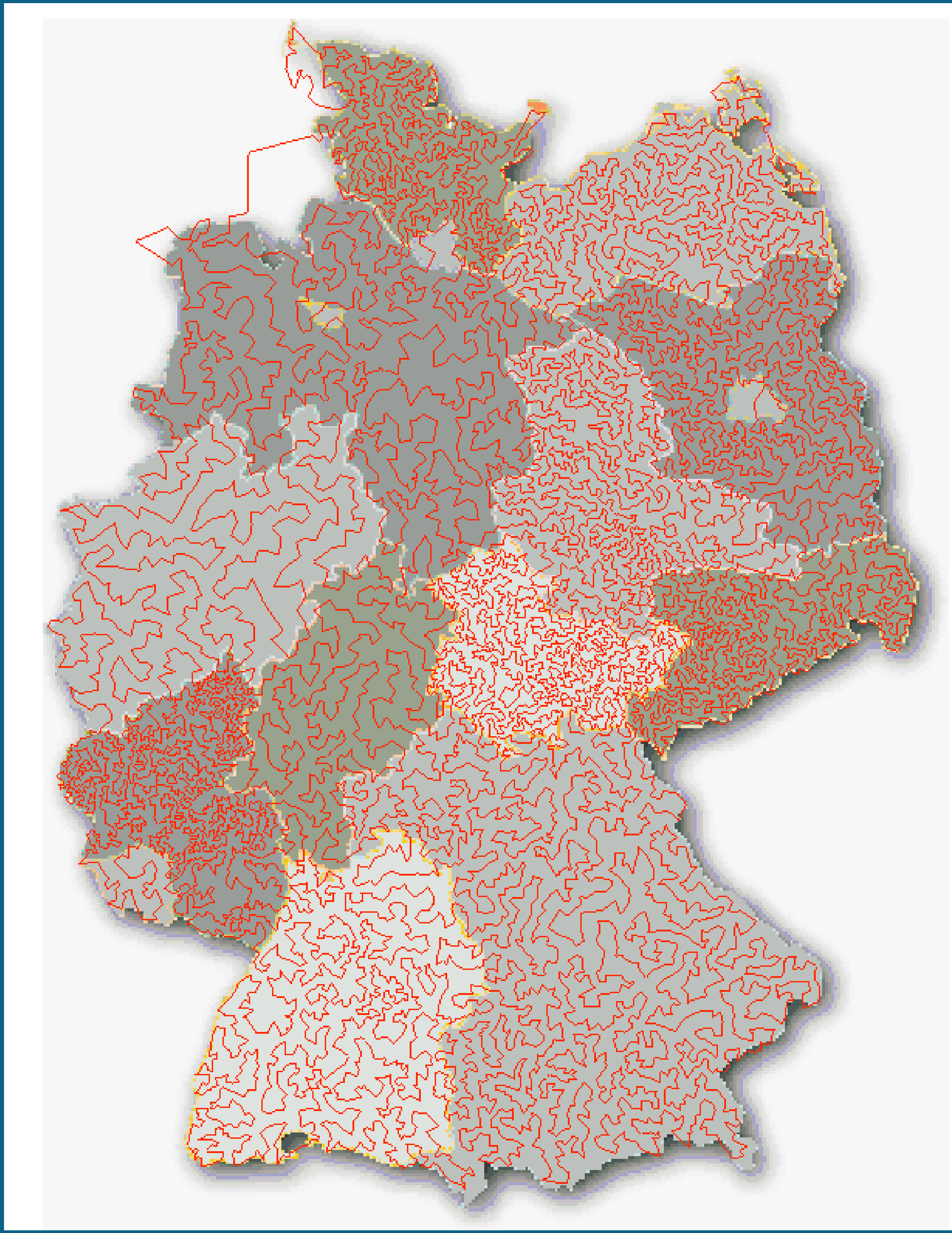
# Lineare Programmierung

Frank Schönmann

WS 2003/04

# Rolle der Linearen Programmierung für das TSP

- 1954: Dantzig, Fulkerson & Johnson lösen das TSP für 49 US-Städte (ca. $6.2 \cdot 10^{60}$ mögliche Touren)
- 1998: 13.509 Städte in den USA
- aktueller Weltrekord: Applegate, Bisby, Chvátal & Cook lösen das TSP für 15.112 Städte in Deutschland (ca. $7.3 \cdot 10^{56593}$ mögliche Touren)
- berechnet mit *cutting planes*-Methoden, basierend auf linearer Programmierung



# Überblick

- Einführung in die Lineare Programmierung
- Algorithmen zur Lösung von LP-Problemen
- Formulierung des TSP als LP-Problem
- *cutting planes*
- Zusammenfassung

# Beispiel (I)

- Ein Politiker möchte seine Wahlkampfgelder möglichst effizient einsetzen und in Stadt (100.000), Vorstadt (200.000) und Land (50.000) jeweils eine absolute Mehrheit erringen.
- Gewinne/Verluste an Wählerstimmen pro \$1000:

| Wahlkampfthema | Variable | Stadt | Vorstadt | Land |
|---|---|---|---|---|
| Straßenbau | $x_1$ | $-2$ | $5$ | $3$ |
| Waffenkontrolle | $x_2$ | $8$ | $2$ | $-5$ |
| Subventionen | $x_3$ | $0$ | $0$ | $10$ |
| Mineralölsteuer | $x_4$ | $10$ | $0$ | $-2$ |

# Beispiel (II)

- minimiere die Ausgaben:

$$x_1 + x_2 + x_3 + x_4$$

- Bedingungen für Mindestanzahl an Wählerstimmen:

$$\begin{array}{rcrcrcrcr} -2x_1 & + & 8x_2 & + & 0x_3 & + & 10x_4 & \geq & 50 \\ 5x_1 & + & 2x_2 & + & 0x_3 & + & 0x_4 & \geq & 100 \\ 3x_1 & - & 5x_2 & + & 10x_3 & - & 2x_4 & \geq & 25 \end{array}$$

$$x_1, x_2, x_3, x_4 \geq 0$$

# Lineare Programmierung

- Grundidee: optimiere eine lineare Funktion mit $n$ Freiheitsgraden, die durch lineare Gleichungen/Ungleichungen (evtl. widersprüchliche Bedingungen und beschränkte Ressourcen) eingeschränkt ist
- sehr allgemeine Methode zur Optimierung von Problemen, für die keine speziell entwickelten Algorithmen vorhanden sind
- eines der Hauptverfahren des *Operations Research*

# Anwendungsbeispiele

- möglichst effiziente Verteilung von Crews einer Airline auf alle möglichen Flüge unter Einhaltung gesetzlicher Bestimmungen, z. B. maximale Arbeitszeit und Verteilung auf Flugzeugtypen
- Entscheidung für Öl-Bohrstellen gemäß erwarteter Öl-Fördermenge und Kosten für Bohrungen am jeweiligen Ort
- Graphentheoretische Probleme: kürzeste Pfade, maximaler Fluss, minimaler Kostenfluss, . . .

# Formale Formulierung (Standardform)

- maximiere:

$$f(x_1, \ldots, x_n) = \sum_{j=1}^{n} c_j x_j$$

- Bedingungen:

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad \text{für } i = 1, 2, \ldots, m$$

$$x_j \geq 0 \qquad \text{für } j = 1, 2, \ldots, n$$

# . . . in Matrixschreibweise

- maximiere:

$$f(\mathbf{x}) = \mathbf{c}^T\mathbf{x}$$

- Bedingungen:

$$A\mathbf{x} \leq \mathbf{b}$$
$$\mathbf{x} \geq 0$$

$\Rightarrow$ Lineares Problem $(A, \mathbf{b}, \mathbf{c})$ mit $A \in \mathbb{R}^{m\times n}, \mathbf{b} \in \mathbb{R}^m$ und $\mathbf{c}, \mathbf{x} \in \mathbb{R}^n$

# Normalformen

- äquivalente Formulierungen für jedes lineare Programm
- Standardform: alle Bedingungen sind Ungleichungen
- Slackform: alle Bedingungen sind Gleichungen (außer Nichtnegativitätsbedingungen)

# Überführung in Standardform (I)

1. Minimierungsproblem statt Maximierungsproblem
2. Variablen ohne Nichtnegativitätsbedingung
3. Gleichheitsbedingungen statt Ungleichheitsbedingungen
4. Größer-als-Bedingungen statt Kleiner-als-Bedingungen

$\Rightarrow$ finde ein äquivalentes LP-Problem in Standardform

# Überführung in Standardform (II)

1. negiere die Koeffizienten $\mathbf{c}$ in der Zielfunktion
2. ersetze $x_j$ durch $x'_j - x''_j$ mit $x'_j, x''_j \geq 0$
3. ersetze $f(\mathbf{x}) = b$ durch $f(\mathbf{x}) \leq b, f(\mathbf{x}) \geq b$
4. negiere die Koeffizienten $a_{ij}, b_i$ in betreffender Bedingung $i$

# Überführung in Slackform

- die folgenden Ausdrücke sind äquivalent:

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i \qquad \Longleftrightarrow \qquad 0 \leq s_i = b_i - \sum_{j=1}^{n} a_{ij}x_j$$

- $s_i$ heißt Slack-Variable
- man erhält außer den Nichtnegativitätsbedingungen nur noch Gleichungen als Nebenbedingungen

# Geometrische Interpretation

- Lineare Gleichung beschreibt eine Hyperebene $\Rightarrow$ lineare Ungleichung beschreibt einen Halbraum
- Lineares Ungleichungssystem beschreibt daher den Schnitt der einzelnen Halbräume $\Rightarrow$ konvexes Polytop (Simplex)
- Lösung ist dann der erste Schnitt einer sich nähernden Hyperebene (Zielfunktion) $\Rightarrow$ Eckpunkt oder Seite des Polytops

# Beispiel (Problem)

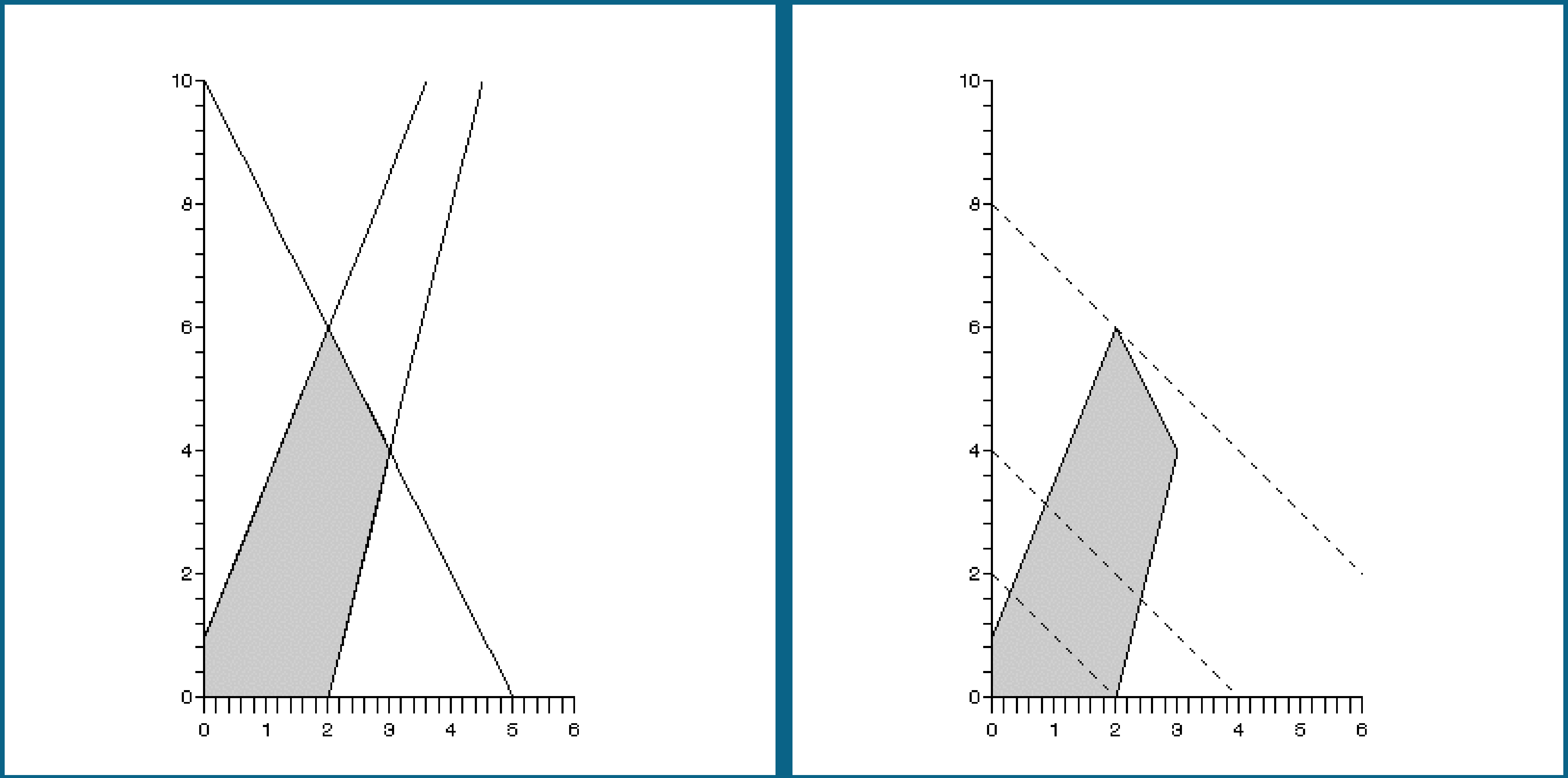
- maximiere (Zielfunktion):

$$x_1 + x_2$$

- Bedingungen:

$$\begin{array}{rcrcr} 4x_1 & - & x_2 & \leq & 8 \\ 2x_1 & + & x_2 & \leq & 10 \\ 5x_1 & - & 2x_2 & \geq & -2 \\ & x_1, x_2 & & \geq & 0 \end{array}$$

# Beispiel (Darstellung)



# Algorithmen zur Lösung von LP-Problemen

- es existieren effiziente Algorithmen: Ellipsoidverfahren von Khachian in $O(n^6)$, *interior-point*-Methode von Karmarkar in $O(n^{3.5})$
- in der Praxis: Simplex-Algorithmus, aber im *worst case* nicht mehr polynomiell
- Simplex-Algorithmus wurde 1947 von Dantzig vorgestellt

# Simplex-Algorithmus (anschaulich)

- erhält als Eingabe ein lineares Programm (in Slackform) und berechnet eine optimale Lösung
- beginnt an einer Ecke des Polytops und führt eine Kette von Iterationen durch, bei denen der Algorithmus jeweils zu einer benachbarten Ecke springt, deren Wert nicht kleiner als der aktuelle ist
- der Simplex-Algorithmus terminiert, sobald er ein lokales Maximum findet (dies ist global wg. Konvexität des Polytops und Linearität der Zielfunktion)

# Spezialfall: IP

- diskrete lineare Programmierung (*integer linear programming*)
- zusätzlich zu den Einschränkungsungleichungen muss gelten $\mathbf{x} \in \mathbb{Z}^n$
- Spezialfall: binäre diskrete lineare Programmierung (*0/1 integer linear programming*) mit $\mathbf{x} \in \{0, 1\}^n$
- nicht effizient lösbar $\Rightarrow \mathcal{NP}$-hart

# Formulierung des TSP als LP-Problem (I)

- Grundlage: TSP mit $n$ Städten, spezifiziert durch einen Kostenvektor $\mathbf{c}$
- jede Tour kann durch einen Inzidenzvektor $\mathbf{x} \in \mathcal{S}$ dargestellt werden

$$\mathbf{c} \in \mathbb{R}^{n(n-1)/2} \qquad \mathbf{x} \in \mathcal{S} \subset \{0,1\}^{n(n-1)/2}$$

- Problem:

$$\text{minimiere } \mathbf{c}^T\mathbf{x} \text{ mit } \mathbf{x} \in \mathcal{S} \tag{1}$$

$\Rightarrow$ nicht ohne weiteres lösbar

# Formulierung des TSP als LP-Problem (II)

- *Relaxation*: Formulierung eines verwandten Problems

$$\text{minimiere } \mathbf{c}^T\mathbf{x} \text{ mit } A\mathbf{x} \leq \mathbf{b} \tag{2}$$

- $A\mathbf{x} \leq \mathbf{b}$ ist ein lineares Ungleichungssystem, das von allen $\mathbf{x} \in \mathcal{S}$ erfüllt wird
- optimale Lösung von (2) ist eine untere Schranke für (1)

$\Rightarrow$ einfach zu lösendes LP-Problem $(A, \mathbf{b}, \mathbf{c})$

# Ausnutzung des verwandten LP-Problems

- findet eine untere Schranke für das TSP
- außerdem: findet man eine Lösung für Problem (2), kann man in der Nähe davon nach einer Lösung für Problem (1) suchen

# Initiales LP-Problem

- Inzidenzvektor:

$$0 \leq x_e \leq 1 \tag{3}$$

- jede Stadt $v$ wird von genau zwei Kanten besucht:

$$\sum \{x_e \mid v \in e\} = 2 \tag{4}$$

- diese Bedingungen werden von allen $\mathbf{x} \in \mathcal{S}$ erfüllt und können daher als initiales LP-Problem verwendet werden

# cutting planes

- hat (2) eine optimale Lösung und das Polytop $\{\mathbf{x} \mid A\mathbf{x} \leq \mathbf{b}\}$ hat einen Extrempunkt, dann findet ein LP-Algorithmus diesen Punkt als eine optimale Lösung $\mathbf{x}^*$ (z. B. der Simplex-Algorithmus)

- falls $\mathbf{x}^*$ nicht in $\mathcal{S}$ liegt, existiert eine lineare Ungleichung, die von allen Punkten in $\mathcal{S}$ erfüllt wird und von $\mathbf{x}^*$ nicht

  diese Ungleichung nennt man *cutting plane* oder *cut* und fügt sie zum Ungleichungssystem $A\mathbf{x} \leq \mathbf{b}$ hinzu

- dieser Schritt wird wiederholt bis $\mathbf{x}^* \in \mathcal{S}$

# Methoden zur Generierung von cutting planes

$\Rightarrow$ Schwierigkeit ist das automatische Finden von *cuts* bis zur optimalen Lösung für das TSP

- *Gomory cuts*: 1958 vorgestellt, werden heute aber nicht mehr verwendet
- Hypergraph-Schnitte: *subtour inequalities*, *blossom inequalities*, *comb inequalities*
- Hypergraph ist die Verallgemeinerung eines Graphen mit Kanten über beliebig viele Knoten

# Subtour inequalities

- Anzahl der Verbindungskanten zwischen $S$ und $T$:

$$x(S,T) = \sum \{x_e \mid e \cap S \neq \emptyset \neq e \cap T\}$$

- Trennt man $V$ in zwei echte Teilmengen $S$ und $T$, so muss jede Tour $S$ und $T$ durch mindestens zwei Kanten verbinden:

$$x(S, V - S) \geq 2 \tag{5}$$

# Zusammenfassung

- Lineare Programmierung ist das Standardverfahren zur exakten Lösung größerer TSP-Instanzen
- TSP ist ein Problem der (binären) diskreten linearen Programmierung ($\mathcal{NP}$-hart)
- Lösung durch iteratives Beschneiden eines hochdimensionalen Polytops mit geeigneten *cut*-Bedingungen/-Ungleichungen