Effiziente Algorithmen (SS2015)

Kapitel 4
Lineare Programme 1

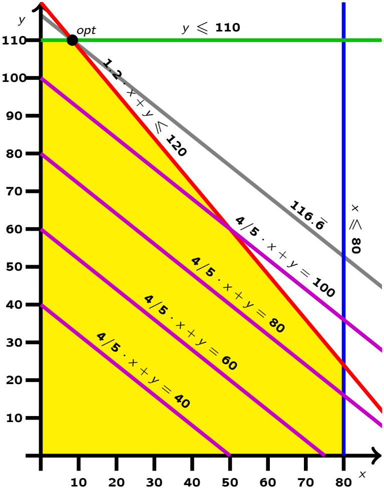
Walter Unger

Lehrstuhl für Informatik 1

14:05 Uhr, den 19. Dezember 2018

Inhalt I

Einfaches Beispiel



- Wir betrachten ein System von linearen Ungleichungen.
- Dabei ist eine "Zielfunktion" zu optimieren.
- Beispiel (Brotrezept):
  - x kg Weizenmehl
  - y kg Roggenmehl
  - Maximal 80 kg Weizenmehl.
  - Maximal 110 kg Roggenmehl.
  - Mischungsverhältnis: $1.2 \cdot x + y$.
  - Maximale Brotmenge: $1.2 \cdot x + y \leqslant 120$.
  - Zu optimieren: $f(x, y) = 4/5 \cdot x + y$.

Beispiel: Flussproblem

- Flussproblem:
  - Gegeben $G = (V, E, s, t, c)$ mit $c : E \mapsto \mathbb{N}$.
  - Maximiere den Fluss.
- als lineares Programm:
  - Variablen x_e für $e \in E$.
  - Maximiere

$$\sum_{e \in N_{out}(s) \in E} x_e.$$

  - unter Einhaltung der Bedingungen:
    - Für jeden Knoten $v \in V \setminus \{s, t\}$: $\sum_{e \in N_{in}(v)} x_e = \sum_{e \in N_{out}(v)} x_e$,
    - $\forall e \in E : x_e \leqslant c_e$, und
    - $\forall e \in E : x_e \geqslant 0$.

Beispiel: Relaxiertes Rucksackproblem

- Relaxiertes Rucksackproblem:
  - gegeben d teilbare Objekte mit Gewichten g_i und
  - die Gewichtsschranke G des Rucksacks.
  - Und v_i sei der Nutzen für $1 \leqslant i \leqslant d$.
  - Sei x_i der Anteil von Objekt i.
  - Fülle den Rucksack. Dabei soll der Nutzen maximal sein.
- Als lineares Programm:
  - Maximiere
$$\sum_{i=1}^{d} v_i \cdot x_i.$$
  - unter den Nebenbedingungen:
    - $\sum_{i=1}^{d} g_i \cdot x_i \leqslant G$,
    - $\forall i : 1 \leqslant i \leqslant d : x_i \leqslant 1$, und
    - $\forall i : 1 \leqslant i \leqslant d : x_i \geqslant 0$.

Beispiel: Routing und Wellenlängenzuweisung

- Routing und Wellenlängenzuweisung: bestimme die Wellenlängen in einem optischen Netzwerk. In dem Netzwerk gibt es Kommunikationsanfragen für Paare von Knoten.
  - N_E Endknoten.
  - N_R Router mit Konverter.
  - $n = N_E + N_R$.
  - Namen der Knoten: N_i $(1 \leqslant i \leqslant n)$.
  - m Anzahl der Lichtwege.
  - E: $(i, j) \in E \iff$ Kante von N_i nach N_j.
  - $src(k)$: ist Startknoten der k-ten Anfrage.
  - $dst(k)$: ist Endknoten der k-ten Anfrage.
  - Ω_{max}: Congestion des Netzwerks.
  - $X_{ij}^k \in \{0, 1\}$ mit:

$$X_{ij}^k = \begin{cases} 1 & \text{der } k\text{-te Weg nutzt Kante } (i,j) \in E \\ 0 & \text{sonst} \end{cases}$$

  - Wegen $X_{ij}^k \in \{0, 1\}$ ist dies hier ein "Integer Linerar Programm".

Routing und Wellenlängenzuweisung als ILP

$n = N_E + N_R$
$src(k) \longleftrightarrow dst(k)$: $(\mathbf{1} \leqslant k \leqslant m)$
X_{ij}^k : k**-te Weg nutzt Kante** (i, j)

- Minimiere Zielfunktion Ω_{max}.
- $\sum_{k=1}^{m} X_{ij}^k \leqslant \Omega_{max}, \forall (i,j) \in E$.
- Für alle $k : 1 \leqslant k \leqslant m$ und alle $i : 1 \leqslant i \leqslant n$:

$$\sum_{j:(i,j)\in E} X_{ij}^k - \sum_{j:(j,i)\in E} X_{ji}^k = \begin{cases} 1 & \text{falls } src(k) = i \\ -1 & \text{falls } dst(k) = i \\ 0 & \text{sonst} \end{cases}$$

- Dies ist eine korrekte Formulierung des Routen- und Wellenlängenzuweisungsproblems als ILP.
- Komplexität:
  - $m \cdot |E|$ Variablen der Form X_{ij}^k.
  - Eine Variable Ω_{max}.
  - Nebenbedingungen: $|E| + n \cdot m$.
  - Schon für relativ kleine Netzwerke zu aufwendig.

Routen- und Wellenlängenzuweisung

- N_E Endknoten; N_R Router ohne Konverter.
- $n = N_E + N_R$; Namen der Knoten: N_i $(1 \leqslant i \leqslant n)$.
- m Anzahl der Lichtwege; n_{ch} Anzahl der Wellenlängen.
- E: $(i,j) \in E \iff$ Kante von N_i nach N_j.
- $src(k)$: ist Startknoten der k-ten Anfrage.
- $dst(k)$: ist Endknoten der k-ten Anfrage.
- Ω_{max}: Congestion des Netzwerks.
- $X_{ij}^{wk} \in \{0,1\}$ mit:

$$X_{ij}^{wk} = \begin{cases} 1 & \text{der } k\text{-te Weg nutzt } (i,j) \in E \text{ und W.länge } w \\ 0 & \text{sonst} \end{cases}$$

- $Y_w^k \in \{0,1\}$ mit:

$$Y_w^k = \begin{cases} 1 & \text{der } k\text{-te Weg nutzt Wellenlänge } w \\ 0 & \text{sonst} \end{cases}$$

ILP:

$n = N_E + N_R$
$src(k) \longleftrightarrow dst(k) : (1 \leqslant k \leqslant m)$
X_{ij}^k : k-**te Weg nutzt Kante** (i, j)
Y_w^k : k-**te Weg nutzt Wellenlänge** w

- Minimiere Zielfunktion Ω_{max}:

$$\sum_{w=1}^{n_{ch}} \sum_{k=1}^{m} X_{ij}^{wk} \leqslant \Omega_{max}, \forall (i,j) \in E$$

- Für alle $k, w : 1 \leqslant k \leqslant m, 1 \leqslant w \leqslant n_{ch}$ und alle $i : 1 \leqslant i \leqslant n$:

$$\sum_{j:(i,j)\in E} X_{ij}^{wk} - \sum_{j:(j,i)\in E} X_{ji}^{wk} = \begin{cases} Y_w^k & \text{falls } src(k) = i \\ -Y_w^k & \text{falls } dst(k) = i \\ 0 & \text{sonst} \end{cases}$$

- Für alle $k : 1 \leqslant k \leqslant m$:

$$\sum_{w=1}^{n_{ch}} Y_w^k = 1$$

- Für alle $w : 1 \leqslant w \leqslant n_{ch}$ und alle $(i,j) \in E$:

$$\sum_{k=1}^{m} X_{ij}^{wk} \leqslant 1$$

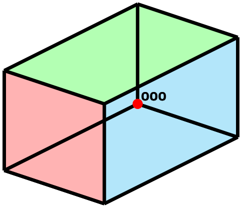
Kanonische Form eines LP

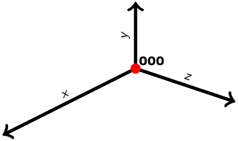
- Ein LP ist in kanonischer Form, falls:
  - Es gibt d Variablen $x_i \in \mathbb{R}$ $(1 \leqslant i \leqslant d)$,
  - Werte $c_j \in \mathbb{R}$ für $1 \leqslant j \leqslant d$,
  - Werte $b_i \in \mathbb{R}$ für $1 \leqslant i \leqslant m$ und
  - Werte $a_{i,j} \in \mathbb{R}$ für $1 \leqslant j \leqslant d$ und $1 \leqslant i \leqslant m$.
  - Gesucht ist eine Belegung der Variablen $x_i \in \mathbb{R}$ mit:
    - Maximiere Zielfunktion $\sum_{j=1}^{d} c_j \cdot x_j$
    - unter den Nebenbedingungen:
    - $\sum_{j=1}^{d} a_{ij} \cdot x_j \leqslant b_j$ für $i \in \{1, 2, \ldots, m\}$ und
    - $x_j \geqslant 0$ für $j \in \{1, 2, \ldots, d\}$.
- Setze:
  - $x = (x_j)$, $c = (c_j)$, $b = (b_i)$ und $A = (a_{ij})$.
- Dann ist kurzgefasst die kanonische Form:
  - Maximiere $c^T \cdot x$ unter den Nebenbedingungen:
  - $A \cdot x \leqslant b$ und $x \geqslant 0$.

Umformungen zur kanonischen Form

- Minimierungsproblem in ein Maximierungsproblem:
  - $c^T \cdot x$ wird zu $-c^T \cdot x$.
- Eine Gleichung $a^T \cdot x = b$ wird ersetzt durch
  - $a^T \cdot x \leqslant b$ und
  - $a^T \cdot x \geqslant b$.
- Eine Gleichung $a^T \cdot x \geqslant b$ wird ersetzt durch
  - $-a^T \cdot x \leqslant -b$.
- Eine möglicherweise negative Variable $x \in \mathbb{R}$ wird ersetzt durch:
  - $x' - x''$ und den Nebenbedingungen:
  - $x' \geqslant 0$ und
  - $x'' \geqslant 0$.

Geometrische Interpretation





$$0 \leqslant x \leqslant t$$
$$0 \leqslant y \leqslant t$$
$$0 \leqslant z \leqslant t$$

- Eine Variablenbelegung $x = (x_1, x_2, \ldots, x_d)$ entspricht einem Punkt im d-dimensionalen Raum.
- Eine Nebenbedingung $a_i \cdot x \leqslant b_i$ definiert einen Halbraum.
- Die Grenze des Halbraums ist die Hyperebene $a_i \cdot x = b_i$.
- Der Schnitt aller Halbräume ist der Raum der zulässigen Lösungen.
- Ein LP wird als zulässig bezeichnet, wenn des zulässige Lösungen gibt.
- Schnittmengen von Halbräumen bilden ein Polyhedron.
- Damit bilden die zulässigen Lösungen ein Polyhedron.

Konvexität

P **Polyhedron**

Lemma

Ein Polyhedron P ist konvex.

Beweis:

- Konvex: $\forall a, b \in P : l(a, b) \in P$ mit:
- $l(a, b) = \{\lambda a + (1 - \lambda)b \mid 0 \leqslant \lambda \leqslant 1\}$.
- Ein Halbraum ist konvex.
- Ein Polyhedron ist der Schnitt von Halbräumen.
- Zeige: Der Schnitt von konvexen Mengen ist konvex.
- D.h. aus A, B konvex folgt $A \cap B$ konvex.
- Damit gilt: $\forall a, b \in A : l(a, b) \in A$ und
- weiter $\forall a, b \in B : l(a, b) \in B$.
- Es folgt: $\forall a, b \in A \cap B : l(a, b) \in A \cap B$.

Lokales und globales Optimum

P **Polyhedron**

Lemma

Sei $z, x \in P$ und $c^T z > c^T x$.
Dann existiert für $\varepsilon > 0$ ein $y \in P$ mit: $||x - y|| \leqslant \varepsilon$ und $c^T y > c^T x$.

Beweis:

- P ist konvex, gilt $l(x, z) \in P$.
- Wähle $y \in l(x, z)$ mit $x \neq y$ und $||x - y|| \leqslant \varepsilon$.
- Nach Definition von l gibt $\lambda > 0$ mit: $y = \lambda x + (1 - \lambda)z$.
- Es folgt:

$$\begin{aligned} c^T y &= c^T(\lambda x + (1-\lambda)z) \\ &= \lambda c^T x + (1-\lambda)c^T z \\ &> \lambda c^T x + (1-\lambda)c^T x \\ &= c^T x \end{aligned}$$

Lemma

Ein lokales Optimum ist auch ein globales Optimum.

Unterräume

P **Polyhedron**

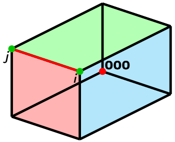
- Eine Hyperebene wird beschrieben durch:

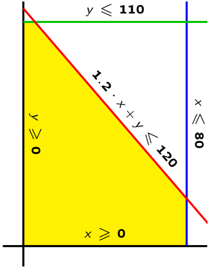
$$\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \ldots + \alpha_d x_d = \beta.$$

  - Es sind $d-1$ Variablen frei wählbar.
  - Der Wert der d-ten Variable ist dann festgelegt.
  - Der durch die Hyperebene beschriebene affine Unterraum ist ein Unterraum der Dimension $d-1$.
  - Ein Unterraum, der als Schnittmenge von k linear unabhängigen Hyperräumen beschrieben wird, hat Dimension $d-k$.
- Falls mehr als d Nebenbedingungen (Hyperebenen) sich in einem Punkt treffen, so ist das LP degeneriert.
- Ein degeneriertes LP kann in ein nicht-degeneriertes LP umgeformt werden, ohne die Form (Zusammensetzung) der Lösung signifikant zu verändern.

Oberfläche eines Polyhedrons

P **Polyhedron**





- Die Oberfläche eines Polyhedrons besteht aus Facetten.
  - Sei P ein Polyhedron und H eine Hyperebene und sei P komplett auf einer Seite der Hyperebene H.
  - Falls $f = P \cap H$ nicht leer ist, so
  - ist f eine Facette von P.
- Eine Facette der Dimension $d-1$ heißt Face.
- Eine Kante entspricht dem Schnitt von $d-1$ Hyperebenen (Facette der Dimension 1).
- Ein Knoten entspricht dem Schnitt von d Hyperebenen (Facette der Dimension 0).
- Zwei Knoten sind benachbart, wenn sie durch eine Kante verbunden sind.
- Falls P unbeschränkt ist, so kann es unbeschränkte Kanten geben.
- Solche Kanten haben nur einen oder keinen Endpunkt.

Zielfunktion

- Die Zielfunktion $c^T x$ (Vektor) gibt eine Richtung in $\mathbb{R}^d$ vor.
- Falls die Nebenbedingungen den Zielwert nach oben beschränken, so wird das LP als beschränkt bezeichnet.
  - Falls der Zielwert nicht beschränkt ist, so heißt das LP unbeschränkt.
- Das Polyhedron muss dabei nur in der Richtung von $c^T x$ beschränkt sein.
- Ist das Polyhedron in alle Richtungen beschränkt (in einer Kugel enthalten), so wird es als Polytop bezeichnet.

Geometrische Bestimmung des Optimums

- Betrachte beschränktes LP in kanonischer Form mit Lösungspolyhedron P und Zielfunktion $c^T x$,
- Sei $\mathcal{H}$ eine zum Vektor c orthogonale Hyperebene.
- Damit gibt es $t \in \mathbb{R}$ mit: $\mathcal{H} = \{x \in \mathbb{R}^d \mid c^T \cdot x = t\}$.
- Setze $\mathcal{H}_t = \{x \in \mathbb{R}^d \mid c^T \cdot x = t\}$.
- Sei $\mathcal{H}$ so gewählt, dass $P \cap \mathcal{H} \neq \emptyset$ gilt.
- Wähle z maximal mit $P \cap \mathcal{H}_z \neq \emptyset$.
- Ein beliebiger Punkt $x^* \in P \cap \mathcal{H}_z$ ist eine optimale Lösung des LPs.
- Beobachtungen:
  - $P \cap \mathcal{H}_z$ ist eine Facette f von P.
  - Falls f nicht in allen Richtungen unbeschränkt ist, so gibt es mindestens einen optimalen Knoten.

Idee des Algorithmus von Seidel

Wenn die Dimension klein ist, oder die Anzahl der Nebenbedingungen klein ist, dann ist das Lösen von einem Ungleichungssystem einfach. Daher wird beim Algorithmus von Seidel eine kleine Menge von Nebenbedingungen zufällig ausgewählt und für dieses vereinfachte System dann eine Lösung bestimmt.

Danach werden zufällig die bisher nicht betrachteten Nebenbedingungen hinzugefügt. Es gibt dann zwei Möglichkeiten:

- Die bisher berechnete Lösung ist zufällig weiterhin gültig und damit weiterhin optimal.
- Die bisher berechnete Lösung ist nicht mehr gültig. Damit muss die neue optimal Lösung auf dem Schnitt mit dem bisherigen Polyhedron und der durch die hinzugefügte Nebenbedingung bestimmte Hyperebene liegen. Nun wird rekursiv auf diesem Schnitt eine neue optimal Lösung bestimmt.

Zu beachten ist, daß das im Weitern beschriebene Verfahren rekursiv ist, d.h. es fängt mit dem Löschen von Nebenbedingungen an, bis ein einfaches kleines System entsteht. Nachdem da eine Lösung gefunden ist, werden die Nebenbedingungen wieder einzeln eingefügt.

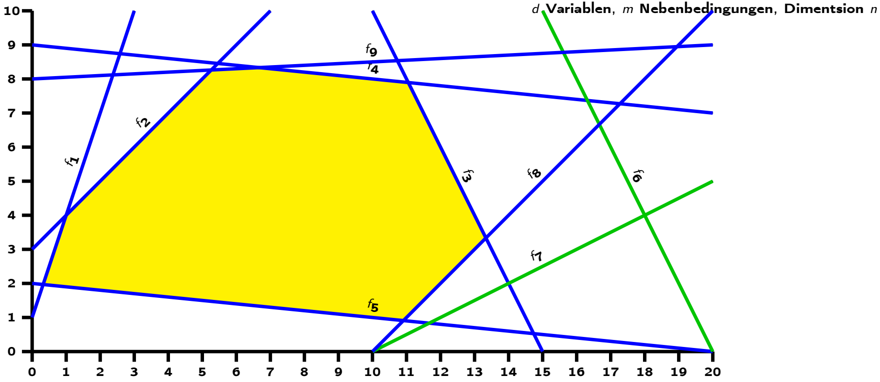
Die Idee

- Kann eine zufällige Auswahl helfen?
- Können effiziente Algorithmen noch schneller gemacht werden?
- Kann bei schweren Problemen mit ausreichender Wahrscheinlichkeit ein gutes Ergebnis erreicht werden?

Einleitung

- Bestimme Extrempunkt eines Polyhedrons bezüglich einer Zielfunktion.
- Eingabe:
  - Gegeben sind d Variablen.
  - Gegeben sind m lineare Ungleichungen (Nebenbedingungen).
  - Gegeben ist eine lineare Zielfunktion f über die d Variablen.
- Gesucht:
  - Maximiere f unter Berücksichtigung aller Nebenbedingungen.
- Das ist ein LP (lineares Programm)
- Obere Schranke für Laufzeit: $O(\binom{m}{d}) = O(n^d)$.
  - Untersuche für jede d-elementige Teilmenge der m Ungleichungen den jeweiligen Schnittpunkt (Basislösung).
- Hier nun Algorithmus mit erwarteter linearer Laufzeit (d.h. linear in m).

Beispiel



Maximiere y unter den Nebenbedingungen $0 \leqslant x \leqslant 20$, $0 \leqslant y \leqslant 10$, und:

$f_1 : y \leqslant 3 \cdot x + 1$	$f_4 : y \leqslant -0.9 \cdot x + 9$	$f_7 : y \geqslant 0.5 \cdot x - 5$
$f_2 : y \leqslant 1 \cdot x + 2$	$f_5 : y \geqslant -0.9 \cdot x + 2$	$f_8 : y \geqslant 1 \cdot x - 10$
$f_3 : y \leqslant -2 \cdot x + 30$	$f_6 : y \leqslant -2 \cdot x + 40$	$f_9 : y \leqslant 1.05 \cdot x + 8$

Vorgaben

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Variablen: $x_1, x_2, \ldots, x_d$ mit:
  - $x = (x_1, x_2, \ldots, x_d)^T$ und
- Nebenbedingungen: $A \cdot x \leqslant b$ mit:
  - A ist eine $m \times d$ Matrix und
  - $b = (b_1, b_2, \ldots, b_d)^T$.
- Zielfunktion $f(x) = c^T \cdot x$ mit:
  - $c = (c_1, c_2, \ldots, c_d)$.
- O.B.d.A. gibt es eine zulässige Lösung.
- O.B.d.A. ist die Lösung eindeutig.
  - Falls nicht, so setze:
  - $c_i = c_i + \varepsilon^i$ für ein $\varepsilon > 0$.
  - Zielfunktion wird virtuell perturbiert (durcheinander wirbeln,stören).
  - D.h. es wird lexikographisch kleinste Basislösung gewählt.
- Jede Nebenbedingung entspricht einer Hyperebene.

Vorgaben

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

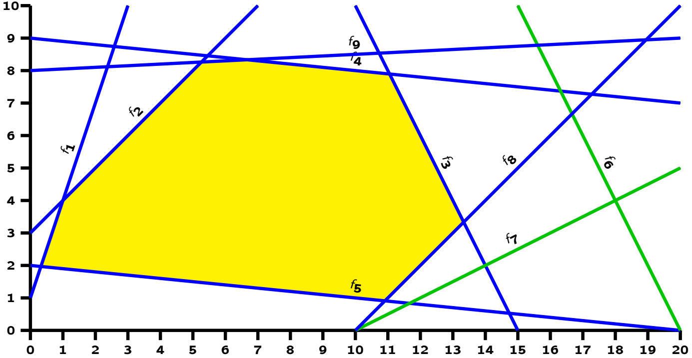
- Das LP sollte nicht unbeschränkt sein.
- Daher werden zusätzlich Box-Bedingungen eingefügt.
  - $-t \leqslant x_i \leqslant t$ für: $(1 \leqslant i \leqslant d)$.
  - t muss so groß sein, dass keine Basislösung verloren geht.
  - D.h. in jeder Basislösung müssen die Variablenwerte zwischen t und $-t$ liegen.
  - So ein Wert t existiert immer.
  - t kann polynomiell in der Eingabelänge bestimmt werden.
  - Man kann auch t als symbolischen Wert darstellen.
  - Dann wird t im folgenden Algorithmus immer größer sein als jeder bis dahin berechnete Wert.

Algorithmus von Seidel (1991)

d **Variablen**, *m* **Nebenbedingungen**, **Dimentsion** *n*

- Idee:
  - Sei H die Menge der Nebenbedingungen (Ohne Box-Bedingungen).
  - Wähle zufällig $h \in H$ aus, und
  - löse dann rekursiv.
- Für $H' \subset H$ sei $LP(H')$ das LP, bei dem alle Nebenbedingungen aus $H \setminus H'$ gestrichen wurden.
- Die optimale Basislösung von $LP(H')$ wird mit $opt(H')$ bezeichnet.
- Algorithmus von Seidel:
  1. Falls $d = 1$ oder $m = 0$, so gebe $opt(H)$ aus.
  2. Ansonsten wähle uniform eine Nebenbedingung $h \in H$ aus, und berechne $opt(H \setminus \{h\})$ rekursiv.
  3. Falls $opt(H \setminus \{h\})$ die Nebenbedingung h nicht verletzt, so gebe $opt(H \setminus \{h\}) = opt(H)$ aus.
  4. Ansonsten berechne den Schnitt des Lösungspolyhedrons mit der Hyperebene h, und löse das so entstandene $(d-1)$-dimensionale LP rekursiv.

Beispiel (Abstieg der ersten vier Rekursionen)



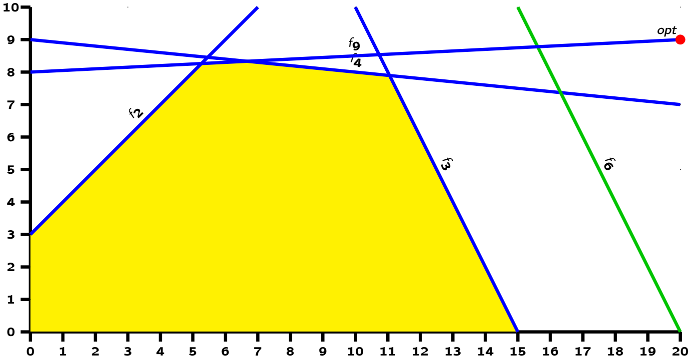
d **Variablen,** m **Nebenbedingungen, Dimentsion** n

1. **Falls** $d = 1$ **oder** $m = 0$**, so gebe** $opt(H)$ **aus.**
2. **Ansonsten wähle** $h \in H$ **aus, und berechne** $opt(H \setminus \{h\})$ **rekursiv.**
3. **Falls** $opt(H \setminus \{h\})$ **die Nebenbedingung** h **nicht verletzt, so gebe** $opt(H \setminus \{h\}) = opt(H)$ **aus.**
4. **Ansonsten berechne den Schnitt mit der Hyperebene** h**, und löse das so entstandene** $(d-1)$**-dimensionale LP rekursiv.**

Entferne nacheinander f_8, f_1, f_5 und f_7 und löse jeweils rekursiv.

$f_1 : y \leqslant 3 \cdot x + 1$	$f_4 : y \leqslant -0.9 \cdot x + 9$	$f_7 : y \geqslant 0.5 \cdot x - 5$
$f_2 : y \leqslant 1 \cdot x + 2$	$f_5 : y \geqslant -0.9 \cdot x + 2$	$f_8 : y \geqslant 1 \cdot x - 10$
$f_3 : y \leqslant -2 \cdot x + 30$	$f_6 : y \leqslant -2 \cdot x + 40$	$f_9 : y \leqslant 1.05 \cdot x + 8$

Beispiel (Abstieg der nächsten vier Rekursionen)



Entferne nacheinander f_4, f_6, f_3 und f_2 und löse jeweils rekursiv.

$$f_4 : y \leqslant -0.9 \cdot x + 9$$

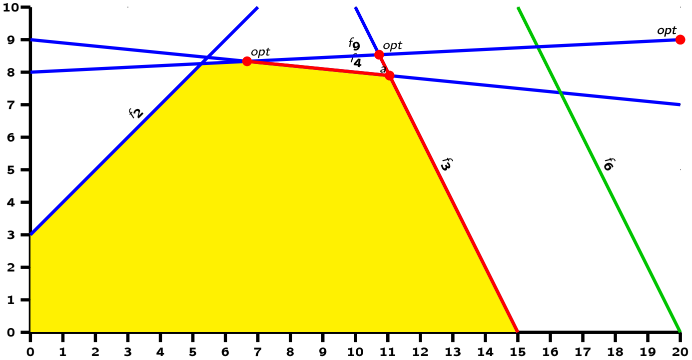
$$f_2 : y \leqslant 1 \cdot x + 2$$

$$f_3 : y \leqslant -2 \cdot x + 30 \quad f_6 : y \leqslant -2 \cdot x + 40 \quad f_9 : y \leqslant 1.05 \cdot x + 8$$

d **Variablen,** m **Nebenbedingungen, Dimentsion** n

1. Falls $d = 1$ **oder** $m = 0$**, so gebe** $opt(H)$ **aus.**
2. Ansonsten wähle $h \in H$ **aus, und berechne** $opt(H \setminus \{h\})$ **rekursiv.**
3. Falls $opt(H \setminus \{h\})$ **die Nebenbedingung** h **nicht verletzt, so gebe** $opt(H \setminus \{h\}) = opt(H)$ **aus.**
4. Ansonsten berechne den Schnitt mit der Hyperebene h**, und löse das so entstandene** $(d-1)$**-dimensionale LP rekursiv.**

Beispiel (Aufstieg der letzten vier Rekursionen)



d **Variablen,** m **Nebenbedingungen, Dimentsion** n

1. **Falls** $d = 1$ **oder** $m = 0$**, so gebe** $opt(H)$ **aus.**
2. **Ansonsten wähle** $h \in H$ **aus, und berechne** $opt(H \setminus \{h\})$ **rekursiv.**
3. **Falls** $opt(H \setminus \{h\})$ **die Nebenbedingung** h **nicht verletzt, so gebe** $opt(H \setminus \{h\}) = opt(H)$ **aus.**
4. **Ansonsten berechne den Schnitt mit der Hyperebene** h**, und löse das so entstandene** $(d-1)$**-dimensionale LP rekursiv.**

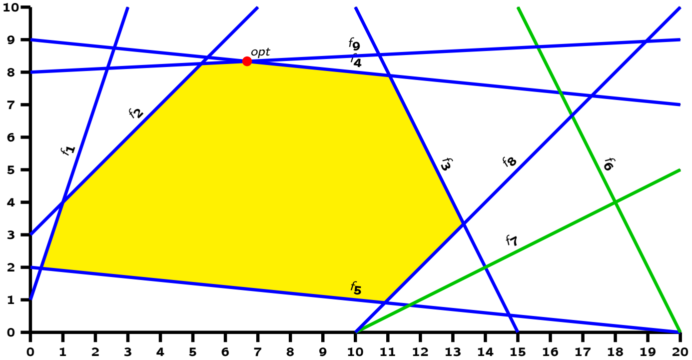
Füge nacheinander wieder ein: f_2, f_3, f_6 und f_4.

$$f_4 : y \leqslant -0.9 \cdot x + 9$$

$$f_2 : y \leqslant 1 \cdot x + 2$$

$$f_3 : y \leqslant -2 \cdot x + 30 \quad f_6 : y \leqslant -2 \cdot x + 40 \quad f_9 : y \leqslant 1.05 \cdot x + 8$$

Beispiel (Aufstieg der ersten vier Rekursionen)



Füge nacheinander wieder ein: f_7, f_5, f_1 und f_8.

$f_1 : y \leqslant 3 \cdot x + 1$	$f_4 : y \leqslant -0.9 \cdot x + 9$	$f_7 : y \geqslant 0.5 \cdot x - 5$
$f_2 : y \leqslant 1 \cdot x + 2$	$f_5 : y \geqslant -0.9 \cdot x + 2$	$f_8 : y \geqslant 1 \cdot x - 10$
$f_3 : y \leqslant -2 \cdot x + 30$	$f_6 : y \leqslant -2 \cdot x + 40$	$f_9 : y \leqslant 1.05 \cdot x + 8$

d **Variablen,** m **Nebenbedingungen, Dimentsion** n

1. **Falls** $d = 1$ **oder** $m = 0$**, so gebe** $opt(H)$ **aus.**
2. **Ansonsten wähle** $h \in H$ **aus, und berechne** $opt(H \setminus \{h\})$ **rekursiv.**
3. **Falls** $opt(H \setminus \{h\})$ **die Nebenbedingung** h **nicht verletzt, so gebe** $opt(H \setminus \{h\}) = opt(H)$ **aus.**
4. **Ansonsten berechne den Schnitt mit der Hyperebene** h**, und löse das so entstandene** $(d-1)$**-dimensionale LP rekursiv.**

Beachte beim Bestimmen der Laufzeit

Wenn m oder d klein ist, dann kann man direkt die Lösung und die Laufzeit zum Bestimmen selbiger angeben.

Wenn eine Nebenbedingung hinzugefügt wird, so ist die Laufzeit zum Test, ob die bisherige Lösung gültig bleibt, einfach zu bestimmen.

Einzig kompliziert bleibt der Fall, wenn die bisherige Lösung nicht gültig bleibt. Dann erfolgt ein rekursiver Aufruf, bei dem der Schnitt einer Hyperebene mit dem bisherigen Polyhedron bestimmet wird. Man beachte:

- Eine Variable wird durch eine Linearkombination der anderen Variablen ersetzt.
- Die Dimension nimmt dabei um eins ab, und eine Nebenbedingung wird entfernt.
- Die Randbedingungen (Box-Bedingung) an die ersetzte Variable werden auch durch eine Linearkombination der anderen Variablen ersetzt. Damit sind nun zwei neue Nebenbedingungen entstanden.

Damit kann nun ein rekursive Formel bestimmt und ausgewertet werden.

Laufzeiten und Details (Schritt 1 bis 3)

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Falls $d = 1$ gilt, gibt es nur noch eine Variable x_i.
  - Es gibt noch m Nebenbedingungen.
  - Damit kann x_i in Zeit $O(m)$ bestimmt werden.
  - Falls $x_i \in \{-t, t\}$ ist das LP ohne Box-Bedingungen unbeschränkt.
  - Ansonsten ist $-t < x_i < t$ die optimale Lösung.
- Falls $m = 1$ gilt, gibt es nur eine Nebenbedingung neben den Box-Bedingungen.
  - Damit haben wir ein relaxiertes Rucksackproblem.
  - Dies kann optimal mittels Greedy gelöst werden.
  - Laufzeit: $O(d \log d) = O(d^2)$.
- In Schritt 2 wird rekursiv ein Problem gelöst, bei dem m um eins verringert ist.
- In Schritt 3 wird getestet, ob die gewählte Hyperebene h nicht die Lösung $opt(H \setminus \{h\})$ verletzt.
  - Das kann in $O(d)$ gelöst werden.

Laufzeiten und Details (Schritt 4)

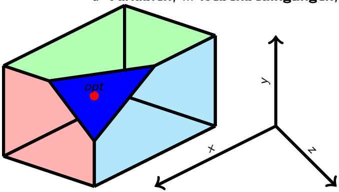
d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Fasse die Nebenbedingung h als Gleichung auf.
- Löse h zu einer beliebigen Variablen x_k auf ($k \in \{1, 2, \ldots, d\}$).
- D.h. $x_k = s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$.
- Ersetze jedes Auftreten von x_k durch $s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$ in:
  - allen Nebenbedingungen aus $H \setminus \{h\}$,
  - in der Zielfunktion, und
  - in der Box-Bedingung für $h_k^- \leqslant x_k \leqslant h_k^+$:
    - $h_k^- \leqslant s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$ und
    - $h_k^+ \geqslant s_k(x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d)$.
  - Nenne diese beiden letzten Bedingungen h' und h''.
- Setze nun $H' = H \cup \{h', h''\} \setminus \{h\}$ und löse H' rekursiv.
- Beachte: H' hat $m + 1$ Nebenbedingungen, aber nur $d - 1$ Variablen.

Beispiel Schritt 4 ($d = 3$)

Nebenbedingungen:

$$0 \leqslant x, \quad x \leqslant 10 \quad x + y + z \leqslant 25$$
$$0 \leqslant y, \quad y \leqslant 10$$
$$0 \leqslant z, \quad z \leqslant 10$$



- Gegeben seien die sechs einfachen Nebenbedingungen
- Zu maximieren sei die Funktion $f(x, y, z) = 0.99 \cdot x + 0.974 \cdot y + z$.
- Das Maximum wird bei $f(10, 10, 10)$ erreicht.
- Die zusätzliche Nebenbedingung $x + y + z \leqslant 25$ widerspricht diesem Maximum
- Das neue Maximum liegt damit auf der blauen Fläche.
- Wir lösen nach z auf: $z = 25 - x - y$. Damit verbleibt:

$$0 \leqslant x, \quad x \leqslant 10, \quad x + y \leqslant 25 \quad f(x, y) = -0.01 \cdot x - 0.026 \cdot y + 25$$
$$0 \leqslant y, \quad y \leqslant 10, \quad 15 \leqslant x + y$$

Korrektheit

d **Variablen**, m **Nebenbedingungen, Dimentsion** n

1. Falls $d = 1$ **oder** $m = 0$**, so gebe** $opt(H)$ **aus.**
2. Ansonsten wähle $h \in H$ **aus, und berechne** $opt(H \setminus \{h\})$ **rekursiv.**
3.Falls $opt(H \setminus \{h\})$ **die Nebenbedingung** h **nicht verletzt, so gebe** $opt(H \setminus \{h\}) = opt(H)$ **aus.**
4. Ansonsten berechne den Schnitt mit der Hyperebene h**, und löse das so entstandene** $(d-1)$**-dimensionale LP rekursiv.**

- Setze $k(d, m) = 2 \cdot d + m$.
- Pro rekursiven Aufruf wird $k(d, m)$ um eins verringert.
- Bei spätestens $k(d, m) = 3$ terminiert das Verfahren mit:
  - $d = 1$ oder $m = 0$.
- Falls h eine Nebenbedingung ist, die keinen Einfluss auf die optimale Lösung hat.
  - Dann bestimmt der Algorithmus $opt(H \setminus \{h\})$ in Schritt 3.
- Falls h eine Nebenbedingung ist, mit der sich die optimale Lösung schneidet.
  - Dann befindet sich die optimale Lösung auf der Hyperebene, die durch h beschrieben wird.
  - Diese betrachtet der Algorithmus in Schritt 4.

Laufzeit

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Sei $T(m, d)$ eine obere Schranke für die erwartete Laufzeit.
- Falls $m > 0$ und $d > 1$ ergibt sich folgende Abschätzung für die Laufzeiten der vier Schritte:
  - Schritt 1: $T_1 = O(1)$.
  - Schritt 2: $T_2 = T(m-1, d)$.
  - Schritt 3: $T_3 = O(d)$.
  - Schritt 4: $T_4 = T(m+1, d-1) + O(d \cdot m)$.
- Dabei wird der Schritt 4 nicht immer ausgeführt.
- Im Folgenden schätzen wir diese Wahrscheinlichkeit ab.

Aussage

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

Lemma

Die Wahrscheinlichkeit, dass Schritt 4 ausgeführt wird, ist höchstens d/m.

Beweis:

- Sei $x^* = opt(H)$.
- x^* ist damit auf dem Schnitt von d Hyperebenen.
- Diese d Hyperebenen sind Nebenbedingungen oder Box-Bedingungen.
- Sei D eine Auswahl mit $|D| = d$ dieser x^* bestimmenden Hyperebenen.
- Damit gilt: $opt(D) = x^*$.
- Schritt 4 wird ausgeführt, falls $opt(H \setminus \{h\})$ die Nebenbedingung h verletzt.
- D.h. Schritt 4 wird ausgeführt, falls $h \in D$.
- Damit ist die Wahrscheinlichkeit, dass Schritt 4 ausgeführt wird:

$$\mathbb{P}\mathrm{r}[h \in D] = \frac{|D \cap H|}{|H|} \leqslant \frac{d}{m}.$$

Abschätzung der Laufzeit

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

- Im Falle $m > 0$ und $d > 1$ gilt: $T(m,d) = T_1 + T_2 + T_3 + \frac{d}{m} \cdot T_4$.
- Unter Vernachlässigung der konstanten Faktoren:
- Im Falle $m > 0$ und $d > 1$ gilt:

$$T(m,d) \leqslant T(m-1,d) + d^2 + \frac{d}{m} \cdot T(m+1,d-1).$$

- Im Falle $m = 0$ oder $d = 1$ gilt:

$$T(m,d) \leqslant d^2 + m.$$

- Wir definieren nun $f(1) = 1$ und für $d > 1$:

$$f(d) = d \cdot f(d-1) + 3 \cdot d^3$$

- Damit gilt:

$$f(d) = d! + \sum_{k=2}^{d} 3 \cdot k^3 \cdot \frac{d!}{(k-1)!} = O(d!).$$

- Denn $\sum_{k=2}^{d} \frac{3 \cdot k^3}{(k-1)!}$ ist durch Konstante beschränkt.

Finale Abschätzung der Laufzeit

d **Variablen**, m **Nebenbedingungen**, **Dimentsion** n

Lemma

Es gilt: $T(m, d) \leqslant (m-1) \cdot f(d) + 2 \cdot d^2$.

Beweis:

- Induktionsanfang: $m = 1$

$$T(m, d) \leqslant d^2 + 1 \leqslant (m-1) \cdot f(d) + 2 \cdot d^2.$$

- Induktionsanfang: $d = 1$

$$T(m, d) \leqslant m + 1 \leqslant (m-1) \cdot f(d) + 2 \cdot d^2.$$

- Für den Induktionsschritt setzen wir:
  - $k = 2 \cdot d + m$
  - Es sei nun $m \geqslant 2$ und $d \geqslant 2$.
  - Behauptung sei gezeigt für alle d' und m' mit $2 \cdot d' + m' < k$.
  - Also auch für $(m-1, d)$ und $(m+1, d-1)$.

Induktionsschluss

$$T(m,d) \leqslant (m-\mathbf{1})\cdot f(d) + \mathbf{2}\cdot d^{\mathbf{2}}$$
$$f(d) = d\cdot f(d-\mathbf{1}) + \mathbf{3}\cdot d^{\mathbf{3}}$$

$$\begin{aligned}
T(m,d) &\leqslant T(m-1,d) + d^2 + \tfrac{d}{m}\cdot T(m+1,d-1)\\
&\leqslant (m-2)\cdot f(d) + 2\cdot d^2 + d^2 + \tfrac{d}{m}\cdot T(m+1,d-1)\\
&\leqslant (m-2)\cdot f(d) + 3\cdot d^2 + \tfrac{d}{m}\cdot m\cdot f(d-1) + 2\cdot (d-1)^2\\
&\leqslant (m-2)\cdot f(d) + 3\cdot d^2 + \tfrac{d}{m}\cdot m\cdot f(d-1) + 2\cdot d^2\\
&= (m-2)\cdot f(d) + 3\cdot d^2 + d\cdot \tfrac{f(d)-3\cdot d^3}{d} + 2\cdot d^2\\
&= (m-2)\cdot f(d) + 3\cdot d^2 + f(d) - 3\cdot d^3 + 2\cdot d^2\\
&\leqslant (m-1)\cdot f(d) + 2\cdot d^2
\end{aligned}$$

Aussage

Theorem

Der Algorithmus von Seidel löst ein zulässiges d-dimensionales LP mit m Nebenbedingungen in erwarteter Laufzeit von $O(m \cdot d!)$.

- Ist d konstant, so ist die erwartete Laufzeit $O(m)$.
- Ist $d > 10$ so ist die Konstante aber ein wenig unpraktisch.

Beispiel

- Maximiere: $5 \cdot x_1 + 7 \cdot x_2$ unter den Nebenbedingungen:

$$x_1 + 4 \cdot x_2 \leqslant 4,\ x_1 + x_2 \leqslant 2, \text{ und } x_1, x_2 \geqslant 0.$$

- Einfache obere Schranke: $5 \cdot x_1 + 17 \cdot x_2 \leqslant 18$, gewonnen aus:

$$4 \cdot (x_1 + 4 \cdot x_2 \leqslant 4) \text{ plus } x_1 + x_2 \leqslant 2.$$

- Weitere einfache obere Schranke: $5 \cdot x_1 + 11 \cdot x_2 \leqslant 14$, gewonnen aus:

$$2 \cdot (x_1 + 4 \cdot x_2 \leqslant 4) \text{ plus } 3 \cdot (x_1 + x_2 \leqslant 2).$$

- Rezept für obere Schranke: Bestimme y_1, y_2 mit $y_1 + y_2 \geqslant 5$ und $4 \cdot y_1 + y_2 \geqslant 7$ und addiere:

$$y_1 \cdot x_1 + y_1 \cdot 4 \cdot x_2 \leqslant y_1 \cdot 4 \text{ plus } y_2 \cdot x_1 + y_2 \cdot x_2 \leqslant y_2 \cdot 2.$$

- Damit haben wir ein Ungleichungssystem zum Bestimmen einer kleinsten oberen Schranke, d.h.:
- Minimiere $4 \cdot y_1 + 2 \cdot y_2$ unter den Nebenbedingungen:

$$y_1 + y_2 \geqslant 5,\ 4 \cdot y_1 + y_2 \geqslant 7, \text{ und } y_1, y_2 \geqslant 0.$$

Definition

Definition

Gegeben sei ein LP (im Weiteren das primale LP) mit n Variablen und m Nebenbedingungen in kanonischer Form:

$$\text{Maximiere } c^T x \text{ unter } Ax \leqslant b,\ x \geqslant 0.$$

Das duale LP hat m Variablem und n Nebenbedingungen und die Form:

$$\text{Minimiere } y^T b \text{ unter } y^T A \geqslant c^T,\ y \geqslant 0.$$

Äquivalente Schreibweise: Minimiere $b^T y$ unter $A^T y \geqslant c$, $y \geqslant 0$.

- Aus den n Variablen des primalen LP werden n Nebenbedingungen des dualen LPs.
- Aus den m Nebenbedingungen des primalen LP werden m Variablen des dualen LPs.

Dual von Dual

primal: $c^T x : Ax \leqslant b, x \geqslant \mathbf{0}$, **dual**: $b^T y : A^T y \geqslant c, y \geqslant \mathbf{0}$

Theorem

Das duale LP des dualen LP ist das primale LP.

Beweis:

- Primale LP:
  Maximiere $c^T x$ unter $Ax \leqslant b$, $x \geqslant 0$.
- Duale LP:
  Minimiere $b^T y$ unter $A^T y \geqslant c$, $y \geqslant 0$.
- Duale LP in kanonischer Form:
  Maximiere $-b^T y$ unter $-A^T y \leqslant -c$, $y \geqslant 0$.
- Davon wieder das duale LP:
  Minimiere $-c^T x$ unter $-Ax \geqslant -b$, $x \geqslant 0$.
- Das wieder in kanonische Form gebracht:
  Maximiere $c^T x$ unter $Ax \leqslant b$, $x \geqslant 0$.

Schwaches Dualitätsprinzip

primal: $c^T x : Ax \leqslant b, x \geqslant \mathbf{0}$, **dual**: $b^T y : A^T y \geqslant c, y \geqslant \mathbf{0}$

Theorem

Sei x eine zulässige Lösung für das primale LP und sei y eine zulässige Lösung für das duale LP. Dann gilt $y^T \cdot b \geqslant x^T \cdot x$.

Beweis:

- Aus der Zulässigkeit folgt jeweils: $x \geqslant 0$ und $y^T A \geqslant c^T$.
- Damit folgt: $c^T x \leqslant y^T A x$.
- Weiter folgt aus der Zulässigkeit: $y \geqslant 0$ und $Ax \leqslant b$.
- Damit folgt: $y^T A x \leqslant y^T b$.
- Damit gilt: $c^T x \leqslant y^T A x \leqslant y^T b$.

Starkes Dualitätsprinzip

primal: $c^T x : Ax \leqslant b, x \geqslant \mathbf{0}$, **dual**: $b^T y : A^T y \geqslant c, y \geqslant \mathbf{0}$

Theorem

Sei x^* eine optimale Lösung für das primale LP und sei y^* eine optimale Lösung für das duale LP. Dann gilt $y^T \cdot b = x^T \cdot x$.

Beweis (siehe Script):

- Der Beweis ist sogar konstruktiv.
- D.h. eine "primale Lösung" kann in polynomieller Zeit in eine "duale Lösung" überführt werden.

Flussproblem als LP

- Gegeben $G = (V, E, s, t, c)$. Sei $P_{s,t}$ die Menge der einfachen Pfade von s nach t.
- Für jeden Pfad $p \in P_{s,t}$ gibt es eine Variable x_p.
- Das primale LP lautet: Maximiere $\sum_{p \in P_{s,t}} x_p$ unter den Nebenbedingungen:

$$\sum_{p: e \in p} x_p \leqslant c(e), \ \forall e \in E \text{ und } x_p \geqslant 0, \ \forall p \in P_{s,t}.$$

- Das duale LP lautet: Minimiere $\sum_{e \in E} c(e) y_e$ unter den Nebenbedingungen:

$$\sum_{e \in p} y_e \geqslant 1, \ \forall p \in P_{s,t} \text{ und } y_e \geqslant 0, \ \forall e \in E.$$

- Für das duale LP gilt (nach dem folgenden Abschnitt): die Werte von y_e sind aus $\{0, 1\}$.
- Das duale LP entspricht dem Finden eines minimalen Schnitts zwischen s und t.

Relaxiertes Matching

primal: $c^T x : Ax \leqslant b, x \geqslant \mathbf{0}$, **dual**: $b^T y : A^T y \geqslant c, y \geqslant \mathbf{0}$

- Gegeben $G = (V, E)$.
- Für jede Kante $e \in E$ gibt es Variable x_e.
- Das primale LP lautet: Maximiere $\sum_{e \in E} x_e$ unter den Nebenbedingungen:

$$\sum_{e \in v} x_e \leqslant 1, \ \forall v \in V \text{ und } x_e \geqslant 0, \ \forall e \in E.$$

- Das duale LP lautet: Minimiere $\sum_{v \in V} y_v$ unter den Nebenbedingungen:

$$\sum_{v \in e} y_v \geqslant 1, \ \forall e \in E \text{ und } y_v \geqslant 0, \ \forall v \in V.$$

- Das duale LP entspricht einem relaxierten Vertex-Cover-Problem.
- Aber hier liegt eine Ganzzahligkeit der Lösungen nur auf bipartiten Graphen vor.
- Somit sind bipartites Matching und Vertex-Cover auf bipartiten Graphen dual zueinander.

Beispiele

- ILPs (Integer Linear Programs) unterscheiden sich nur von LPs dadurch, dass die Variablen aus $\mathbb{N}$ sein müssen.
- Rucksackproblem: Gegeben d Objekte mit Gewichten g_i und Nutzen v_i für $1 \leqslant i \leqslant d$. Weiter sei G die Gewichtsschranke des Rucksacks.
- Maximiere $\sum_{i=1}^{d} v_i \cdot x_i$ unter den Nebenbedingungen:

$$\sum_{i=1}^{d} g_i \cdot x_i \leqslant G, \ \forall i : 1 \leqslant i \leqslant d : x_i \in \{0, 1\}.$$

- Gewichtetes Matchingproblem: Gegeben $G = (V, E)$ mit Kantengewichten $w_e, e \in E$.
- Maximiere $\sum_{e \in E} w_e \cdot x_e$ unter den Nebenbedingungen:

$$\sum_{e \in E : v \in e} x_e \leqslant 1, \ \forall v \in V \text{ und } x_e \in \{0, 1\}, \ \forall e \in E.$$

- Das Rucksackproblem ist NP-hart, aber das gewichtete Matchingproblem ist in $\mathcal{P}$.
- Wir untersuchen im Folgenden, wann ein ILP in $\mathcal{P}$ liegt.

Unimodularität

primal: $c^T x : Ax \leqslant b, x \geqslant \mathbf{0}$, **dual**: $b^T y : A^T y \geqslant c, y \geqslant \mathbf{0}$

Definition

- Eine ganzzahlige quadratische Matrix heißt unimodular, falls ihre Determinante 1 oder -1 ist.
- Eine ganzzahlige quadratische Matrix heißt total unimodular, falls jede quadratische reguläre Teilmatrix unimodular ist.

Theorem

Sei A total unimodular. Dann sind alle Basislösungen von $A \cdot x = b$ ganzzahlig.

Beweis:

- Sei δ Basis von A. Die Basislösung ergibt sich aus $A_\delta \cdot x_\delta = b$.
- A_δ ist dabei eine quadratische Teilmatrix.
- Damit folgt die Aussagen nach der Cramerschen Regel:

$$x_{\delta(i)} = \frac{\det(A_{\delta(1)}, \ldots, A_{\delta(i-1)}, b, A_{\delta(i+1)} \ldots, A_{\delta(k)})}{\det(A_\delta)}.$$

LPs

Theorem

Sei A total unimodular, Dann sind alle Basislösungen von dem LP $A \cdot x \leqslant b$ ganzzahlig.

- Sei m die Anzahl der Zeilen von A.
- Mit Schlupfvariablen erhalten wir ein LP in Gleichungsform: $(A \mid E_m)$.
- Wir zeigen nun, dass $(A \mid E_m)$ total unimodular ist.
- Sei C beliebige quadratische Teilmatrix von $(A \mid E_m)$.
- Wir können C wie folgt umgeformt darstellen:

$$C' = \begin{pmatrix} M & 0 \\ * & E_k \end{pmatrix}.$$

  - Dabei ist E_k eine $k \times k$-Einheitsmatrix.
  - M ist eine $(m-k) \times (m-k)$-Teilmatrix von A.
- Es folgt: $|\det(C)| = |\det(C')| = |\det(M)| = 1$.
- Damit ist $(A \mid E_m)$ total unimodular.

Eine Eigenschaft für unimodular

primal: $c^T x : Ax \leqslant b, x \geqslant \mathbf{0}$, **dual**: $b^T y : A^T y \geqslant c, y \geqslant \mathbf{0}$

Theorem

Eine Matrix A mit Einträgen aus $\{-1, 0, 1\}$ ist total unimodular,

- *nicht mehr als zwei Einträge pro Spalte aus $\{-1, 1\}$ sind und*
- *die Zeilen in zwei Mengen I_1 und I_2 aufteilen lassen, für die gilt:*
    - *Falls eine Spalte zwei unterschiedliche Einträge (-1 und 1) enthält, so sind die zugehörigen Zeilen in der gleichen Menge I_i.*
    - *Falls eine Spalte zwei Einträge -1 oder zwei Einträge 1 hat, so sind die zugehörigen Zeilen in der unterschiedlichen Mengen I_i.*

Beweis:

- Beweis erfolgt per Induktion über die Größe der Teilmatrizen.
- Induktionsanfang: eine 1×1 Teilmatrix ist offensichtlich unimodular.
- Induktionsannahme: jede $(k-1) \times (k-1)$ Teilmatrix sei unimodular.

Beweis

Induktionsannahme: jede $(k-1) \times (k-1)$ Teilmatrix sei unimodular.

- Sei C eine $k \times k$ Teilmatrix von A.
- Wir müssen zeigen: C ist nicht regulär oder C ist unimodular.
  - Falls C eine Spalte enthält, die 0^k ist, so ist C nicht regulär.
  - Falls C eine Spalte enthält, die nur einen Eintrag aus $\{-1, 1\}$ enthält, so können wir die Determinante nach dieser Spalte entwickeln.
    - Dazu sind die Zeile und Spalte dieses Eintrags zu entfernen und die Determinante der verbleibenden $(k-1) \times (k-1)$ ist zu betrachten.
    - Damit gilt die Behauptung in diesem Fall.
  - Falls jede Spalte aus C zwei Einträge enthält, so gilt für jede Spalte j:
    - $\sum_{i \in I_1} a_{i,j} = \sum_{i \in I_2} a_{i,j}$.
    - Damit können wir die Zeilenvektoren aus I_1 aufsummieren und die aus I_2 subtrahieren.
    - Das Ergebnis ist ein Nullvektor. Damit ist C nicht regulär.

Inzidenzmatrix

Induktionsannahme: jede $(k-1) \times (k-1)$ Teilmatrix sei unimodular.

- Die Inzidenzmatrix eines Graphen hat folgende Gestalt:
  - Für jeden Knoten gibt es eine Zeile.
  - Für jede Kante gibt es eine Spalte.
  - Eine 1 besagt: die ungerichtete Kante ist zu dem Knoten inzident.
  - Eine 1 besagt: die gerichtete Kante endet in dem Knoten.
  - Eine -1 besagt: die gerichtete Kante startet in dem Knoten.
- Die Inzidenzmatrix eines gerichteten Graphen genügt den obigen Bedingungen:
  - Alle Zeilen gehören hier zur Menge I_1.
- Die Inzidenzmatrix eines ungerichteten bipartiten Graphen (V, W, E) genügt den obigen Bedingungen:
  - Alle Zeilen die Knoten aus V repräsentieren gehören zur Menge I_1.
  - Alle Zeilen die Knoten aus W repräsentieren gehören zur Menge I_2.

Folgerungen

Induktionsannahme: jede $(k-1) \times (k-1)$ Teilmatrix sei unimodular.

Lemma

Ein LP in Standardform oder in kanonischer Form hat nur ganzzahlige Basislösungen, falls die Nebenbedingungsmatrix

- *der Inzidenzmatrix eines gerichteten Graphen oder*
- *der Inzidenzmatrix eines bipartiten Graphen entspricht.*

Damit liefen die relaxierten LP-Formulierungen eine ganzzahlige optimale Lösung für:

- maximalen Fluss,
- kürzester Weg,
- gewichtetes bipartites Matching, und
- bipartites Vertex-Cover.

Bemerkungen

Induktionsannahme: jede $(k-1) \times (k-1)$ Teilmatrix sei unimodular.

- Damit folgt auch die Dualität des Matchings und des Vertex-Cover auf bipartiten Graphen.
- Betrachte folgende Matrix (Inzidenzmatrix von K_3):

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

- Die Determinante dieser Matrix ist 2.
- Der zugehörige Graph K_3 hat nur ein Matching der Größe 1.
- Das relaxierte Matching hat aber einen Wert von $3/2$.
- Dazu wird jede Kante zur Hälfte gematcht.

Literatur

Induktionsannahme: jede $(k-1) \times (k-1)$ **Teilmatrix sei unimodular.**

- B. Korte, J. Vygen. Combinatorial Optimization: Theory and Algorithms, 2nd Edition, Springer, 2002.
- E. Lawler. Combinatorial Optimization: Networks and Matroids. Dover Publications, 1976.
- C. Papadimitriou und K. Steiglitz. Combinatorial Optimization: Algorithms and Complexity. Prentice Hall, 1982.
- A. Schrijver. Combinatorial Optimization: Polyhedra and Efficiency. Springer, 2003.

Fragen

- Wie arbeitet der Algorithmus von Seidel?
- Wie ist die Laufzeit für den Algorithmus von Seidel?

Legende

n : Nicht relevant

g : Grundlagen, die implizit genutzt werden

i : Idee des Beweises oder des Vorgehens

s : Struktur des Beweises oder des Vorgehens

w : Vollständiges Wissen