Traveling Salesman Problem (TSP)

Exakte Algorithmen für NP-schwere Probleme
Integer Lineare Programme
Branch-and-Cut

VO Graphenalgorithmen – WiSe 2009/10

Markus Chimani – TU Dortmund

NP-schwere Probleme...

2

Probleme der kombinatorischen Optimierung
>> Wie ist das formal definiert?

Wenn P≠NP:
Lösen von NP-schweren Problemen benötigt **im Allgemeinen** **exponentiell** viel Zeit.

Traditionelle Algorithmik
Polynomielle Zeit...

- Heuristiken: finde eine Lösung die hoffentlich möglichst gut ist
- Approximationen (PTAS, FPTAS, ...): garantiere dabei eine minimale Güte der Lösung (in Relation zum Optimum)
- FPT (fixed parameter tractable)

Aber: Veröffentlichungen in den letzten Jahren/-zehnten...

NP-schweres Problem ***XY***
in der Praxis beweisbar-optimal lösbar –
mit Hilfe von **Mathematischer Programmierung**

Übersicht

3

- Was sind **kombinatorische Optimierungsprobleme** (formal)?
- Was ist **Mathematische Programmierung**?
- Was sind **Integer Lineare Porgamme** (ILPs)?
- Wie kann ich **ILPs lösen**?
- **Wozu** kann ich die benutzen? Wozu nicht?

Allgemein vs. Beispiel >> Traveling Salesman Problem (TSP)

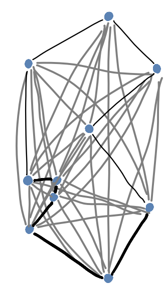
- **Warum gerade TSP?**
  - Der "Urvater" der Probleme bei denen sich ILPs als erfolgreich erwiesen haben.
  - Haufenweise Forschungsergebnisse, Studien, Literatur,...

Traveling Salesman Problem (TSP)

4

(Symmetrisches) Travelling Salesman Problem
Gegeben: **n** Städte und Distanzen zwischen ihnen.
Gesucht: Kürzerste Rundtour in der jede Stadt genau einmal besucht wird

Als Graphproblem
Gegeben: Vollständiger Graph **K_n=G=(V,E)** mit Kantengewichten **w: E→$\mathbb{R}^+$**
Gesucht: Hamiltonkreis **C ⊆ E** mit min $\Sigma_{e\in C}$**w(e)**



Anwendungen

- Routing, Schweißen, Robotik, u.v.m.
- Basis für viele weitere Probleme (z.B. Vehicle Routing)

Traveling Salesman Problem (TSP)

5

1954
[Dantzig **(Simplex-Algo!)**, Fulkerson, Johnson]
49 Städte (Hauptstädte der US Bundesstaaten + Washington, DC)
beweisbar-optimale Lösung mit Hilfe von ILPs
manuell gerechnet!

"Geburt" von **Polyedrischer Kombinatorik** und **Schnittebenen-Verfahren** (Cutting-Planes).
viele Ideen (z.B. **Branch-and-Cut**) die erst später (von anderen) detailliert ausgearbeitet wurden

Heute
Exakte Algorithmen sogar für riesig große Instanzen:

Sweden	24.978	✓
VLSI	85.900	✓
World TSP	1.904.711	<0.05%

George B. Dantzig
TSP: 25.000 Städte
(von Robert Bosch)

Lineare Programme

6

Lineares Programm (LP)
Variablen,
eine lineare Zielfunktion (**objective function**),
mehrere linearen Nebenbedingungen (**constraints**)

(Sinnloses) Beispiel

max $1\cdot x_1 + 4\cdot x_2 - 2\cdot x_3$
s.t. $x_1 + x_2 \leq 4$ (s.t. = "subject to")
$x_3 \geq x_1$
$x_1, x_2, x_3 \geq 0$

Normierte Matrixdarstellung

min $c^T x$
s.t. $A x \leq b$
$x \geq 0$

x = **n**-dimensionaler Variablenvektor
c = **n**-dimensionaler Kostenvektor
A = **n**x**m** Constraint-Matrix
b = **m**-dimensionaler Right-Hand-Side

dominierende Größe: Spalten und Zeilen in der Constraint-Matrix

Integer Lineare Programme

Integer Lineares Programm (ILP)

Variablen die ganzzahlig sein müssen,
eine lineare Zielfunktion (**objective function**),
mehrere linearen Nebenbedingungen (**constraints**)

(Sinnloses) Beispiel

max $1 \cdot \mathbf{x_1} + 4 \cdot \mathbf{x_2} - 2 \cdot \mathbf{x_3}$

s.t. $\mathbf{x_1} + \mathbf{x_2} \leq 4$ (s.t. = "subject to")

$\mathbf{x_3} \geq \mathbf{x_1}$

$\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3} \in \mathbb{N}$

Normierte Matrixdarstellung

min $c^T\mathbf{x}$

s.t. $A\,\mathbf{x} \leq b$

$\mathbf{x} \geq \mathbf{0}$

$\mathbf{x} \in \mathbb{Z}$

x = **n**-dimensionaler Variablenvektor
c = **n**-dimensionaler Kostenvektor
A = **n**x**m** Constraint-Matrix
b = **m**-dimensionaler Right-Hand-Side

dominierende Größe: Spalten und Zeilen in der Constraint-Matrix

TSP als ILP

Variablen

$\forall e \in E$: $\mathbf{x_e} \in \begin{cases} 0 \text{ ... Kante } e \text{ wird nicht benutzt} \\ 1 \text{ ... Kante } e \text{ wird benutzt} \end{cases}$

Zielfunktion

min $w^T\mathbf{x} = \Sigma_{e \in E}\, w_e \cdot \mathbf{x_e}$

Constraints

$\forall v \in V$: $\mathbf{x}(\delta(v)) = 2$ (Grad Ungleichungen)

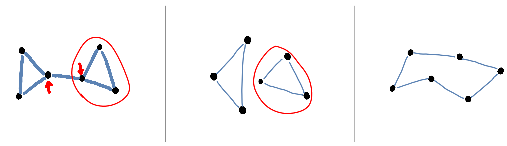
$\forall \emptyset \neq W \subset V$: $\mathbf{x}(\delta(W)) \geq 2$ (Undirected-Cut bzw. Subtour Ungleichungen)

Notation

$\boldsymbol{\delta(v)} := \{\, \{v,u\} \in E \,\}$
= zu v inzidente Kanten

$\mathbf{x(F)} := \Sigma_{e \in F}\, \mathbf{x_e}$
für $F \subseteq E$

$\boldsymbol{\delta(W)} := \{\, \{v,u\} \in E \mid v \in W,\ u \notin W \,\}$
für $W \subseteq V$



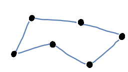
TSP als ILP

min $w^T\mathbf{x}$

s.t. $\mathbf{x}(\delta(v)) = 2$ $\forall v \in V$

$\mathbf{x}(\delta(W)) \geq 2$ $\forall \emptyset \neq W \subset V$

$\mathbf{x} \in \{0,1\}^{|E|}$



Theorem

Jede **gültige** (feasible) Lösung dieses ILPs mit Zielfunktionswert z entspricht einer gültigen TSP-Route der Länge z.

Korrolar

Eine **optimale** (d.h. minimale) Lösung dieses ILPs mit Zielfunktionswert z entspricht einer optimalen (d.h. kürzesten) TSP-Route der Länge z.

Allgemein: Kombinatorische Opt.Prob.

Kombinatorisches Optimierungsproblem

Gegeben: Grundmenge M, Kostenfunktion $c: M \to \mathbb{R}$

Gesucht: Teilmenge $L \subseteq M$ mit minimalem/maximalem $\Sigma_{i \in L}\, c(i)$ die eine Eigenschaft $\mathcal{E}$ erfüllt.

Darstellung der Eigenschaft $\mathcal{E}$

Potenzmenge $\mathcal{P} = 2^M = \{N \mid N \subseteq M\}$

Gültige (feasible) Teilmengen $\mathcal{F} \subseteq \mathcal{P}$

Dann: $L \in \mathcal{F}$

Also

$L = \operatorname{argmin}_{L' \in \mathcal{F}} \{\, c(L') \,\}$

Allgemein: Komb.Opt.Prob. als 0/1-ILP

Theorem

Jedes kombinatorische Optimierungsproblem kann als binäres (0/1) ILP formuliert werden.

Variablen

$\forall i \in M$: $x_i \in \{\mathbf{0},\mathbf{1}\}$ → **1** wenn i in L sein soll; **0** sonst

Zielfunktion

min/max $c^T x$

Constraints

$\forall N \in \mathcal{P} \setminus \mathcal{F}$: $\Sigma_{i \in N}\, x_i + \Sigma_{i \in M \setminus N}\, (1 - x_i) \leq |M| - 1$

→ exponentiell viele und sehr schwach

→ **bessere Constraints?**

LP vs. ILP

Komplexität

- **LPs** lassen sich in **polynomieller Zeit** (in **n,m**) lösen
  - mit Ellipsoid-Methode, Innere-Punkte-Methode;
  - Simplex-Algorithmus im worst exponentiell, aber in der Praxis top
- **ILPs** sind **NP-schwer** zu lösen

Idee

Wieviel verliert man, wenn man statt **ILP** nur **LP** rechnet?

Definition LP Relaxierung

Das LP das entsteht, wenn man aus einem ILP die Ganzzahligkeitsbedingung streicht.

TSP		
min	$w^T\mathbf{x}$	
s.t.	$\mathbf{x}(\delta(v)) = 2$	$\forall v \in V$
	$\mathbf{x}(\delta(W)) \geq 2$	$\forall \emptyset \neq W \subset V$
	$\cancel{\mathbf{x_e} \in \{0,1\}}$	$\cancel{\forall e \in E}$
	$0 \leq \mathbf{x_e} \leq 1$	$\forall e \in E$

Eigenschaften einer LP-Relaxierung

13

Lösung $\mathcal{L}$ der **LP Relaxierung** =
Duale Schranke für das ILP

Dual? **Minimierung** → **untere** Schranke
Maximierung → **obere** Schranke

TSP

min	$w^T \mathbf{x}$	
s.t.	$\mathbf{x}(\delta(v)) = 2$	$\forall v \in V$
	$\mathbf{x}(\delta(W)) \geq 2$	$\forall \emptyset \neq W \subset V$
	~~$\mathbf{x}_e \in \{0,1\}$~~	~~$\forall e \in E$~~
	$0 \leq \mathbf{x}_e \leq 1$	$\forall e \in E$

Schwächere Bedingungen → „bessere" (in Bezug auf Zielfunktion) Lösungen

$\mathcal{L}$ ist i.A. **infeasible** für das ILP wegen **fraktionalen Werten**.
Falls $\mathcal{L}$ „zufällig" ganzzahlig → optimale Lösung!

Primale Schranken = feasible Lösungen die aber nicht unbedingt optimal sind (Heuristiken, Approximationsalgorithmen, etc.)

(Einfacher) Branch-and-Bound Algorithmus (für 0/1-ILPs)

- Rechne LP Relaxierung.
- **Nicht ganzzahlig?** 2 Subprobleme: Fixiere fraktionale Variable auf 0 und 1).
  Berechne jeweils wieder LP Relaxierungen, …..
- (Globale) primale Schranke kleiner als (lokale) duale Schranke?
  → Subproblem verwerfen

Schnittebenenverfahren

14

LP lösen: O(poly(#variablen,#constraints))

Problem
Was wenn #constraints exponentiell in V?

TSP

min	$w^T \mathbf{x}$	
s.t.	$\mathbf{x}(\delta(v)) = 2$	$\forall v \in V$
	$\mathbf{x}(\delta(W)) \geq 2$	$\forall \emptyset \neq W \subset V$
	~~$\mathbf{x}_e \in \{0,1\}$~~	~~$\forall e \in E$~~
	$0 \leq \mathbf{x}_e \leq 1$	$\forall e \in E$

Lösung: Schnittebenenverfahren

1. Starte mit einer „aktiven" Teilmenge C_{active} aller Ungleichungen C (z.B. nur Gradungleichungen).
2. Löse LP Relaxation mit C_{active} → Lösung L.
3. **Separation:** Finde (einige) Ungleichungen $C_{violated}$ in $C \backslash C_{active}$, die durch L verletzt werden.
4. Falls $C_{violated} \neq \emptyset$: Füge $C_{violated}$ zu C_{active} hinzu. Goto 2.
5. Gesamt LP (für alle Constraints C) ist optimal gelöst.

Theorem [Grötschel, Lovasz, Schrijver]
Ein Optimierungsproblem ist in polynomieller Zeit lösbar genau dann wenn das zugehörige Separationsproblem in polynomieller Zeit lösbar ist.

Separationsalgorithmus

15

1. Starte mit einer „aktiven" Teilmenge C_{active} aller Ungleichungen C (z.B. nur Gradungleichungen).
2. Löse LP Relaxation mit C_{active} → Lösung L.
3. **Separation:** Finde (einige) Ungleichungen $C_{violated}$ in $C \backslash C_{active}$, die durch L verletzt werden.
4. Falls $C_{violated} \neq \emptyset$: Füge $C_{violated}$ zu C_{active} hinzu. Goto 2.
5. Gesamt LP (für alle Constraints C) ist optimal gelöst.

Separationsalgorithmus
$C_{violated}$ ist i.A. **nicht** die Menge **aller** verletzten Ungleichungen (exponentiell!)
Exakter Sep.Alg.: findet mindestens eine verletzte Ungleichungen, falls nicht alle Ungleichungen erfüllt sind.
Heuristischer Sep.Alg.:
L ganzzahlig: **exakt** (= Feasibility-Test einer Lösung)
L fraktional: Liefert möglicherweise $C_{violated} = \emptyset$, obwohl verletzte Ungleichungen existieren.
→ In Schritt 5: Nur untere Schranke für Lösung des Gesamt-LPs

Branch-and-Cut Algorithmus

16

Branch-and-Cut Algorithmus
Kombination aus Branch-and-Bound Algorithmus und Schnittebenenverfahren
Einfachste Form: Branch-and-Bound Algorithmus in dem die LPs jeweils mit Schnittebenenverfahren gelöst werden
Kompliziertere Formen: Bei den LP Berechnungen noch weitere Constraints hinzufügen, die gültig aber nicht "eigentlicher" Teil der Relaxierung sind.

Beobachtung
Für einen **exakten Branch-and-Cut Algorithmus** genügt ein **heuristischer Separationsalgorithmus**.
Der Fall $C_{violated} = \emptyset$ trotz verletzter Ungleichungen tritt nur bei fraktionalen Lösungen auf. → Unnötig früher Branch, aber Gültigkeit bleibt erhalten

Faustregel
Einen Branch nur machen wenn unbedingt nötig!
– In dieser Operation steckt die Exponentialität des Verfahrens.

Separation bei TSP

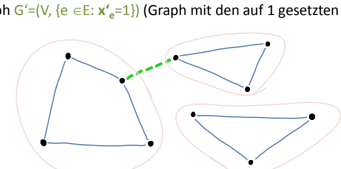
17

Wie separiert man bei TSP?
Lösung ist eine Belegung **x'** der Variablen **x**

TSP

min	$w^T \mathbf{x}$	
s.t.	$\mathbf{x}(\delta(v)) = 2$	$\forall v \in V$
	$\mathbf{x}(\delta(W)) \geq 2$	$\forall \emptyset \neq W \subset V$
	~~$\mathbf{x}_e \in \{0,1\}$~~	~~$\forall e \in E$~~
	$0 \leq \mathbf{x}_e \leq 1$	$\forall e \in E$

Ganzzahlige Lösung ($\forall e \in E$: $\mathbf{x}'_e \in \{0,1\}$)
Hilfsgraph $G'=(V, \{e \in E: \mathbf{x}'_e=1\})$ (Graph mit den auf 1 gesetzten Kanten)



(2-)Zusammenhangstest

Separation bei TSP

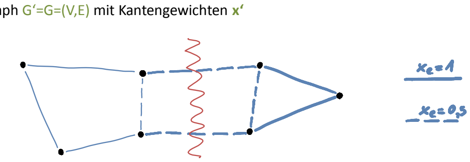
18

Wie separiert man bei TSP?
Lösung ist eine Belegung **x'** der Variablen **x**

TSP

min	$w^T \mathbf{x}$	
s.t.	$\mathbf{x}(\delta(v)) = 2$	$\forall v \in V$
	$\mathbf{x}(\delta(W)) \geq 2$	$\forall \emptyset \neq W \subset V$
	~~$\mathbf{x}_e \in \{0,1\}$~~	~~$\forall e \in E$~~
	$0 \leq \mathbf{x}_e \leq 1$	$\forall e \in E$

Fraktionale Lösung ($\exists e \in E$: $0 < \mathbf{x}'_e < 1$)
Hilfsgraph $G'=G=(V,E)$ mit Kantengewichten **x'**



Minimum Cut muss größer 2 sein.

Separation durch Minimum Cut

Separationsalgorithmus

1. Betrache Graph G mit Kantengewichten x'
2. Berechne Minimum Cut in (G,x')
   → Cutmenge W, Cutkanten F={ (u,v) ∈E | u∈W, v∈V\W }, Cutgröße γ=|F|
3. Falls γ<2:
   Erzeuge Constraint $x(F) \geq 2$ (äquivalent zu $x(\delta(W)) \geq 2$)
4. Sonst:
   x' erfüllt alle Undirected-Cut Ungleichungen

Theorem

Obiger Algorithmus ist ein exakter Separationsalgorithmus

Beweis

1. Er erzeugt nur verletzte Constraints. ✓
2. Er findet immer ein verletztes Constraint, so mindestens eines existiert.
   Annahme: $\exists \emptyset \neq W' \subset V$ mit $\gamma' = x'(\delta(W')) < 2$ → Die Kanten $\delta(W')$ sind ein Cut in G.
   → Minimum Cut $\gamma \leq \gamma'$ → Algorithmus erzeugt ein Constraint ✓

Exaktes Berechnen von TSP mit B&C

LP_A := LP Relaxierung des TSP-ILPs nur mit den Constraints A.

TSP:
$\min w^T x$
$x(\delta(v)) = 2 \quad \forall v \in V \quad (1)$
$x(\delta(W)) \geq 2 \quad \forall \emptyset \neq W \subset V \quad (2)$
$\cancel{x_e \in \{0,1\} \quad \forall e \in E}$
$0 \leq x_e \leq 1 \quad \forall e \in E \quad (3)$

- ub := ∞ oder **Heuristik()** //Obere Schranke
- A' := Ungleichungen der Klassen (1) und (3)
- Subproblem-Menge S := { $LP_{A'}$ }
- Solange S≠∅:

0.	Wähle und entferne ein beliebiges LP_A aus S	
1.	Löse LP_A → Lösung **x'** mit Zielfunktionswert **z**	
2.	ub := min{ ub, **Heuristik(x')** }	//Exploit (falls gewünscht)
3.	Falls **z** ≥ ub: **continue**	//Teilproblem uninteressant
4.	Finde verletzte Undirected-Cut Ungl. (2) → B	//Separierung
5.	Falls B≠∅: A := A∪B. Goto 1.	//Cut
6.	Falls **x'** ganzzahlig: ub := min{ ub, **z** }. **continue**	//Teilproblem optimal
7.	Wähle eine fraktionale Variable x_e ($0 < x'_e < 1$), S := S ∪ { $LP_{A\cup\{xe=0\}}$, $LP_{A\cup\{xe=1\}}$ }	//Branch

- **Optimaler Zielfunktionswert:** ub

Weitere Constraint-Klassen

Komplexität des Separierungsproblems…

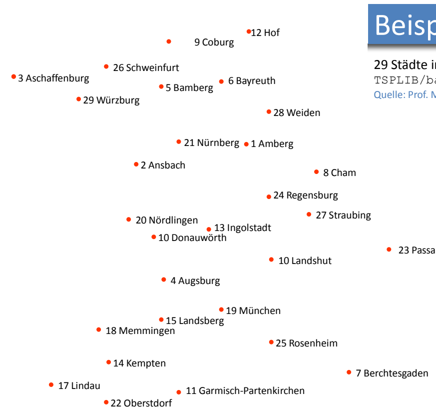
… polynomiell:

- Grad Gleichungen ✓
- Undirected-Cut / Subtour Ungleichungen ✓
- Subtour Eliminationsungleichungen >> Übung
- 2-Matching Ungleichungen >> Übung

… offen:

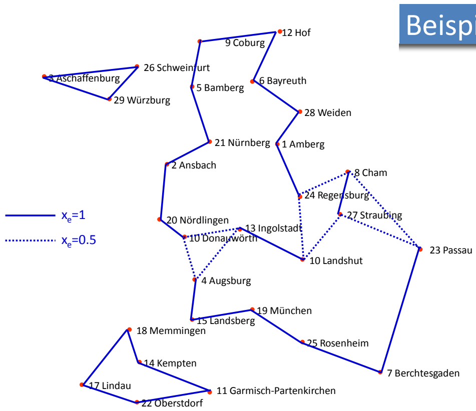
- Blossom Ungleichungen
- Comb/Kamm Ungleichungen >> Übung
- Cliquenbaum Ungleichungen
- Hypergraph Ungleichungen
- etc.

Beispiel

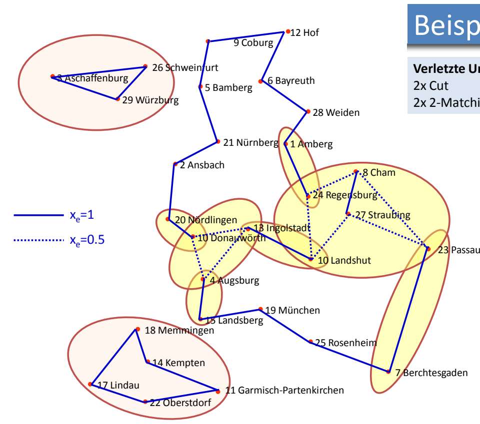


29 Städte in Bayern
`TSPLIB/bayg29.tspx`
Quelle: Prof. Mutzel

Beispiel



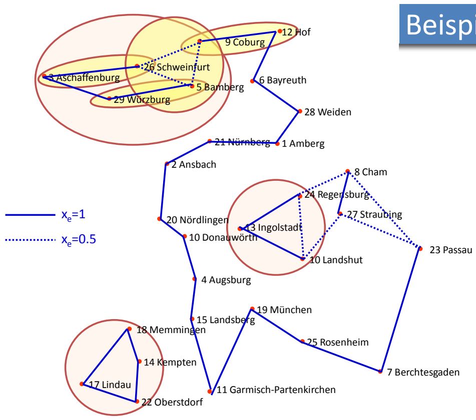
Beispiel



Verletzte Ungleichungen:
2x Cut
2x 2-Matching

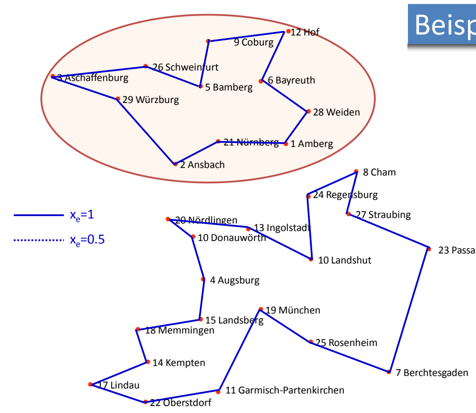
Beispiel

25



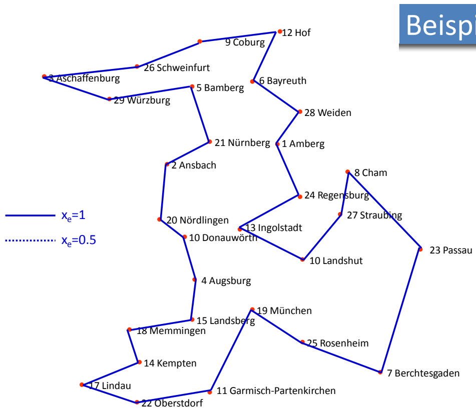
Beispiel

26



Beispiel

27



TSP

28

Damit: **Kapitel fertig!**
(Aber am Ende des Semesters werden wir nochmals ILPs in anderem Zusammenhang sehen)

Nächste Wochen: Frohe Weihnachten und guten Rutsch!

Nächster Termin: **6. Januar**
Bernd Zey wird die Vorlesung halten!

Neues Thema: Biconnectivity Augmentation