

Dualität

Primales Problem

$$\max_{Ax \leq b} c^T x \quad (P)$$

äquivalent: ($z \in \mathbb{R}$)

$$\max z \\ z - c^T x \leq 0 \\ Ax \leq b$$

$$P_z := \left\{ x \in \mathbb{R}^n : \begin{pmatrix} -c^T \\ A \end{pmatrix} x \leq \begin{pmatrix} -z \\ b \end{pmatrix} \right\}$$

- 145 -

$$\begin{aligned} & \max\{c^T x : Ax \leq b\} \\ &= \max\{z : z - c^T x \leq 0, Ax \leq b\} \\ &= \max\{z : P_z \neq \{\}\} \\ &\leq \min\{z : P_z = \{\}\} \\ &= \min\{z : \exists u \geq 0, \lambda \geq 0 : -\lambda c^T + u^T A = 0, -\lambda z + u^T b < 0\} \end{aligned}$$

Wenn Lösung mit $\lambda = 0$ existiert, ist $P_z = \{\} \forall z$.

Wenn Lösung mit $\lambda > 0$ existiert:

$$\begin{aligned} &= \min\{z : \exists u \geq 0 : u^T A = c^T, u^T b < z\} \\ &= \min\{u^T b : u^T A = c^T, u \geq 0\} \end{aligned}$$

- 147 -

Lemma E.9 (Farkas-Lemma)

Sei $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. Dann gilt entweder

$$\exists x \in \mathbb{R}^n : Ax \leq b$$

oder

$$\exists u \in \mathbb{R}^m : u \geq 0, u^T A = 0, u^T b < 0.$$

- 146 -

Resultat: das Maximum von Problem (P) ist kleinergleich als das Minimum von Problem

$$\min u^T b \\ u^T A = c^T \\ u \geq 0 \quad (D)$$

($u \in \mathbb{R}^m$)

(D) heißt das zu (P) gehörende **duale Problem**.

- 148 -

Folgerung E.10

1. (P) ist unbeschränkt $\implies (D)$ ist unzulässig.
2. (D) ist unbeschränkt $\implies (P)$ ist unzulässig.

Schwache Dualität:

x zulässig für (P) , u zulässig für (D) . Dann gilt

$$c^T x = u^T A x \leq u^T b$$

- 149 -

Komplementarität

Im Optimum gilt:

1. Primale Zulässigkeit: $Ax \leq b$
2. Duale Zulässigkeit: $u^T A = c^T$, $u \geq 0$
3. ZF-Werte sind gleich: $c^T x = u^T b$

$$0 = u^T b - c^T x = u^T b - u^T A x = u^T (b - A x) = \sum_{i=1}^m u_i \cdot (b - A x)_i$$

Wegen $u_i \geq 0$ und $(b - A x)_i \geq 0$ gilt also:

1. Wenn $u_i \neq 0$, ist $(b - A x)_i = 0$.
2. Wenn $(b - A x)_i \neq 0$, ist $u_i = 0$.

Dies bezeichnet man mit **Komplementarität**.

- 151 -

Satz E.11 (Dualitätssatz)

Die beiden linearen Programme (P) und (D) haben optimale Lösungen mit dem gleichen Zielfunktionswert genau dann, wenn beide zulässige Lösungen haben.

(Beweis mit Farkas-Lemma)

Folgerungen:

1. (P) hat endliches Optimum $\iff (D)$ hat endliches Optimum, beide haben den gleichen Zielfunktionswert.
2. (P) ist unbeschränkt $\implies (D)$ ist unzulässig.
3. (D) ist unbeschränkt $\implies (P)$ ist unzulässig.
4. (P) ist unzulässig $\implies (D)$ ist unzulässig oder unbeschränkt.
5. (D) ist unzulässig $\implies (P)$ ist unzulässig oder unbeschränkt.

- 150 -

Im Simplex-Algorithmus, Schritt 1:

berechne Dualvariable:

$$u^T := c^T A_B^{-1}$$

Teste, ob

$$u \geq 0$$

\implies duale Zulässigkeit
+ primale Zulässigkeit
 \implies Optimalität

Wir sind also immer primal zulässig und im Lösungspunkt auch dual zulässig.

- 152 -

Variante: Starte mit dualer Zulässigkeit, iteriere, bis auch primale Zulässigkeit erfüllt ist.

→ **Dualer Simplex-Algorithmus**

Anwendung: Re-Optimierung mit Warm-Start, Vermeidung erneuter Phase I

d. h. nach erfolgter Optimierung: modifiziere Problem, optimiere erneut

- ▶ zusätzliche Variablen: setze zugehörige x -Werte auf 0, bleibt primal zulässig → weiter mit primalem Simplex
- ▶ zusätzliche Nebenbedingungen: setze zugehörige u -Werte auf 0, bleibt dual zulässig → weiter mit dualem Simplex (wichtig für Schnittebenenverfahren)

Lineare ganzzahlige Optimierung

Alle oder einige der Variablen müssen eine zusätzliche Ganzzahligkeitsbedingung erfüllen, typischerweise

$$x_i \in \{0; 1\} \text{ oder } x_i \in \mathbb{Z} \text{ oder } \dots$$

Modellierung von: Anzahlen, Entscheidungen, usw.

Durch die Ganzzahligkeitsbedingung erhalten wir Kombinatorische Optimierungsprobleme.

Simplex-Software

Kommerziell:

- ▶ CPLEX (ILOG)
- ▶ Xpress (Dash)
- ▶ ...

Akademisch:

- ▶ SoPlex (ZIB Berlin)
- ▶ Ipsolve
- ▶ ...

Traveling Salesman Problem (TSP)

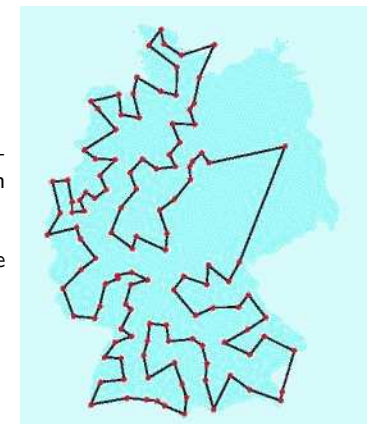
gegeben: Graph (V,E) mit Ecken V (Städten) und Kanten $E \subset V \times V$ (Straßen) und Kantengewichten c_e (Streckenlängen)

gesucht: die kürzeste Rundreise, d. h. die geschlossene Tour mit kürzester Länge durch alle Knoten

ordne jeder Kante $e \in E$ eine 0-1-Variable zu:

$$x_e = \begin{cases} 1 & \text{Kante } e \text{ gehört zur Tour} \\ 0 & \text{sonst} \end{cases}$$

(x : Inzidenzvektor)



Zielfunktion:

$$\min \sum_{e \in E} c_e x_e$$

Nebenbedingungen:

- zu jedem Knoten gehen zwei Kanten der Tour

$$\sum_{e \in \delta(v)} x_e = 2 \quad \forall v \in V$$

mit $\delta(v) := \{e \in E : \exists v \neq u : e = uv \vee e = vu\}$
 (Degree Equation)

- auf geschlossenen Strecken (Kreisen) mit Länge $< |V|$ dürfen nicht alle Kanten zur Tour gehören

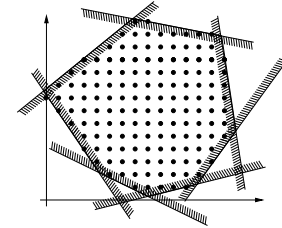
$$\sum_{e \in C} x_e \leq |C| - 1 \quad \forall \text{ Kreise } C \subset E, |C| < |V|$$

(Subcircle Elimination Constraint)

- 157 -

Allgemeines Mixed-Integer-LP (MILP)

$$\begin{aligned} \min & c^T x \\ Ax & \leq b \\ x_j & \text{ ganzzahlig, } j \in J \subset \{1, \dots, n\} \end{aligned}$$



mit der zulässigen Menge

$$S = \{x \in \mathbb{R}^n : Ax \leq b, x_j \text{ ganzzahlig, } j \in J \subset \{1, \dots, n\}\}$$

Bemerkung: Wenn man einen Punkt x^* hat, der (MILP) löst, gibt es keine Kriterien, mit denen man die Optimalität leicht nachweisen könnte. Schlimmstenfalls muß man alle zulässigen Punkte untersuchen.
 → Exponentielle Laufzeit.

(MILP) ist ein \mathcal{NP} -schweres Problem.

- 159 -

0-1-LP für das TSP

$$\begin{aligned} \min & \sum_{e \in E} c_e x_e \\ \sum_{e \in \delta(v)} & x_e = 2 \quad \forall v \in V \\ \sum_{e \in C} & x_e \leq |C| - 1 \quad \forall \text{ Kreise } C \subset E, |C| < |V| \\ x_e & \in \{0, 1\} \quad \forall e \in E \end{aligned}$$

- 158 -

Lösungsstrategien

- ▶ **Relaxierungen:** Vergrößere die zulässige Menge.
 z. B. lasse die Ganzzahligkeitsbedingungen weg → LP.
- ▶ **Teilprobleme:** Zerlege die zulässige Menge.
 z. B. links: $x_i \leq \lfloor x_i^S \rfloor$, rechts: $x_i \geq \lfloor x_i^S \rfloor + 1$
- ▶ **Heuristiken:** Finde schnell zulässige Punkte.
 z. B. Runden, Greedy-Heuristik

Diese Strategien müssen an das konkrete Problem angepasst sein!

- 160 -

- **Relaxierungen** liefern lokale untere Schranken:

$$\bar{S}_i \supseteq S_i \implies \min_{x \in \bar{S}_i} c^T x \leq \min_{x \in S_i} c^T x$$

- Lösungen von **Teilproblemen** liefern globale obere Schranken:

$$S = S_1 \cup S_2 \implies \min_{x \in S} c^T x \geq \min_{x \in S_i} c^T x$$

- **Zulässige Punkte** liefern globale obere Schranken:

$$\bar{x} \in S \implies c^T \bar{x} \geq \min_{x \in S} c^T x$$

Konvexe Hülle

Polyeder: $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ (Annahme: $\dim P = n$)

Zulässige Menge von (MILP): $S := P \cap \mathbb{Z}^{|J|}$

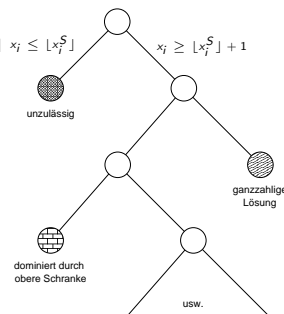
Die konvexe Hülle von S ist die kleinste konvexe Menge, die S enthält.

$$\text{conv} S = \left\{ x \in \mathbb{R}^n : x = \sum_{i=1}^q \lambda_i x^i, \sum_{i=1}^q \lambda_i = 1, \lambda_i \geq 0, \right.$$

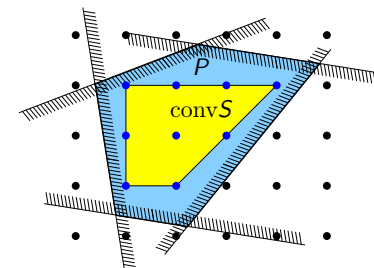
$\left. \{x^1, \dots, x^q\} \text{ ist eine beliebige endliche Menge von Punkten aus } S \right\}$

Das Branch-&-Bound-Verfahren

1. Initialisiere die Liste der aktiven Subprobleme mit dem gegebenen Problem (MILP), $x^* := \text{NULL}$.
2. Wenn die Liste leer ist, Stop: Problem (MILP) ist gelöst, Lösung: x^* .
3. Entferne ein Subproblem (SUB) aus der Liste, und arbeite es wie folgt ab.
4. Löse die LP-Relaxierung von (SUB).
5. (SUB) ist unzulässig, gehe zu (10).
6. Lösung x^S ist schlechter als bisher gefundener bester Punkt x^* , gehe zu (10).
7. x^S ist zulässig für (MILP), neuer bester Punkt: $x^* := x^S$, gehe zu (10).
8. Wende Heuristik an, um zulässigen Punkt x^H zu finden. Ist dieser besser als x^* : Setze $x^* := x^H$.
9. Teile (SUB) in zwei (oder mehr) neue Subprobleme auf, schreibe diese in die Liste.
10. (SUB) ist abgearbeitet, gehe zu (2).



Konvexe Hülle



Satz E.12

$\text{conv} S$ ist ein Polyeder mit Punkten von S als Ecken.

Die komplette Beschreibung von $\text{conv} S$ erfordert u. U. sehr viele ($\sim \exp n$) Ungleichungen. Deshalb arbeitet man besser mit Schnittebenen.

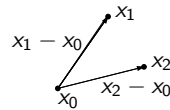
Gültige Ungleichungen und Facetten

Gültige Ungleichungen für $\text{conv}S$ sind Ungleichungen, für die gilt:

$$\alpha^T x \leq \beta \quad \forall x \in \text{conv}S$$

Eine **Seitenfläche** von $\text{conv}S$ der Dimension k wird beschrieben durch eine gültige Ungleichung für $\text{conv}S$, die von genau $k + 1$ affin unabhängigen Punkten aus $\text{conv}S$ mit Gleichheit erfüllt wird.

x_0, x_1, \dots, x_k affin unabhängig
 $\iff x_1 - x_0, \dots, x_k - x_0$ linear unabhängig:



Seitenflächen der Dimension 0 heißen Ecken
 Seitenflächen der Dimension 1 heißen Kanten
 ...
 Seitenflächen der Dimension $n - 1$ heißen **Facetten**.

Schnittebenenalgorithmus

für das Problem $\min\{c^T x : x \in P \cap \mathbb{Z}^{|J|}\}$

1. $t := 1, P^1 := P.$
2. Löse die LP-Relaxierung

$$c^T x^t := \min\{c^T x : x \in P^t\}.$$

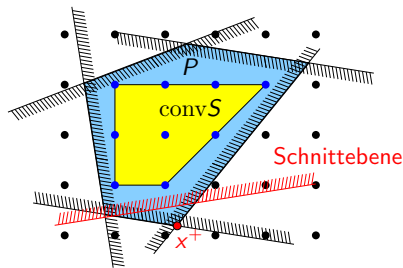
- Falls $x_j^t \in \mathbb{Z} \quad \forall j \in J$, STOP: (MILP) gelöst.
3. Generiere eine odere mehrere Schnittebenen

$$\alpha^j T x \leq \beta^j,$$

- die x^t von P^t abschneiden.
4. Definiere P^{t+1} durch Hinzufügen der Ungleichung(en) $\alpha^j T x \leq \beta^j$ zu P^t (und evtl. durch Entfernen einiger vorher hinzugefügter Ungleichungen).
 5. Setze $t := t + 1$, und gehe zu (2).

Schnittebenen

(eigentlich Schnitthyperebenen)



Abschneiden eines Punktes $x^+ \notin S$:

1. $\alpha^T x \leq \beta \quad \forall x \in S$
2. $\alpha^T x^+ > \beta$

Separationsproblem: Finde eine Ungleichung (aus einer Familie von möglichen Ungleichungen), die x^+ abschneidet.

Am besten: Facetten von $\text{conv}S$ als Schnittebenen.

Generieren von Schnittebenen

- ▶ **Problemspezifische Facetten**
 z. B. Facetten des TSP-Polytops
 Problem: Separation in polynomialer Laufzeit
- ▶ **Lift-&-Project-Cuts** für 0-1-Probleme
 Betrachte Facetten von

$$\begin{aligned} \min c^T x \\ Ax \leq b \\ x_j \in [0; 1] \quad \forall j \in J \\ x_i = 0 \vee x_i = 1 \end{aligned}$$

- Farkas-Lemma \rightarrow Charakterisierung der Facetten als Ecken eines Polyeders (Polare). Generierung von Facetten durch Lösung von LPs, die im wesentlichen doppelt so groß sind wie $Ax \leq b$.
- ▶ **Gomory-Cuts**
 Runden von Koeffizienten, so daß die Ungleichung für ganzzahlige Punkte erfüllt bleibt, aber für nichtganzzahlige Punkte verschärft wird.

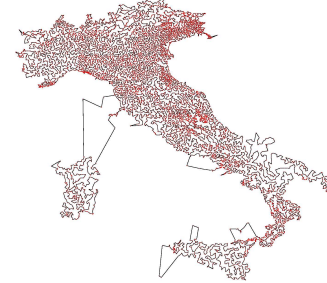
Branch & Cut

Kombiniere Branch & Bound und Schnittebenenalgorithmus.

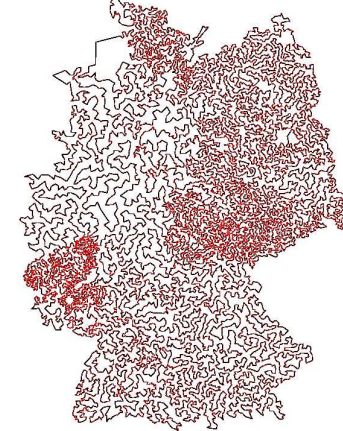
Generiere Schnittebenen in (einigen, vor allem frühen) Knoten, um (schnell) bessere Schranken zu erhalten.

TSP Beispiele

8246 Knoten:

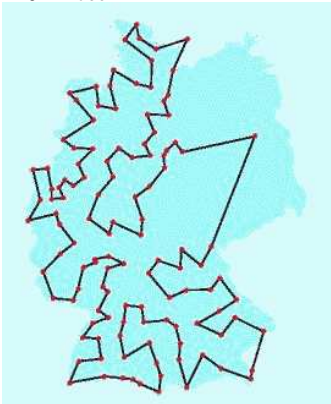


15112 Knoten:



TSP Beispiele

120 Knoten:



1000 Knoten:

