- B & B
- MIPs mit CPLEX
- Lagrange Relaxation

Lösen von ILPs

$$\min \; c^T x$$
$$\text{s.t.} \quad Ax = b$$
$$x \geq 0 \quad \text{und} \quad x \text{ ganzzahlig}$$

- enorme Modellierungskraft

z.B. Fixkosten: $c(x) = \begin{cases} ax + b & x > 0 \\ 0 & \text{sonst} \end{cases}$

Modellierung: $x \leq \delta \cdot U$, $c(x) = c \cdot x + \delta \cdot b$, $\delta \in \{0,1\}$

Falls x zulässig:

$$\boxed{x > 0 \Rightarrow \delta = 1}$$

z.B. disjunktive Bedingung: $x \leq a$ oder $y \geq b$

Modellierung: $x \leq a + (1-\delta)U$, $y \geq (1-\delta) b$, $\delta \in \{0,1\}$

$$\boxed{\delta = 1 \Rightarrow x \leq a, \quad \delta = 0 \Rightarrow y \geq b}$$

• auch NP-schwere Probleme modellierbar

z.B.

$$\min \; w^T x$$
$$x_u + x_v \geq 1 \quad \forall (u,v) \in E$$
$$x_v \in \{0,1\} \quad \forall v \in V$$

Satz: Frage der Existenz zul. Lösung für ILP ist NP-vollständig.

Wie ILPs lösen?

① **Spezialfall:** A vollst. unimodular, b ganzz.

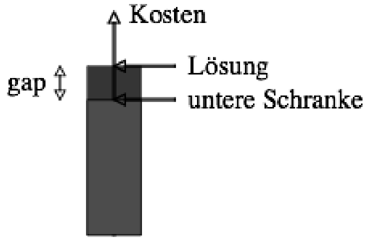
$\Rightarrow$ alle Ecken des von $Ax \leq b$ definierten Polyeders ganzz.

$\Rightarrow$ Simplex liefert ganzz. Optimallösung.

② allgemeiner Fall: **Branch & Bound**

Idee: geschickte Enumeration + untere Schranken

liefern Qualitätsgarantie:



"100%" } z.B. 120% => "Wert der Lösung ist innerhalb von 20% des Optimums". "Garantie"

<u>Annahme 1:</u> Können für gegebene Teilmenge des eigentlichen Zulässigkeitsbereichs den Optimalwert $l(L)$ einer Relaxation des eigentlichen Problems berechnen => untere Schranke an Optimalwert über der Teilmenge d. Zul.bereichs.

<u>Annahme 2:</u> Kennen eine zulässige Lösung x mit Wert u "incumbent" (amtierend)

"Knoten" v wird mit Teilmenge L_v des eigentlichen Zul.ber. und Optimalwert der Relaxation über L_v $l_v := l(L_v)$.

<u>Vorgehen</u>
- verwalte "open nodes" in D, zu Beginn ist nur der Knoten mit ganzem Zul.ber in D ("root node")
- $l := l(L)$ globale untere Schranke
- u Wert des incumbent
- main loop: bis $\frac{u-l}{l}$ klein genug do {
  - hole Knoten v aus D "search strategy"
  - berechne l_v; Falls $l_v \geq u$, continue w/ next iteration "pruning"
  - teile Zulässigkeitsbereich L_v in Teilbereiche L_i

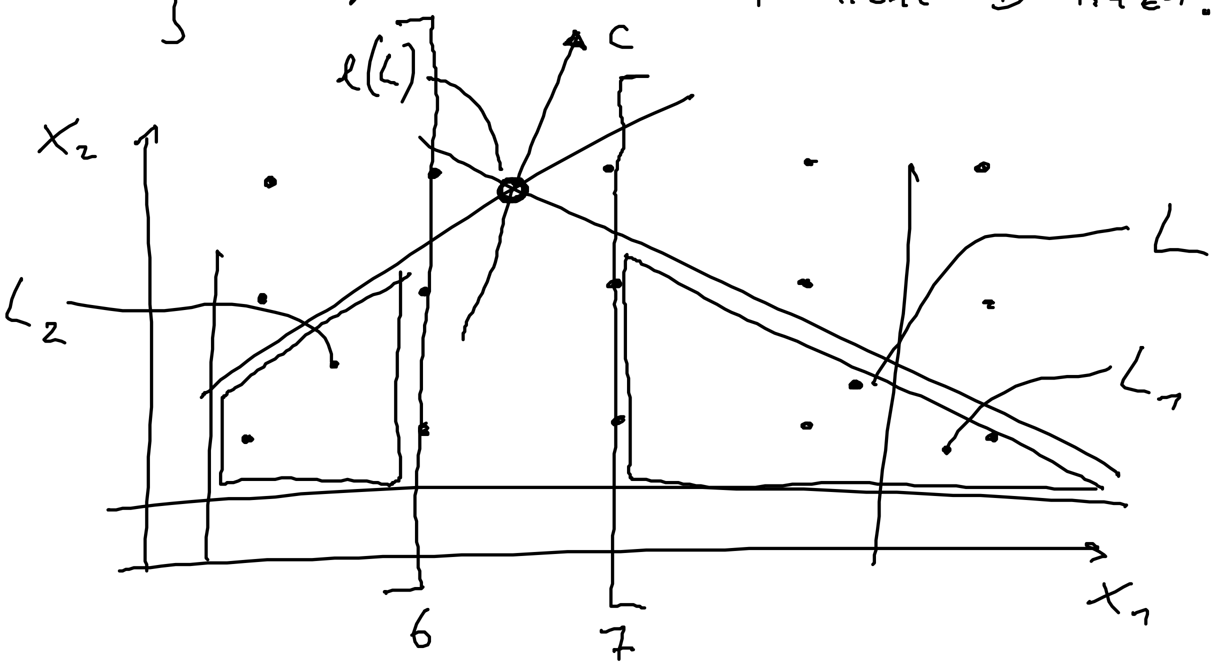
"branching"

- berechne Optimallösung x_i der Relaxation über L_i
  $\Rightarrow$ untere Schranken $l_i := l(L_i)$

"bounding"

- füge neue Knoten D hinzu für alle L_i mit $l_i < u$
- Falls eine Lösung x_i "zufällig" zulässig für eigentliches Problem, aktualisiere ggf. incumbent und u und füge diesen Knoten L_i nicht D hinzu.

}



Branching "auf x_1": $L_1 := \{x \in L \mid x_1 \geq 7\}$

$L_2 := \{x \in L \mid x_1 \leq 6\}$

"Definierende Zutaten" für B & B:

- search strategy: z.B. "best node first": wähle Knoten mit kleinster unterer Schranke l_v
- branching rule: z.B. ganzz. branching auf Variable, die am nächsten an einem ganzz. Wert ist.

• bounding strategy: z.B. LP-Relaxation oder bei kombinatorischen Problemen: kombinatorische Relaxation, z.B. "Cycle Cover Relaxation" für TSP:

<u>eine mögliche Formulierung:</u> $x_{ij} = 1 \iff$ Kante (i,j) in Tour

$$\min \sum_{(i,j)} c(i,j) \cdot x_{ij}$$

$$\sum_{j} x_{ij} = 1 \qquad \forall\, i$$

$$\sum_{i} x_{ij} = 1 \qquad \forall\, j$$

$$(3) \qquad \sum_{(i,j) \in S} x_{ij} \leq |S| - 1 \qquad \forall\, \emptyset \neq S \subset \{1, \ldots, n\}$$

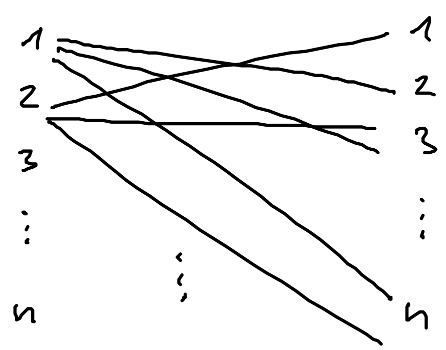
$$x_{ij} \in \{0, 1\}$$

(3) heißt Subtour-Elimination Constraint.

Kombinatorische Relaxation: Weglassen von (3)

$\Rightarrow$ "jeder Knoten genau eine ein- und eine ausgehende Kante"

$\Rightarrow$ baue bipartiten Graph



$\Rightarrow$ Optimallösung dieser Relaxation ist ein Min Weight Perfect Matching in diesem bipartiten Graph

$\Rightarrow$ polynomiell berechenbar, z.B. mit Min Cost Flow (ADM I)

<u>passende Branching-Regel:</u> z.B. wähle Kreis, am besten einen mit wenig Kanten, und zwinge die Variable einer seiner Kante auf 0 $\Rightarrow$ neues L_i.

MIPs mit ZIMPL / CPLEX

<u>ZIMPL:</u> keywords "binary" und "integer" für Variablen

<u>CPLEX Ausgabe:</u> 1 Zeile je Knoten

<u>Spalte</u>

0: * $\hat{=}$ ganzz. Lösung der LP-Relaxation

1: Nummer des Knotens

2: # open nodes

3: LP-objective oder "infeasible" oder "integral" oder "cutoff"

4: # nicht-ganzz. Var. in LP-Optimallösung

5: incumbent objective

6: Wert der besten unteren Schranke

7: Summe aller Simplex-Iterationen über alle Knoten

8: "Optimalitätsabstand" $\frac{u-l}{u}$ (!) $\neq \frac{u-l}{l}$

z.B. $u=2$, $l=1$

=> cplex: gap = 50%

S. auch User Guide → Discr. Opt. → Solving MIPS
→ Progress Reports