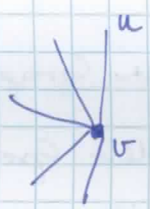
3.) MATCHING (LP-Relaxierung als exakter Algorithmus)

Eingabe: $G(V,E)$ mit Kantengewichten w_e

Ausgabe: Ein Matching $M \subseteq E$ mit maximalem Gesamtgewicht der Kanten in M
(die Kanten in M haben keine gemeinsame Endknoten)

LP-Relaxierung:

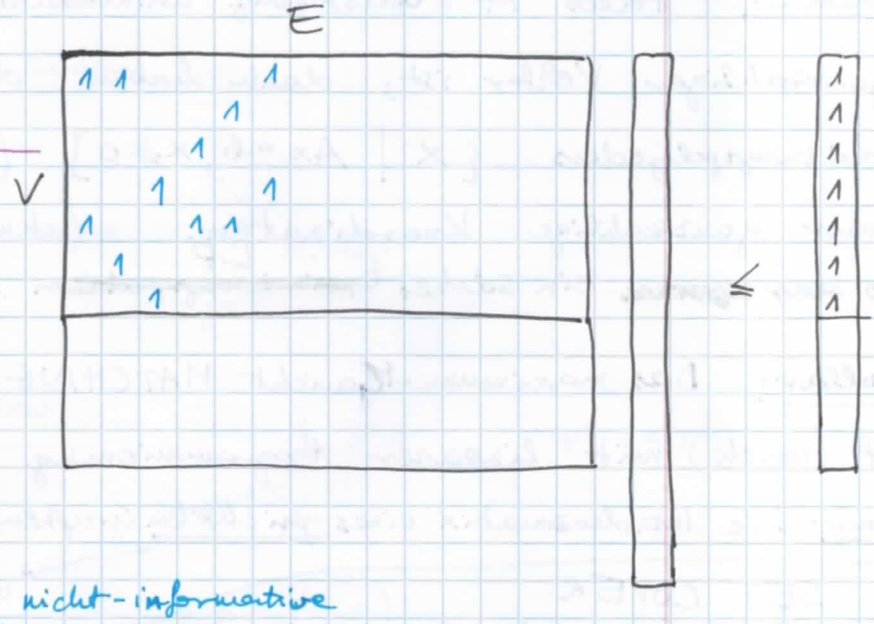
maximiere $\sum\limits_{e \in E}' x_e \cdot w_e$



so dass $\boxed{\sum\limits_{u:\{v,u\} \in E}' x_{\{v,u\}} \leq 1}$ $\quad \forall v \in V$

$0 \leq x_e \left(\leq 1\right)$ $\quad \forall e \in E$

die Matrix für diese Bedingungen ist die Knoten-Kanten Inzidenzmatrix



(hier gibt es auch nicht-informative fraktionale Lösungen, z.B. für einen vollständigen Graphen $x_e = \frac{1}{n-1}$ ist eine gültige aber nicht verwertbare Lösung, mit Wert $\frac{n(n-1)}{2} \cdot \frac{1}{n-1} = \frac{n}{2}$
aber:)

LP. 43. max - BIPARTITE MATCHING

Für bipartite Graphen hat die Inzidenzmatrix eine wichtige Eigenschaft:

Definition: Eine Matrix A heißt vollständig unimodular, falls die Determinante $\det B \in \{-1, 0, 1\}$ für jede quadratische Teilmatrix B von A gilt.
(Insbesondere sollen alle Einträge 1, 0 oder −1 sein)

Theorem 1. Die Inzidenzmatrix für einen ungerichteten Graphen ist genau dann vollständig unimodular, wenn der Graph bipartit ist. (ohne Beweis)

$\left\{ \begin{array}{l} (\Longrightarrow) \text{ ungerader Kreis:} \\ \text{hat Det} = 2 \end{array} \right.$

$$\begin{matrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & 1 & 1 & \\ & & & 1 & 1 \\ & & & & 1 \end{matrix}$$

Theorem 2. Falls A vollständig unimodular, und b ein ganzzahliger Vektor ist, dann haben die Ecken des Lösungspolyeders $\{x \mid Ax = b, x \geq 0\}$ (bzw. $\{x \mid Ax \geq b\}$) nur ganzzahlige Koordinaten. (ohne Beweis)

$\Longrightarrow$ ~~das Optimum~~ ein solches ~~Optimierungsproblem~~ LP besitzt eine ganzzahlige optimale Lösung

Korollar: Das maximum-Gewicht MATCHING Problem für bipartite Graphen ist (auch) mit linearer Programmierung exakt lösbar. (Warum?)

Theorem 3. Die Inzidenzmatrix eines gerichteten Graphen ist immer vollständig ganzzahl. unimodular

$$\begin{array}{c|c|} & (v,w) \\ v & 1 \\ w & -1 \end{array}$$

$\Longrightarrow$ Flussprobleme haben ganzzahlige Lösungen

4.) SET COVER

Eingabe: Grundmenge $\{1, 2, 3, \ldots, m\}$

n Teilmengen $S_j \subseteq \{1, 2, \ldots, m\}$ $j = 1, 2, \ldots, n$

mit Gewichten w_j

Ausgabe: Eine Überdeckung von $\{1, 2, \ldots, m\}$ mit einer leichtesten Auswahl ~~der~~ von Teilmengen

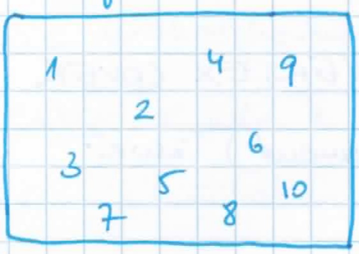
$C \subseteq \{1, 2, \ldots n\}$

((C ist die Indexmenge der ausgewählten Teilmengen S_j.

C ist eine Überdeckung der Grundmenge, wenn jedes Element $1 \le i \le m$ ist Element von mindestens einer ausgewählten Teilmenge $i \in S_j$ ($\exists j \in C$))

Beispiel:

Grundmenge



$S_1 = \{3, 4, 5, 8\}$

$S_2 = \{1\}$

$S_3 = \{1, 3, 5, 7, 9\}$

$S_4 = \{2, 4, 10\}$

$S_5 = \{2, 3, 4, 5, 6\}$

$S_6 = \{2, 6, 10\}$

$\{S_1, S_2, S_4, S_5\}$ und $\{S_1, S_3, S_6\}$ sind verschiedene Mengenüberdeckungen.

LP-Relaxierung:

minimiere $\displaystyle\sum_{j=1}^{n} x_j \cdot w_j$

Element i überdeckt wird $\xleftarrow{\text{so dass}}$ $\displaystyle\sum_{j:\, i \in S_j} x_j \ge 1 \quad \forall\, i = 1, 2, \dots, m$

(m Bedingungen)

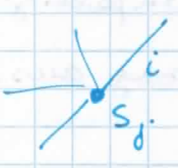
$0 \le x_j (\le 1)$

beabsichtigte Bedeutung: $x_j = \begin{cases} 1 & \text{falls } S_j \text{ in der Überdeckung } (j \in C) \\ 0 & \text{sonst.} \end{cases}$

SET COVER ist eine Verallgemeinerung des VERTEX COVER Problems

Kanten $\longleftrightarrow$ Elemente $1, 2, \dots, m$

Knoten $\longleftrightarrow$ Teilmengen S_j



Analog zur 2-Approximierung mit deterministem Runden für VERTEX COVER wird folgendes gezeigt:

<u>Thm:</u> Falls jedes Element i der Grundmenge in höchstens l Teilmengen S_j enthalten ist, dann ergibt ein deterministisches Runden der optimalen fractionalen Lösung eine l-approximative Lösung. (siehe Übung)

(beachte, dass im Spezialfall vom VERTEX COVER $l=2$ gilt, da jede Kante (=Element) zwei Endknoten (=Teilmengen) hat.)

<u>Für den allgemeinen Fall ohne solche l:</u>

<u>Ein Approximationsalgorithmus</u> (<u>randomisiertes Runden</u>)

(<u>die Idee</u>: wir lösen die LP-Relaxierung. Die optimale Fractionale Lösung $(x_1^* x_2^* \dots x_n^*)$ bestimmt die Wahrscheinlichkeiten x_j^*, dass die entsprechende Teilmenge S_j in die Mengenüberdeckung gewählt wird $(Prob(j \in C) = x_j^*)$. ABER: erhalten wir so tatsächlich eine Überdeckung aller Elemente der Grundmenge? NEIN)

⇒ mehrere (T) solche zufällige "Überdeckungen" werden vereinigt:

- löse die LP-Relaxierung;

  sei: $x^* = [x_1^* x_2^* \dots x_n^*]$ eine optimale fractionale Lösung

- FOR $t = 1$ to T do
  - setze $C_t = \emptyset$
  - FOR $j = 1$ to n do

    sei $j \in C_t$ mit Wahrscheinlichkeit x_j^*

- gib die Vereinigung $C = \bigcup_{t=1}^{T} C_t$ der T zufällig gewählten "Überdeckungen" C_t als Lösung aus.

<u>Theorem 1.</u> Mit der Wahl $T = \ln m + 1$ für die Anzahl der Runden erhalten wir eine vollständige Überdeckung der Grundmenge mit Wahrscheinlichkeit mindestens $\frac{1}{2}$.

(Warum reicht das? Siehe bei Monte Carlo Algorithmen!)

[<u>Beweis:</u>

Die Wahrscheinlichkeit dass ein konkretes Element i in <u>einer</u> Runde t <u>nicht</u> überdeckt wird (d.h. mit C_t) ist höchstens $\frac{1}{e}$.

<u>Warum?</u> Sei o.B.d.A i in den ersten k Mengen $S_1\ S_2 \ldots S_k$ enthalten. Wegen der LP-Bedingungen, für die fraktionale Lösung $[x_1^*\ x_2^* \ldots x_n^*]$ gilt

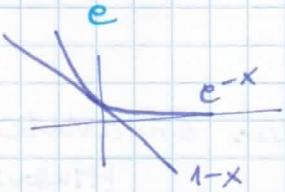
$$x_1^* + x_2^* + \cdots + x_k^* \geq 1. \quad (✿)$$

Die Wahrscheinlichkeit, dass i in <u>einer</u> Runde nicht überdeckt wird, ist

$$\text{Prob}\,(i \text{ wird nicht überdeckt}) = \underbrace{(1-x_1^*)}_{\nearrow\ \text{Prob.}\,(1 \notin C_t)} \cdot (1-x_2^*) \cdot \ldots \cdot (1-x_k^*)$$

$$\leq \left(1 - \frac{1}{k}\right)^k \leq \left(e^{-\frac{1}{k}}\right)^k = \frac{1}{e}$$

$$1 - x \leq e^{-x}$$



geometrisches Mittel: $\sqrt[k]{(1-x_1)\cdot(1-x_2)\cdot \ldots \cdot(1-x_k)}$

arithmetisches Mittel: $\dfrac{k - (x_1 + x_2 + \cdots + x_k)}{k} \leq \dfrac{k-1}{k} = 1 - \dfrac{1}{k}$ (✿)

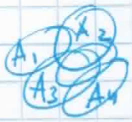
geom $\leq$ arithm.

$$(1-x_1)(1-x_2)\cdots(1-x_k) \leq \left(1 - \frac{1}{k}\right)^k$$

- Die Wahrscheinlichkeit, dass i nach T Runden immer noch nicht überdeckt wird, ist höchstens $\left(\frac{1}{e}\right)^T$

- Sei A_i das Ereignis, dass Element i nach T Runden nicht überdeckt wird. Dann ist $A_1 \cup A_2 \cup \ldots \cup A_m$ das Ereignis, dass die Grundmenge nicht vollständig überdeckt wird. Es gilt: $Prob(A_i) \le \left(\frac{1}{e}\right)^T$

$$Prob(A_1 \cup A_2 \cup \ldots \cup A_m) \le Prob(A_1) + Prob(A_2) \ldots + Prob(A_m)$$



$$\le m \cdot \left(\frac{1}{e}\right)^T = \frac{m}{e^T} = \frac{m}{e^{\ln m + 1}} = \frac{m}{m \cdot e} \le \frac{1}{2}$$

<u>Theorem 2.</u> Das erwartete Gewicht der Ausgabe $C = \bigcup_t C_t$ ist höchstens $(\ln m + 1) \cdot OPT_{SET\ COVER}$

<u>Beweis:</u>

Das erwartete Gewicht <u>eines</u> C_t ist $\le OPT_{SET\ COVER}$ und von allen C_t ist es $\le T \cdot OPT_{SET\ COVER} =$

$$= (\ln m + 1) \cdot OPT_{SET\ COVER}.$$

Warum ist das erwartete Gewicht (Erwartung über die zufällige ~~Wahl~~ Hinzunahme von j mit Prob x_j^* für alle $j = 1 \ldots n$) von $C_t \le OPT$?

Sei die Zufallsvariable W_j das Gewicht von S_j in der "Überdeckung" C_t. Dann ist $W_j = \begin{cases} w_j & \text{mit Prob } x_j^* \\ 0 & \text{mit Prob } 1 - x_j^* \end{cases}$

$$\mathbb{E}[W_j] = w_j \cdot x_j^* + 0 \cdot (1 - x_j^*)$$

das erwartete Gewicht von C_t ist $\mathbb{E}[W_1 + W_2 + W_3 + \ldots + W_n] =$

$$= \mathbb{E}[W_1] + \mathbb{E}[W_2] + \ldots + \mathbb{E}[W_n] = \sum_j w_j \cdot x_j^* = OPT_{FRAC} \le OPT_{SET\,C}$$

Dualität in der Linearen Programmierung

Illustration/Beispiel: (siehe Vazirani 12.1)

minimiere $7x_1 + x_2 + 5x_3$

so dass

$$x_1 - x_2 + 3x_3 \geq 10$$

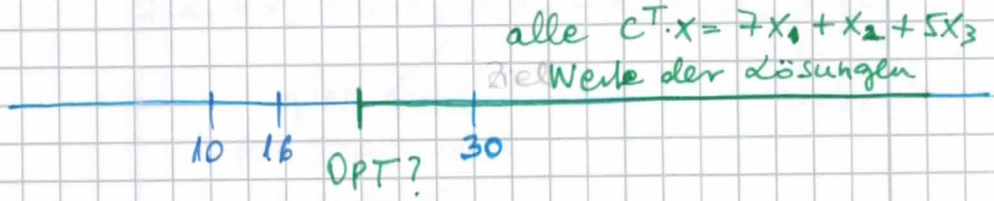
(P)

$$5x_1 + 2x_2 - x_3 \geq 6$$

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

(ein LP in kanonischer Form)

Statt dieses LP zu lösen, werden wir versuchen „schneller" (obere und) untere Schranken für den optimalen Wert OPT finden



- jede Lösung ergibt eine <u>obere Schranke</u> für OPT z.B. $x = (2, 1, 3)$

→ erfüllt die Nebenbedingungen:

$$x_1, x_2, x_3 \geq 0 \quad ✓$$

$$2 - 1 + 3 \cdot 3 \geq 10 \quad ✓$$

$$5 \cdot 2 + 2 \cdot 1 - 3 = 9 \geq 6 \quad ✓$$

→ und hat Zielwert

$$7x_1 + x_2 + 5x_3 =$$
$$= 7 \cdot 2 + 1 + 5 \cdot 3 = 30$$

Wie findet man gute untere Schranken für OPT?

1.) Eine einfache untere Schranke:

$$7 \cdot x_1 + x_2 + 5 \cdot x_3 \geq x_1 - x_2 + 3x_3 \geq 10$$

gilt Term für Term, weil $x_1, x_2, x_3 \geq 0$

Nebenbedingung

10 ist untere Schranke für den Zielwert jeder Lösung, also auch für OPT

2.) eine bessere untere Schranke:

weil $x_1, x_2, x_3 \geq 0$

$$7x_1 + x_2 + 5x_3 \geq \underbrace{(x_1 - x_2 + 3x_3) + (5x_1 + 2x_2 - x_3)}_{6x_1 + x_2 + 2x_3} \geq$$

$$\geq 10 + 6 = \underline{16}$$

laut Nebenbedingungen

3.) eine noch bessere untere Schranke für OPT

Term für Term weil $x_1, x_2, x_3 \geq 0$

$$7x_1 + x_2 + 5x_3 \geq 2 \cdot (x_1 - x_2 + 3x_3) + (5x_1 + 2x_2 - x_3)$$

$$\geq 2 \cdot 10 + 6 = \underline{26}$$

laut Nebenbedingungen

4.) <u>Allgemeine Idee</u>: finden wir le

$y_1 \geq 0$ und $y_2 \geq 0$

so dass

$$\underbrace{7x_1 + x_2 + 5x_3}_{c^T \cdot x} \geq \underbrace{y_1 (x_1 - x_2 + 3x_3) + y_2 (5x_1 + 2x_2 - x_3)}_{y^T \cdot A \cdot x}$$

$$\geq \underbrace{y_1 \cdot 10 + y_2 \cdot 6}_{y^T \cdot b} \longrightarrow \text{untere Schranke für OPT}$$

gilt laut Nebenbedingungen, weil $y_1, y_2 \geq 0$

Wir brauchen also:

B $y_1, y_2 \geq 0$

Und, für die Koeffizienten von x_1, x_2, x_3

$$7 \geq y_1 + 5y_2$$

$$1 \geq -y_1 + 2y_2 \qquad (D)$$

$$5 \geq 3y_1 + (-y_2)$$

und $10 \cdot y_1 + 6y_2$ so groß wie möglich.

(maximiere $10 \cdot y_1 + 6y_2$ unter den obigen Bedingungen)

das ist ein anderes LP, ein Maximierungsprogramm

(D) ist das sog. <u>duale LP</u> von (P)

(und umgekehrt: (P) ist auch das duale LP von (D))