aligned} \tag{P}$$

$$\begin{aligned} \text{Maximiere } w &= b^T y \\ \text{unter } \bar{A}^T y + s' &= \bar{c} \\ s' &\geq 0 \end{aligned} \tag{D}$$

Dabei ist $\bar{A} = A\bar{X}$ und $\bar{c} = \bar{X}c$.

Die Dualitätslücke

Es stellt sich heraus, daß

$$s' = \bar{X}s = \begin{pmatrix} s_1\bar{x}_1 \\ s_2\bar{x}_2 \\ \vdots \\ s_n\bar{x}_n \end{pmatrix}.$$

Die *Dualitätslücke* $c^T\bar{x} - b^T\bar{y} = \bar{x}^T\bar{s}$ ändert sich durch Skalierung nicht: Es gilt ja $x^Ts = {x'}^Ts'$.

(Sobald die Dualitätslücke sehr klein ist, kann der Algorithmus terminieren.)

Die Potentialfunktion

Definition

$$G(x, s) = (n + \sqrt{n}) \ln(x^T s) - \sum_{j=1}^{n} \ln(x_j s_j)$$

1. Die Potentialfunktion ist unter Skalierung invariant
2. Man kann leicht einen zulässigen Startwert $\bar{x} > 0$, $\bar{s} > 0$ finden, mit $G(\bar{x}, \bar{s}) = O(\sqrt{n}L)$
3. Ist $G(\bar{x}, \bar{s}) \leq -2\sqrt{n}L$ dann sind wir nah am Optimum und können stoppen.
4. Wir können stets einen *Schritt* $\bar{x}, \bar{s} \mapsto \tilde{x}, \tilde{s}$ machen, so daß $G(\tilde{x}, \tilde{s}) - G(\bar{x}, \bar{s}) \leq -7/120$.

$\Rightarrow$ Nach $O(\sqrt{n}L)$ Iterationen sind wir fertig.

Der primale Schritt

Wir machen entweder einen *primalen* oder einen *dualen Schritt*.

Der primale Schritt ändert nur $\bar{x}$ und sieht so aus:

Zunächst gehen wir in den Bildraum (unser Punkt ist jetzt (e, s')).

Jetzt möchten wir in eine Richtung gehen, in der die Potentialfunktion um möglichst viel kleiner wird.

Dazu betrachten wir den Gradienten von $G(x, s)$ am Punkt (e, s'):

$$
\begin{aligned}
g &= \nabla_x G(x, s)\big|_{(e,s')} \\
&= \frac{n + \sqrt{n}}{x^T s} s - \left.\begin{pmatrix} 1/x_1 \\ \vdots \\ 1/x_n \end{pmatrix}\right|_{(e,s')} \\
&= \frac{n + \sqrt{n}}{e^T s'} s' - e
\end{aligned}
$$

Wir können aber nicht in Richtung $-g$ gehen, da wir den zulässigen Bereich verlassen könnten.

Sei d die Projektion von g auf $\{\, x \mid \bar{A}x = 0 \,\}$.

Wir können in Richtung $-d$ gehen!

Der primale Schritt

Der Vektor d läßt sich so ausdrücken:

$$d = (I - \bar{A}(\bar{A}\bar{A}^T)^{-1}\bar{A})g$$

Der primale Schritt geht zu

$$\tilde{x} = e - \frac{1}{4|d|}d, \ \ \tilde{s} = s'.$$

(Danach muß wieder zurückskaliert werden.)

Um zu garantieren, daß die Potentialfunktion genügend abnimmt (um 7/120), muß $|d| = \sqrt{d^T d} \geq 0.4$ gelten. Nur wenn dies der Fall ist, machen wir einen primalen Schritt.

Der duale Schritt

Falls $|d| < 0.4$ ist, machen wir einen *dualen Schritt*, der nur s' verändert. Der Gradient ist jetzt:

$$h = \nabla_s G(x, s)|_{(e,s')} = \frac{n + \sqrt{n}}{e^T s'} e - \begin{pmatrix} 1/s'_1 \\ \vdots \\ 1/s'_n \end{pmatrix}$$

Wir brauchen: Es muß ein y mit $\bar{A}^T y + \tilde{s} = \bar{c}$ geben.

Es geht, wenn wir den Schritt so wählen:

$$\tilde{s} = \frac{e^T s'}{n + \sqrt{n}}(d + e), \ \tilde{x} = x' = e$$

Man kann zeigen, daß $\tilde{s} > 0$ und daß die Potentialfunktion um mindestens $1/6$ abnimmt.

Ganzzahliges Programmieren (ILP)

Sei A eine rationale Matrix und b, c rationale Vektoren. Das *ILP* ist es, folgende Optimierungsaufgaben zu lösen:

$$\begin{aligned} \text{Maximiere } & c^T x \\ \text{unter } & Ax \leq b \\ & x \text{ ganzzahlig} \end{aligned}$$

oder

$$\begin{aligned} \text{Maximiere } & c^T x \\ \text{unter } & Ax = b \\ & x \geq 0 \\ & x \text{ ganzzahlig} \end{aligned}$$

Eine einfache Abschätzung

Dualität liefert:

$$\begin{aligned} &\max\{\, c^T x \mid Ax \le b,\ x \text{ ganzzahlig}\,\} \\ \le &\min\{\, b^t y \mid A^T y = c,\ y \ge 0,\ y \text{ ganzzahlig}\,\} \end{aligned}$$

Im Gegensatz zu linearem Programmieren, ist die Ungleichung bei ILP normalerweise echt. Durch die *LP-Relaxation* erhalten wir eine Abschätzung, falls die Voraussetzung für starke Dualität vorliegt:

$$\begin{aligned} &\max\{\, c^T x \mid Ax \le b,\ x \text{ ganzzahlig}\,\} \\ \le &\max\{\, c^T x \mid Ax \le b\,\} = \min\{\, b^t y \mid A^T y = c,\ y \ge 0\,\} \\ \le &\min\{\, b^t y \mid A^T y = c,\ y \ge 0,\ y \text{ ganzzahlig}\,\} \end{aligned}$$

Beispiel

Sei $A = (2)$, $b = (1)$, $c = (1)$.

$$
\begin{aligned}
& \max\{\, c^T x \mid Ax \leq b,\ x \text{ ganzzahlig} \,\} && = 0 \\
\leq\ & \max\{\, c^T x \mid Ax \leq b \,\} && = 1/2 \\
=\ & \min\{\, b^t y \mid A^T y = c,\ y \geq 0 \,\} && = 1/2 \\
\leq\ & \min\{\, b^t y \mid A^T y = c,\ y \geq 0,\ y \text{ ganzzahlig} \,\} && = \text{existiert nicht}
\end{aligned}
$$

Beispiel

Sei $A = (2)$, $b = (1)$, $c = (1)$.

$$\begin{aligned}
&\max\{\, c^T x \mid Ax \le b,\ x \ge 0\, x \text{ ganzzahlig}\,\} && = 0\\
\le\, &\max\{\, c^T x \mid Ax \le b,\ x \ge 0\,\} && = 1/2\\
=\, &\min\{\, b^t y \mid A^T y \ge c,\ y \ge 0\,\} && = 1/2\\
\le\, &\min\{\, b^t y \mid A^T y \ge c,\ y \ge 0,\ y \text{ ganzzahlig}\,\} && = 1
\end{aligned}$$

Komplexität von ILP

Theorem

ILP ist NP-vollständig

Beweis

Reduktion von 3SAT auf ILP.

Gegeben ist eine Instanz F von 3SAT bestehend aus n Variablen $x_1, \ldots x_n$ und m Klauseln der Form $\{l_1, l_2, l_3\}$, wobei $l_i \in \{x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n\}$.

Wir konstruieren aus F ein ILP I mit:

F hat eine erfüllende Belegung $\iff$ der optimale Wert von I ist 0

Unser ILP I hat die Literale von F als Variablen und sieht so aus:

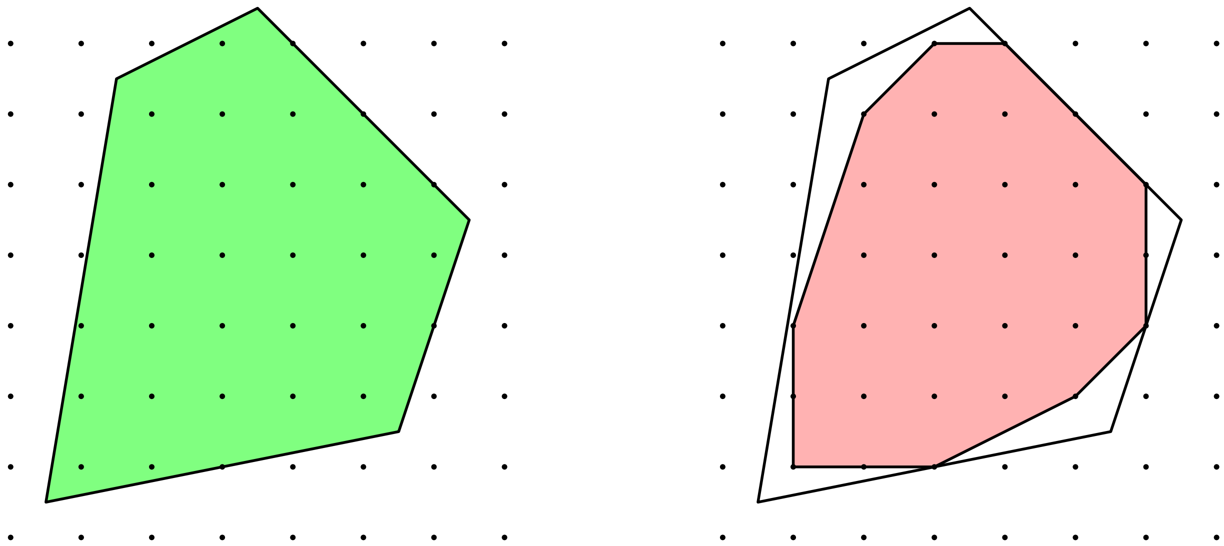
$$\text{Maximiere } 0$$

unter folgenden Bedingungen:

$$\begin{aligned} l_1 + l_2 + l_3 \geq 1 &\text{ für alle Klauseln } \{l_1, l_2, l_3\} \text{ in } F \\ 0 \leq x_i, \bar{x}_i \leq 1 &\text{ für } i = 1, \ldots, n \\ x_i + \bar{x}_i = 1 &\text{ für } i = 1, \ldots, n \\ x_1, \ldots, x_n &\text{ ganzzahlig} \end{aligned}$$

Es gibt genau dann eine Lösung, wenn F erfüllbar ist. Die Konstruktion ist in polynomieller Zeit durchführbar. $\square$

Die ganzzahlige Hülle



Die *ganzzahlige Hülle* P_I eines Polyeders P ist die konvexe Hülle aller ganzzahliger Punkte in P.

Falls alle Ecken von P ganzzahlig sind, das ist $P = P_I$ und das ILP kann durch lineares Programmieren gelöst werden.

Ein einfaches Lösungsverfahren

Wir wollen $\max\{\, c^T x \mid Ax \leq b,\ x \text{ ganzzahlig} \,\}$ bestimmen.

Wir starten mit $\Pi_1 = P$, $P = \{\, x \mid Ax \leq b \,\}$.

Im Schritt k haben wir eine Menge $\Pi_k = \{P_1, \ldots, P_k\}$ von Polyedern mit

1. $P_1, \ldots, P_k$ sind paarweise disjunkt (repräsentiert durch lineare Ungleichungen),
2. jeder ganzzahlige Punkt in P ist in $P_1 \cup \cdots \cup P_k$ enthalten.

Ein Schritt sieht so aus (Eingabe Π_k):

Sei $\mu_j = \max\{\, c^T x \mid x \in P_j \,\}$ und j^* so, daß μ_{j^*} maximal unter allen μ_j ist.

Sei x^* so, daß $\mu_{j^*} = c^T x^*$ (diese Werte lassen sich durch lineares Programmieren finden).

Falls x^* ganzzahlig ist, dann ist x^* eine optimale Lösung.

Andernfalls, sei x_i^* eine nicht-ganzzahlige Komponente. Wir definieren

$$Q_1 = \{\, x \mid P_{j^*} \mid x_i \geq \lceil x_i^* \rceil \,\}$$
$$Q_2 = \{\, x \mid P_{j^*} \mid x_i \leq \lfloor x_i^* \rfloor \,\}$$

Wir setzen

$$\Pi_{k+1} := \{P_1, \ldots, P_{j^*-1}, Q_1, Q_2, P_{j^*+1}, \ldots, P_k\}.$$

(Falls $P_i = \emptyset$ für alle i, dann gibt es keine Lösung.)

Beispiel