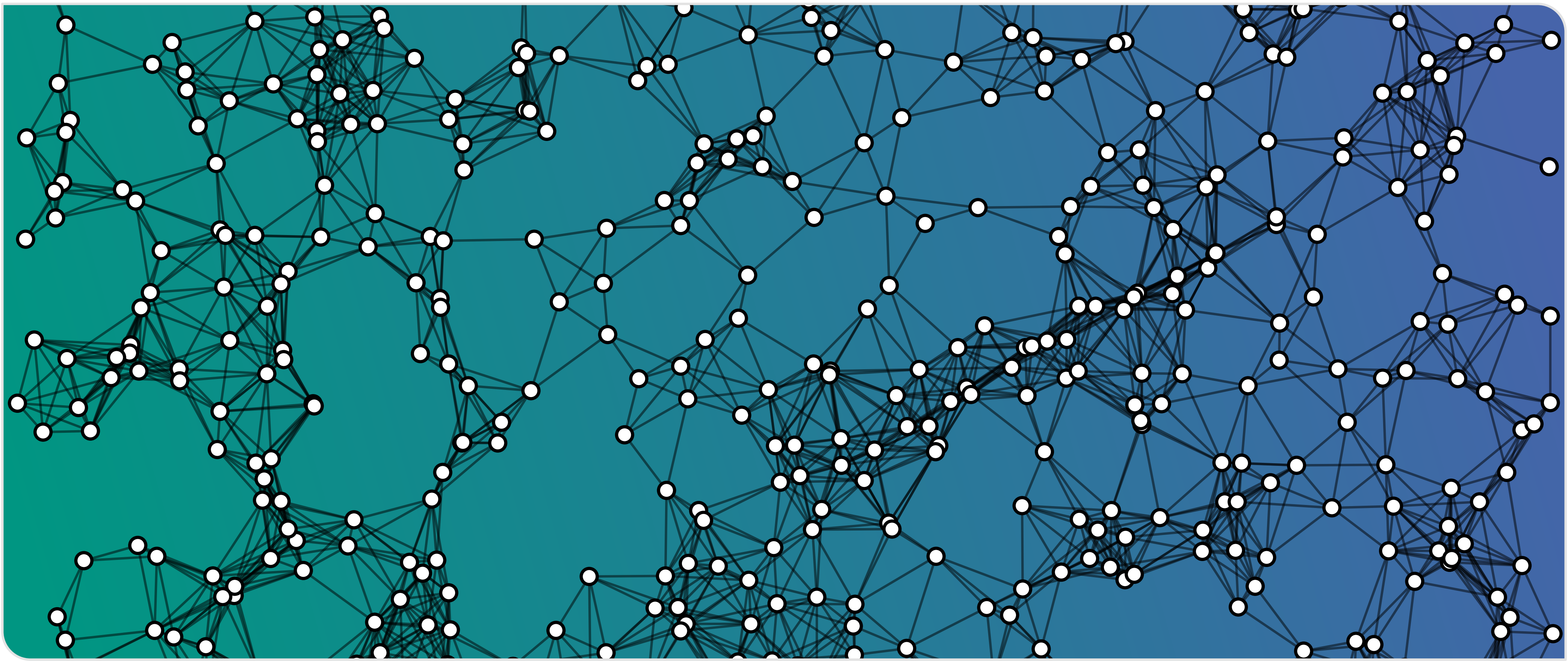
Karlsruher Institut für Technologie

Parametrisierte Algorithmen

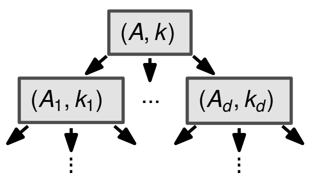
Lineare Programme: Dualität und Lenstras Theorem

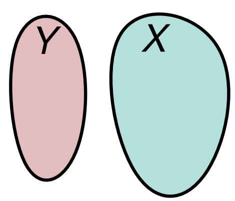


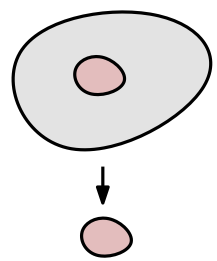
Inhalt

Basic Toolbox

- beschränkte Suchbäume
- iterative Kompression
- Kernbildung

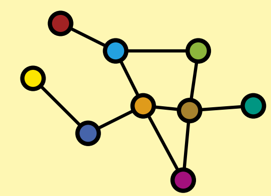


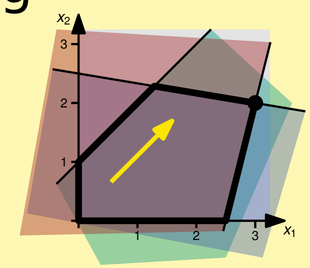


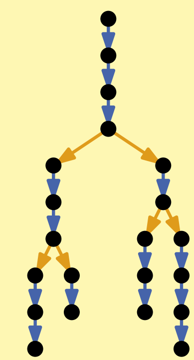


Erweiterte Toolbox

- lineare Programme
- Branch-and-Reduce
- Color Coding





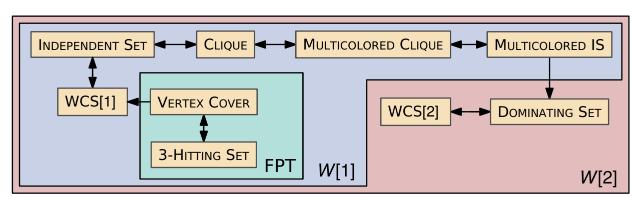


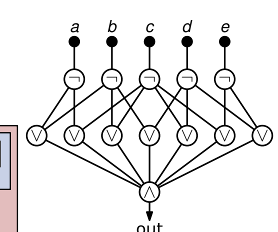
Baumweite

- dynamische Programme
- chordale & planare Graphen
- Courcelles Theorem

Untere Schranken

- parametrisierte Reduktionen
- boolesche Schaltkreise und die W-Hierarchie
- ETH und SETH





Beispiel: Ausgewogen und Billig

Karlsruher Institut für Technologie

Problem

- Burger entsprechen nicht den offiziellen Ernährungsrichtlinien
- pro Gericht fehlen 0,5 mg Vitamin A, 15 mg Vit. C, 4 g Ballaststoffe
- Ziel: Behebung dieses Problems bei möglichst geringen Kosten
- nutze dazu Karotten, Weißkohl und Gewürzgurken

	Karotten	Weißkohl	Gewürzgurken
Vitamin A (mg/kg)	35	0, 5	0, 5
Vitamin C (mg/kg)	60	300	10
Ballaststoffe (g/kg)	30	20	10
Preis (€/kg)	0, 75	0, 5	0, 15

. . . and when Rabbid said, "Honey or condensed milk with your bread?" he was so excited that he said, "Both," and then, so as not to seem greedy, he added, "But don't bother about the bread, please."

A. A. Milne, Winnie the Pooh

Lösung

- x_1, x_2, x_3 repräsentieren die Menge an Karotten, Kohl und Gurken

$$
\begin{aligned}
\text{minimiere:}\quad & 0,75x_1 + 0,5x_2 + 0,15x_3 \\
\text{Nebenbedingungen:}\quad & 35x_1 + 0,5x_2 + 0,5x_3 \geq 0,5 \\
& 60x_1 + 300x_2 + 10x_3 \geq 15 \\
& 30x_1 + 20x_2 + 10x_3 \geq 4 \\
& x_i \geq 0
\end{aligned}
$$

- optimale Lösung:
  - 9,5 g Karotten
  - 38 g Kohl
  - 290 g Gurken

Lineare Programme – Trivia

- wurden bereits in den 40er Jahren verwendet (und manuell gelöst)
- „Programm“ ist ein militärischer Begriff für verschiedene Arten von Plänen (z.B. Versorgungsplan, Verlegungsplan für Truppen etc.)
- erstes großes LP, das mit dem Simplex-Algorithmus gelöst wurde
  - optimiere Kosten für ausgewogene Ernährung
  - 77 Variablen, 9 Nebenbedingungen
  - Simplex-Methode (per Hand in 1947): 120 Personentage
- etwas später (mittels Computer): George Dantzig versucht seine eigene Ernährung zu optimieren
  - erster Versuch: mehrere Liter Essig pro Tag
  - zweiter Versuch: 200 Brühwürfel pro Tag
  - $\Rightarrow$ ein sinnvolles LP zu formulieren ist nicht immer trivial

Lineare Programme

Finde Reellwertige Belegung für Variablen $x_1, \dots, x_n$

- lineare Funktion in $x_1, \dots, x_n$ wird maximiert (minimiert)
- eingeschränkt durch lineare Nebenbedingungen (Ungleichungen)

Beispiel

maximiere: $x_1 + x_2$

sodass: $x_1 \geq 0$

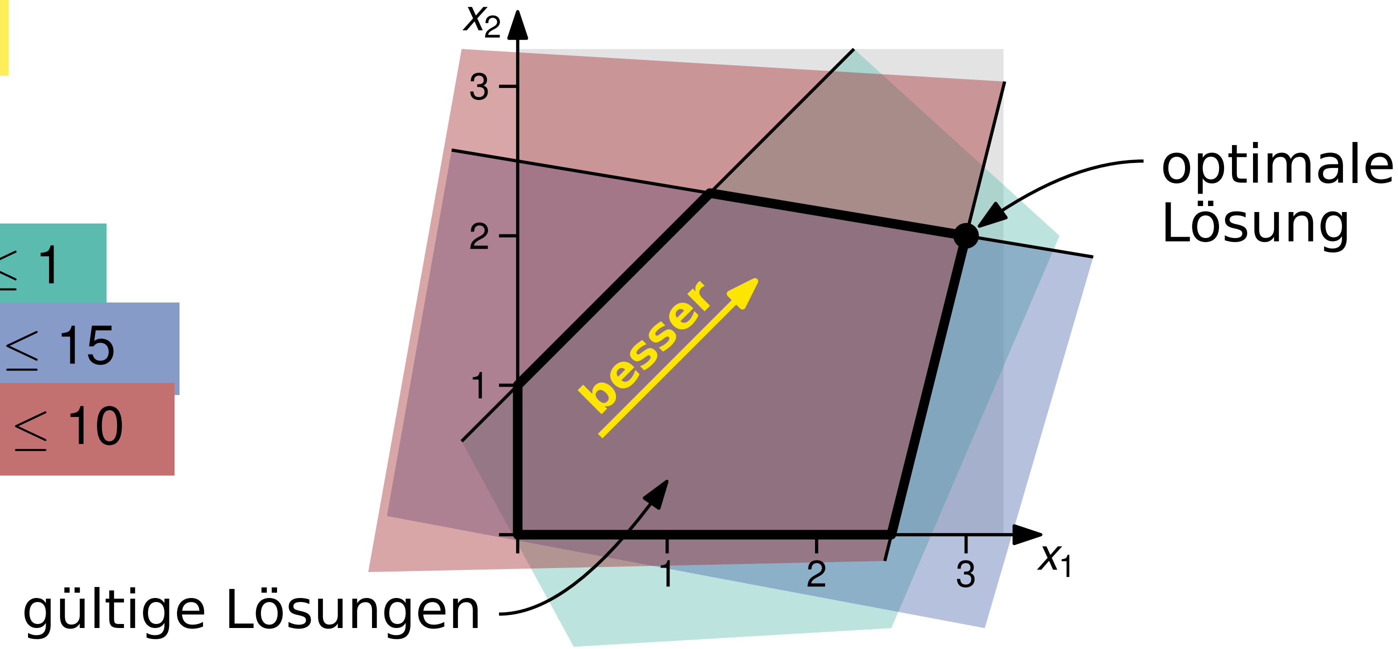
$x_2 \geq 0$

$x_2 - x_1 \leq 1$

$x_1 + 6x_2 \leq 15$

$4x_1 - x_2 \leq 10$

Geometrische Interpretation



Lösbarkeit

- LP ist **unlösbar (infeasible)**, wenn es keine gültige Lösung gibt
- LP ist **unbeschränkt (unbounded)**, wenn es beliebig gute gültige Lösungen gib (Optimierungsfunktion wird beliebig groß)

Matrixschreibweise

$$\begin{array}{rl} \text{max.:} & 2x_1 + x_2 \\ \text{sodass:} & x_1 \geq 0 \\ & x_2 \geq 0 \\ & x_2 - x_1 \leq 1 \\ & x_1 + 6x_2 \leq 15 \\ & 4x_1 - x_2 \leq 10 \end{array} \longrightarrow \begin{array}{rcrcr} 2x_1 & + & 1x_2 & & \\ 1x_1 & + & 0x_2 & \geq & 0 \\ 0x_1 & + & 1x_2 & \geq & 0 \\ -1x_1 & + & 1x_2 & \leq & 1 \\ 1x_1 & + & 6x_2 & \leq & 15 \\ 4x_1 & + & (-1x_2) & \leq & 10 \end{array} \longrightarrow \begin{array}{rcrcr} 2x_1 & + & 1x_2 & & \\ -1x_1 & + & 0x_2 & \leq & 0 \\ 0x_1 & + & (-1x_2) & \leq & 0 \\ -1x_1 & + & 1x_2 & \leq & 1 \\ 1x_1 & + & 6x_2 & \leq & 15 \\ 4x_1 & + & (-1x_2) & \leq & 10 \end{array}$$

Matrizen und Vektoren

$c = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$, $A = \begin{pmatrix} -1 & 0 \\ 0 & -1 \\ -1 & 1 \\ 1 & 6 \\ 4 & -1 \end{pmatrix}$, $b = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 15 \\ 10 \end{pmatrix}$ finde $x \in \mathbb{R}^2$ das $c^T x$ maximiert mit $Ax \leq b$

- allgemein: $x, c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$
- LPs sind häufig in dieser Form gegeben
- jedes LP lässt sich in diese Form bringen

Beispiel: Eis für ein ganzes Jahr

Probleme bei der Eisproduktion

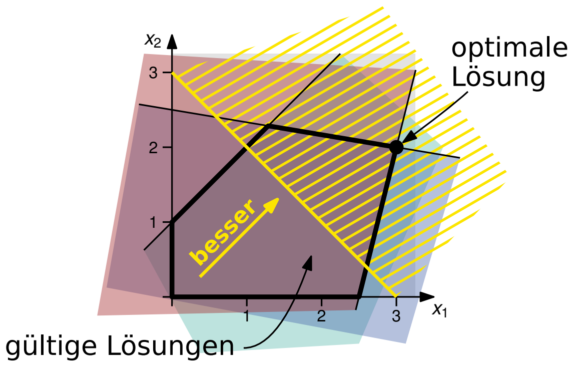
- der Eisverbrauch d_i hängt stark von dem aktuellen Monat i ab
- schwankende Produktionsmengen verursachen Kosten:
  50€ pro Tonne Veränderung von Monat i zu Monat $i+1$
- Lagerung kostet Geld: 20€ pro Tonne Überschuss am Ende jeden Monats

Formulierung als LP

- x_i repräsentiert die Produktionsmenge im Monat i
- s_i repräsentiert den Überschuss nach Monat i
- ausreichend Eis im Monat i: $x_i + s_{i-1} \geq d_i$
- neuer Überschuss nach Monat i: $s_i = x_i + s_{i-1} - d_i \quad \Leftrightarrow \quad x_i + s_{i-1} - s_i = d_i$
- minimiere Kosten: $\text{min.: } 20 \sum_{i=1}^{12} s_i + 50 \sum_{i=1}^{12} \overset{a_i}{\cancel{|x_i - x_{i-1}|}}$
- Lösung mittels Hilfsvariable a_i: $a_i \geq x_i - x_{i-1} \qquad a_i \geq x_{i-1} - x_i$
- in minimaler Lösung ist a_i das Maximum aus $x_i - x_{i-1}$ und $x_{i-1} - x_i$

Gültig vs. Optimal

Gültige Lösung $\rightarrow$ Optimum



- gegeben: Algo, der eine gültige Lösung findet
- beschränke Optimierungsfunktion durch festen Wert $\rightarrow$ binäre Suche
- Beispiel:

$$\begin{array}{rl} \text{maximiere:} & x_1 + x_2 \\ \text{sodass:} & x_1 \geq 0 \\ & x_2 \geq 0 \\ & x_2 - x_1 \leq 1 \\ & x_1 + 6x_2 \leq 15 \\ & 4x_1 - x_2 \leq 10 \end{array} \quad \rightarrow \quad \begin{array}{rl} \text{löse:} & x_1 + x_2 \geq 3 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \\ & x_2 - x_1 \leq 1 \\ & x_1 + 6x_2 \leq 15 \\ & 4x_1 - x_2 \leq 10 \end{array}$$

Optimierung $\rightarrow$ gültige Lösung

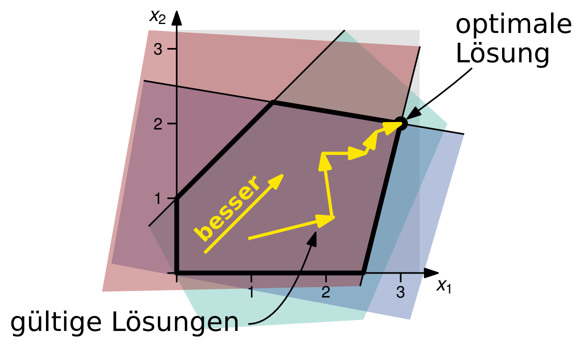
- gegeben: Algo, der gültige Lösung verbessert, bis sie optimal ist
- Idee: erlaube Verletzung der Ungleichungen; minimiere den Fehler
- Beispiel:

$$\begin{array}{rl} \text{löse:} & x_1 + 3x_2 - 2x_3 \leq -2 \\ & -2x_1 + x_2 + x_3 \leq -5 \\ & x_1, x_2, x_3 \geq 0 \end{array} \qquad \begin{array}{rl} \text{minimiere:} & \delta_1 + \delta_2 \\ \text{sodass:} & x_1 + 3x_2 - 2x_3 - \delta_1 \leq -2 \\ & -2x_1 + x_2 + x_3 - \delta_2 \leq -5 \\ & x_1, x_2, x_3, \delta_1, \delta_2 \geq 0 \\ \\ \text{Initiallösung:} & x_1, x_2, x_3 = 0, \delta_1 = 2, \delta_2 = 5 \end{array}$$

Algorithmen für LPs

Effizient lösbar in der Praxis

- Simplex-Verfahren
  - verbessert Lösung schrittweise
  - läuft auf dem Rand des Polytops
- Innere-Punkte-Verfahren
  - verbessert Lösung schrittweise
  - läuft im Inneren des Polytop



Effizient lösbar in der Theorie

- Innere-Punkte-Verfahren
  - polynomielle Laufzeit
- Ellipsoidmethode
  - findet eine gültige Lösung (wenn sie existiert)
  - kreist den Lösungsraum Schritt für Schritt weiter ein (mit Ellipsen)
  - polynomielle Laufzeit (aber langsam in der Praxis)
- Simplex-Verfahren
  - exponentielle worst-case Laufzeit
  - average-case: polynomiell
  - „smoothed analysis“

Oberer Schranken

Ziel: finde obere Schranken für die optimale Lösung

maximiere: $2x_1 + 3x_2$
sodass: $4x_1 + 8x_2 \leq 12$
$2x_1 + 1x_2 \leq 3$
$3x_1 + 2x_2 \leq 4$
$x_1, x_2 \geq 0$

- einfache Schranke: 12
  $2x_1 + 3x_2 \leq 4x_1 + 8x_2 \leq 12$
- besser: 6 ($\cdot \frac{1}{2}$)
  $2x_1 + 3x_2 \leq 2x_1 + 4x_2 \leq 6$
- Kombination mehrere Gleichungen: 5 ($\cdot \frac{1}{3}$, $\cdot \frac{1}{3}$)
  $2x_1 + 3x_2 \leq \frac{1}{3}(4x_1 + 8x_2 + 2x_1 + 1x_2) \leq \frac{1}{3}(12 + 3)$

Systematische Kombination mehrerer Gleichungen

- bestimme möglichst gute Faktoren y_1, y_2 und y_3 für die Gleichungen
- wir erhalten: $y_1(4x_1 + 8x_2) + y_2(2x_1 + 1x_2) + y_3(3x_1 + 2x_2) \leq 12y_1 + 3y_2 + 4y_3$
- umgestellt: $\underbrace{(4y_1 + 2y_2 + 3y_3)}_{\geq 2}x_1 + \underbrace{(8y_1 + 1y_2 + 2y_3)}_{\geq 3}x_2 \leq 12y_1 + 3y_2 + 4y_3$

(damit es eine Schranke liefert)

- also:

minimiere: $12y_1 + 3y_2 + 4y_3$
sodass: $4y_1 + 2y_2 + 3y_3 \geq 2$
$8y_1 + 1y_2 + 2y_3 \geq 3$
$y_1, y_2, y_3 \geq 0$

Duales Programm

- findet kleinste obere Schranke, die man so erhalten kann

warum $y_1, y_2, y_3 \geq 0$?

Matrixschreibweise

Primales Programm

maximiere: $2x_1 + 3x_2$

sodass: $4x_1 + 8x_2 \leq 12$

$2x_1 + 1x_2 \leq 3$

$3x_1 + 2x_2 \leq 4$

$x_1, x_2 \geq 0$

$$\text{maximiere } \begin{pmatrix} 2 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\text{mit } \begin{pmatrix} 4 & 8 \\ 2 & 1 \\ 3 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} 12 \\ 3 \\ 4 \end{pmatrix} \text{ und } \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \geq \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

allg.: maximiere $c^T x$ mit $Ax \leq b$ und $x \geq 0$

Duales Programm

minimiere: $12y_1 + 3y_2 + 4y_3$

sodass: $4y_1 + 2y_2 + 3y_3 \geq 2$

$8y_1 + 1y_2 + 2y_3 \geq 3$

$y_1, y_2, y_3 \geq 0$

$$\text{minimiere } \begin{pmatrix} 12 & 3 & 4 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

$$\text{mit } \begin{pmatrix} 4 & 2 & 3 \\ 8 & 1 & 2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \geq \begin{pmatrix} 2 \\ 3 \end{pmatrix} \text{ und } \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \geq \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

allg.: minimiere $b^T y$ mit $A^T y \geq c$ und $y \geq 0$

Beachte

- das duale vom dualen ist das primale Programm
- die Forderung $x \geq 0$ ist keine echte Einschränkung

Warum?

- duales Programm liefert eine obere Schranke (untere bei Minimierung)

Dualitätssatz

Theorem
Für die linearen Programme

$$\text{maximiere } c^T x \text{ mit } Ax \leq b \text{ und } x \geq 0 \text{ und} \tag{P}$$

$$\text{minimiere } b^T y \text{ mit } A^T y \geq c \text{ und } y \geq 0 \tag{D}$$

gilt genau eine der folgenden Aussagen:

- Weder (P) noch (D) hat eine gültige Lösung.
- (P) ist unbeschränkt und (D) hat keine gültige Lösung.
- (P) hat keine gültige Lösung und (D) ist unbeschränkt.
- (P) und (D) sind gültig und beschränkt. Das Maximum von (P) ist dann gleich dem Minimum von (D).

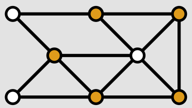
Beachte
- das duale Programm liefert also eine perfekte obere Schranke
- das LP muss nicht in der obigen Form vorliegen

ILPs

Ganzzahliges lineares Programm (ILP)

- genauso, wie LP, nur dass $x \in \mathbb{Z}^n$ gesucht wird, statt $x \in \mathbb{R}^n$
- macht das Problem NP-schwer

Problem: VERTEX COVER
Finde ein minimales Vertex Cover in einem Graphen $G = (V, E)$. (Knotenmenge $V' \subseteq V$ mit $e \cap V' \neq \emptyset$ für alle $e \in E$)



Beispiel: VERTEX COVER

- Variablen: x_v für jeden Knoten $v \in V$
- Bedeutung: $x_v = 1 \Leftrightarrow v \in V'$ (und $x_v = 0$ sonst)
- ILP:

$$\begin{aligned} \text{minimiere:} \quad & \sum_{v \in V} x_v \\ \text{sodass:} \quad & 0 \leq x_v \leq 1 \text{ für } v \in V \qquad (\Leftrightarrow x_v \in \{0, 1\}) \\ & x_u + x_v \geq 1 \text{ für } uv \in E \end{aligned}$$

LP-Relaxierung

- fasst man ein ILP als LP auf, so spricht man von der LP-Relaxierung
- die LP-Lösung (und auch zug. duale Lösung) liefert Schranke für ILP (insbesondere nützlich bei Approximation)
- manchmal liefert die LP-Relaxierung sogar die optimale Lösung

Lenstras Theorem

Theorem (ohne Beweis)
Für $A \in \mathbb{Z}^{m \times n}$ und $b \in \mathbb{Z}^m$ kann die Frage, ob es ein $x \in \mathbb{N}^n$ mit $Ax \leq b$ gibt in $O(n^{2,5n}|A, b|)$ entschieden werden, wobei $|A, b|$ die Länge der Binärkodierung für die Instanz bezeichnet.

Folgerungen

- ILP (als Entscheidungsproblem) mit Parameter n = Anzahl der Variablen, auch *Dimension* genannt, ist in FPT
- Anzahl der Ungleichungen geht nur polynomiell in Laufzeit ein
- Größe der Zahlen geht nur logarithmische in Laufzeit ein

Metatheorem
Ein parametrisiertes Problem mit Parameter k, dass sich als ILP mit $f(k)$ vielen Variablen darstellen lässt, ist in FPT.

Beispiel: Closest String

Problem: Closest String (*k* wird unser Parameter sein)
Gegeben k Strings $s_1, \dots, s_k \in \Sigma^n$ und $D \in \mathbb{N}$. Gibt es einen String $s \in \Sigma^n$, der von jedem s_i Hamming Distanz maximal D hat?

Beobachtung

- nur (Un)gleichheit innerhalb jeder Spalte relevant
- es gibt äquivalente Instanz mit $\Sigma = [k]$
- Anzahl verschiedener Spalten nur von k abhängig ($\leq k!$)

s_1	A	B	C	F	G	D	C	G	E	B	A
s_2	B	B	D	A	G	D	B	G	B	E	C
s_3	B	A	E	F	C	A	C	G	B	B	F

↓

s_1	1	1	1	1	1	1	1	1	1	1	1
s_2	2	1	2	2	1	1	2	1	2	2	2
s_3	2	2	3	1	2	2	1	1	2	1	3
s	2	1	3	1	2	1	2	1	2	1	2

0×1	2×2	0×3
2×1	1×2	0×3
0×1	1×2	1×3
2×1	1×2	0×3
1×1	0×2	0×3

Formulierung als ILP

- reduziere Lösung auf: Wie oft wurde für den Spalten-Typ t der Wert a gewählt? $\rightarrow$ Variable $x_{t,a}$
- wähle ein Zeichen für jede Spalte von Typ t:

$$\sum_{a \in \Sigma} x_{t,a} = \text{Anzahl Spalten von Typ } t$$

- Distanz $\leq D$ für jedes s_i: $\displaystyle\sum_{t \in \text{Typen}} \sum_{\substack{a \in \Sigma \\ a \neq t[i]}} x_{t,a} \leq D$
- $\rightarrow$ ILP mit „nur“ $k! \cdot k$ Variablen

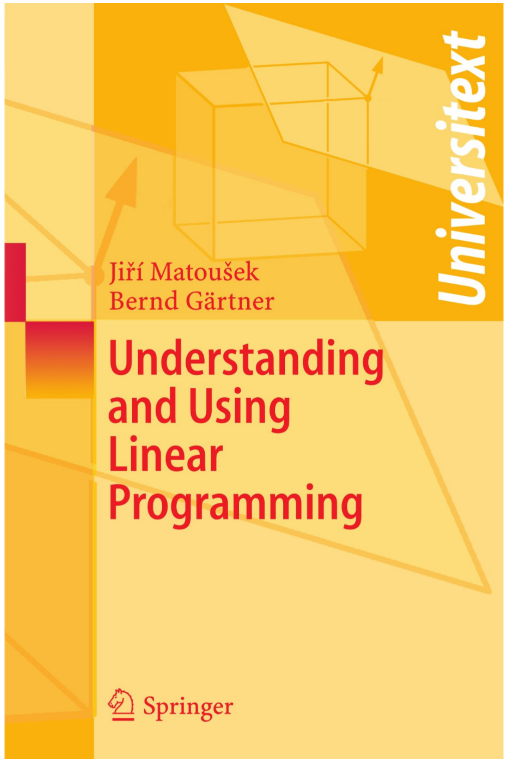
Zusammenfassung

Lineare Programme
- einfache Modellierung anderer Probleme
- verschiedene Arten der Normalisierung möglich
- manchmal helfen zusätzliche Variablen
- geometrische Interpretation
- gültige Lösungen vs. Optimalität

Dualität
- systematische Berechnung oberer Schranken
- Dualitätssatz (ohne Beweis)

ILP
- kann viele NP-harte Probleme modellieren
- Metatheorem für FPT aus Satz von Lenstra (ohne Beweis)

Literaturhinweise



Anmerkungen

- hervorragend geschrieben und schön kompakt
- aus dem Uninetz kostenlos abrufbar

`link.springer.com/book/10.1007/978-3-540-30717-4`

Integer Programming in Parameterized Complexity: Three Miniatures

- Tomás Gavenciak, Dusan Knop, Martin Koutecký [2019]
- guter Überblick über ILPs in der parametrisierten Welt, mit vielen Referenzen

`drops.dagstuhl.de/opus/volltexte/2019/10222/`