24/a

(Shmoys-Williamson Chapter 7.3)

Beispiel 2. Primal-dual Algorithmus für das shortest-s-t-Path Problem

(Schulbeispiel für ein exakt (effizient) approximierbares Problem)

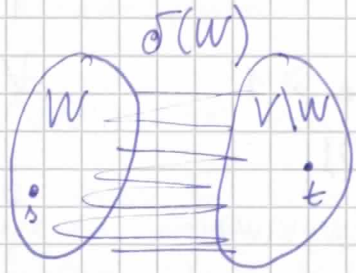
Eingabe: ungerichteter Graph $G(V, E)$
- mit einer Länge $l_e \geq 0$ für jede Kante $e \in E$
- zwei ausgezeichnete Knoten $s, t \in V$

Ausgabe: ein kürzester s-t Pfad

Definitionen:

→ eine Knotenmenge $W \subseteq V$ mit $s \in W$ $t \notin W$ definiert einen s-t Schnitt, d.h. eine Partition $(W, V \setminus W)$ so dass $s \in W$ und $t \in V \setminus W$.
Der Einfachheit halber nennen wir eine solche W auch einen s-t-Schnitt. (nicht nur die Partition)

→ Bezeichne $\delta(W)$ die Menge aller kreuzenden Kanten im Schnitt $(W, V \setminus W)$



→ Sei $\mathcal{S} = \{W \subseteq V \mid s \in W, t \notin W\}$ die Menge aller s-t-Schnitte

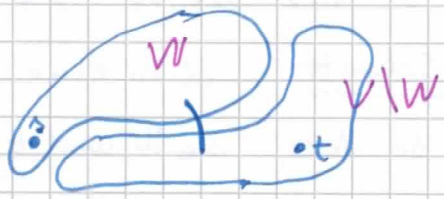
24/6

Eine IP-Formulierung des Shortest-s-t-Path Problems

(für konkrete Instanz (G, l, s, t))

- wir nehmen eine Variable x_e für jede Kante $e \in E$
  Wir möchten, dass die Kanten mit $x_e = 1$
  in der (optimalen) Lösung einen s-t-Pfad ergeben,
  oder zumindest einen s-t-Pfad <u>enthalten</u>.

- Diesmal fordern wir nicht die Flusserhaltung-Bedingungen!
  Wir fordern, dass jeder s-t-Schnitt von mindestens einer Kante e mit $x_e = 1$ überquert wird



Somit wird jede Lösung (Kanten mit $x_e = 1$) einen s-t-Pfad <u>enthalten</u> (siehe Behauptung 1).

<u>IP-Formulierung:</u>

minimiere $\sum_{e \in E} l_e \cdot x_e$

so dass $\sum_{\{e \mid e \in \delta(W)\}} x_e \geq 1 \qquad \forall\, W \in \mathcal{S}$

$\{e \mid e \in \delta(W)\}$: e kreuzt $(W, V \setminus W)$

$W \in \mathcal{S}$: für jeden s-t-Schnitt

$x_e \in \{0, 1\} \qquad \forall\, e \in E$

<u>Behauptung 1:</u> Die Kanten mit $x_e=1$ enthalten genau dann einen s-t-Pfad, wenn x die Bedingungen im IP für jeden s-t-Schnitt W erfüllt.

<u>Beweis:</u>

1. Die Bedingungen, dass jeder s-t-Schnitt von einer Kante überquert wird, sind <u>notwendig</u>:

   Trivial. x enthält einen s-t Pfad, und dieser muss jeden s-t-Schnitt überqueren (mindestens einmal)

2. Die Bedingungen sind auch <u>hinreichend</u>:

   D.h. eine solche Lösung x enthält mindestens einen s-t Pfad. (d.h. die Kantenmenge mit $x_e=1$) enthält einen Pfad

   Wir zeigen 2.: Sei x eine Lösung des IP, und seien

   $$E' = \{e \in E \mid x_e = 1\}$$

   $$W' = \{v \in V \mid v \text{ erreichbar aus } s \text{ über Kanten in } E'\}$$

   Angenommen, dass E' keinen s-t Pfad enthält (durch Widerspruch), dann $t \notin W'$

   $\Rightarrow$ $(W', V\setminus W')$ ist ein s-t-Schnitt, ohne kreuzende Kante aus E', also ohne kreuzende Kante mit $x_e=1$
   (weil die Knoten ausserhalb W' nicht erreichbar sind)

   $\Rightarrow$ die lineare Nebenbedingung $\sum_{\{e \mid e\in\delta(W')\}} x_e \geq 1$ für diesen W' s-t-Schnitt wird verletzt, also x war doch keine Lösung. ↯ □

24/d

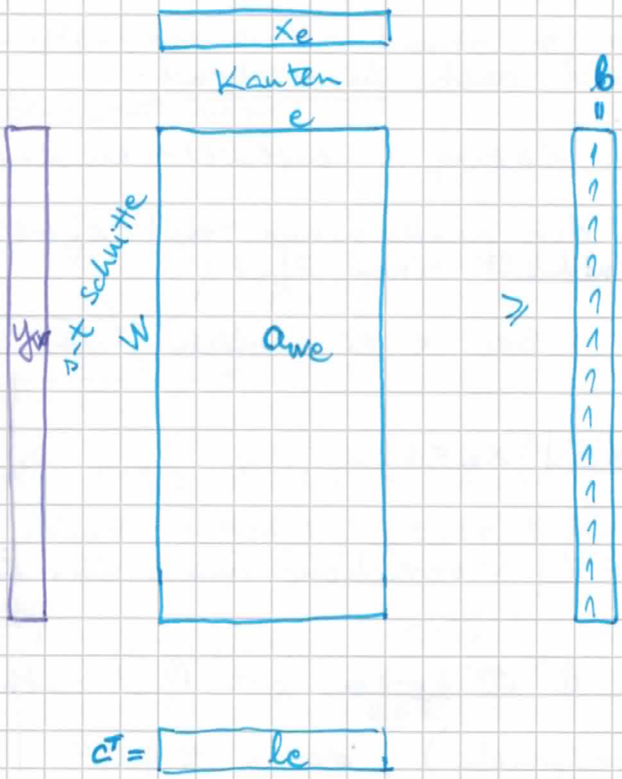
→ in der LP-Relaxierung des IP wird $x_e \in \{0,1\}$ durch $0 \leq x_e$ ersetzt für jede $e \in E$.

($x_e \leq 1$ wird in einer optimalen Lösung automatisch erfüllt.)

→ Beachte: Wir haben exponentiell viele Nebenbedingungen! (Warum? Wie viele s-t-Schnitte gibt es?)
ABER: wir werden das LP gar nicht lösen müssen!

<u>Wir formulieren das duale Programm der LP-Relaxierung</u>

(P)



a_{We}: ob e kreuzt Schnitt W

→ Im dualen Programm (D) haben wir eine y_W Variable für jeden s-t-Schnitt W

→ und eine Nebenbedingung für jede Kante $e \in E$

maximiere $\sum_{W \in S} y_W$ $(y^T \cdot b)$

(D) so dass

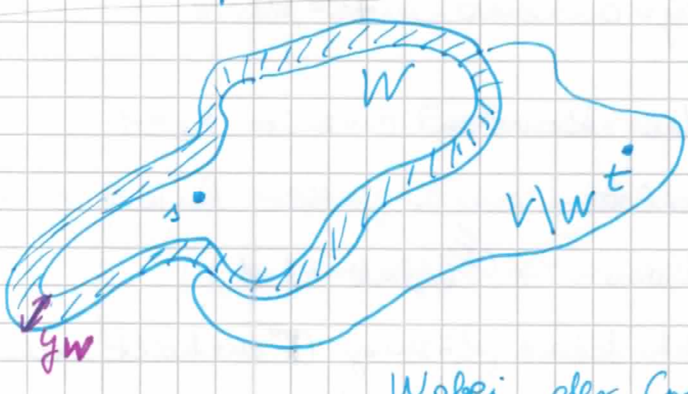
$$\sum_{\{W \mid e \in \delta(W)\}} y_W \leq l_e \qquad \forall e \in E$$

alle s-t-Schnitte die e kreuzt

$$y_W \geq 0$$

(Die Summe der y_W für alle s-t-Schnitte W die e kreuzt ist höchstens l_e)

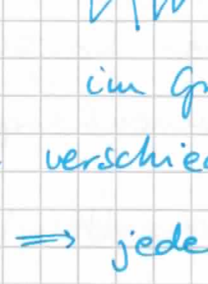
→ Interpretation der y_W: mindestens so viel kostet es noch im Zielwert (Pfadlänge) den Schnitt $(W, V \setminus W)$ zu überqueren.



(y_W: die Breite eines "Grabens" um W herum, wobei die Kanten "Brücken" entsprechen.

Wobei der Graph allgemein nicht planar ist; W nicht "links" und $V \setminus W$ nicht "rechts" ist sondern überall im Graph sein kann

Die Graben verschiedener s-t-Schnitte überlappen sich nicht ⇒ jede Kante e kann Graben mit Gesamtbreite $\leq l_e$ überqueren.



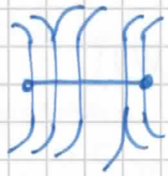
Ein s-t Pfad muss jeden s-t-Schnitt (jeden Graben) überqueren, und hat Gesamtlänge $\geq \sum_{W \in S} y_W$)

24/f

→ die PKS-Bedingungen: (primale komplementäre-slackness Bedingungen)

entweder $x_e = 0$ (Kante e nicht gewählt in der Lösung)

oder $\sum_{W \mid e \in \delta(W)}' y_W = l_e$



die Gesamtbreite der Graben die e kreuzt, ist genau die Länge der Kante e

<u>Primal-dualer Algorithmus</u>

Eingabe: $G(V,E)$, $l: E \to \mathbb{R}_+$, $s,t \in V$

– starte mit $y=0$ $x=0$ $F=\emptyset$

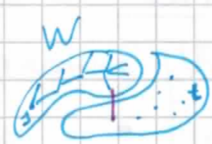
(die Kanten e mit $x_e=1$ bilden nach jeder Runde einen Baum $F \subseteq E$ (statt Pfad))

– WHILE x noch keine Lösung (F enthält noch keinen s-t Pfad)

~~– sei $F = \{e \in E \mid x_e = 1\}$~~

– sei W die Menge aller Knoten erreichbar über Kanten in F [redacted] aus s (Zusammenhangskomponente von s)

(wir müssen $(W, V \setminus W)$ mit einer weiteren Kante überqueren aber mit welcher?



Wir wählen die neue Kante nicht beliebig, sondern die y Lösung hilft zu wählen.)

- erhöhe y_W bis für irgendeine Kante $e \in \delta(W)$ gilt:

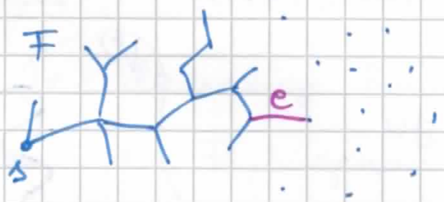
$$\sum_{W' \mid e \in \delta(W')} y_{W'} = l_e$$

- setze $\begin{cases} x_e = 1 & \text{für diese Kante} \\ F = F \cup \{e\} \end{cases}$

  (e kommt zu den Baumkanten F)

(die PKS-Bedingungen bleiben erfüllt)

Wichtige Bemerkung: F bleibt ein Baum:



weil e den bisherigen Baum F mit einem nicht-Baum-Knoten verbindet

— am Ende enthält F einen s-t Pfad, nenne ihn P, da F ein Baum ist, ist dieser s-t Pfad $P \subseteq F$ eindeutig (der einzige s-t Pfad in F), und ihn gibt der Algorithmus aus

---

Theorem: Dieser Primal-dual Algorithmus gibt einen kürzesten s-t-Pfad aus.

Beweis: Wir nutzen wieder, dass die PKS-Bedingungen gelten, also, dass für jede Kante $e \in F$ (d.h. mit $x_e = 1$) gilt

$$l_e = \sum_{\{W' \mid e \in \delta(W')\}} y_{W'}$$

24/11

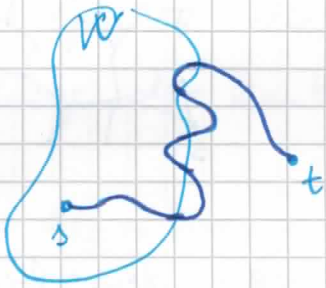
PCF war die Kantenmenge des Pfades ausgegeben vom Algorithmus. Sei $x' \leq x$ der 0-1 Vektor, der P entspricht (statt x, der F entspricht)

$$\text{Länge des s-t Pfades } P = \sum_{e \in P} l_e = \sum_{e \in P} \underbrace{\sum_{\{W \mid e \in \delta(W)\}} y_W}_{\text{Summe der } y_W \text{ Spaltenweise}} = \underbrace{\sum_{W \in S}}_{\text{jeder s-t-Schnitt}} \underbrace{\sum_{\substack{e \in P \\ e \text{ kreuzt } W}} y_W}_{\text{Summe der } y_W \text{ Zeilenweise}} =$$

$$\sum_{e \in P} l_e = \sum_e x'_e \cdot l_e = c^T \cdot x'$$

wie oft kreuzt P den $(W, V \setminus W)$

$$= \sum_{W \in S} \underbrace{|\delta(W) \cap P|}_{= 1 \text{ falls } y_W > 0} \cdot y_W$$



Brauchen: P kreuzt W genau 1 mal, falls $y_W > 0$ ◆

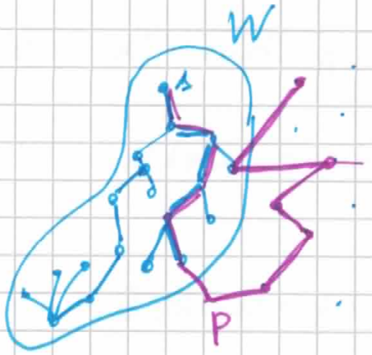
$$= \sum_{W \in S} y_W = \sum_{W \in S} y_W \cdot 1 = y^T \cdot b$$

Wert der gebildeten dualen Lösung y

$\Longrightarrow$ der karakteristische Vektor x' des Pfades P ist eine optimale Primale Lösung, weil eine duale Lösung y den selben Wert hat.

◆ sagt, dass in diesem Algorithmus sogar die DKS (Duale Komplementäre Slackness Bedingungen) gelten für die gefundenen Lösungen x und y (wobei x nur für den Pfad P der kar. Vektor ist)

Warum gilt ◆ ?



Pfadkante

Kante des damaligen F als y_W erhöht wurde

- als der Algorithmus y_W hochgesetzt hat ($y_W > 0$)

enthielt ~~bestand~~ der damalige Baum, gespannt durch die Kanten von F, genau die Knoten W

- wenn der Pfad P den Schnitt W ≥ 2 mal kreuzen würde, dann würde F am Ende einen Kreis enthalten ↯.

$$\left(\text{weil} \quad \begin{matrix} F_{\text{damals}} \subset F_{\text{letzter}} \\ P \subset F_{\text{letzter}} \end{matrix} \right)$$

Allgemein:

→ da x' und y beide optimal waren (da $c^T \cdot x = y^T \cdot b$)

gelten auch die DKS (Duale Komplementäre Slackness)

– Bedingungen, weil (mit allgemeiner Notation)

$$c^T \cdot x' - y^T \cdot b = \sum_{j=1}^{n}{}' \underbrace{(c_j - y^T \cdot a^j) \cdot x_j'}_{=0 \text{ (PKS)}} + \sum_{i=1}^{m}{}' \underbrace{y_i (a_i^T \cdot x' - b_i)}_{=0 \text{ (DKS)}} = 0$$

in unserem Fall $a_w^T \cdot x'$ ist die Anzahl der Kanten von P die W kreuzen und $b_w = 1$

DKS: $y_w = 0$, oder genau eine Kante von P kreuzt W ◆

→ Anders geschrieben: die Analyse des P-D Algorithmus entspricht:

$$c^T \cdot x' \underset{\text{(PKS)}}{\overset{\Uparrow}{=}} (y^T \cdot A) \cdot x' = y^T \cdot (A x') \underset{\text{(DKS)}}{\overset{\Uparrow}{=}} y^T \cdot b$$

→ Jetzt zurück zur Analyse vom P-D-Algorithmus für SET COVER:

DKS für SET COVER:

in der LP-Formulierung (P) und (D) für SET COVER, $a_i^T \cdot x$ ist die Anzahl der Teilmengen in der Überdeckung C, die Element i überdecken, und $b_i = 1$

DKS: $y_i = 0$, oder genau eine Teilmenge aus der Lösung C enthält Element i

→ die Analyse des P-D Algorithmus für SET COVER entspricht

$$c^T \cdot x \underset{(PKS)}{=} (y^T \cdot A) \cdot x = y^T (Ax) \leq \ell \cdot y^T \cdot b$$

$a_i^T \cdot x \leq \ell \cdot b_i = \ell$
weil jedes Element in höchstens ℓ Teilmengen enthalten

Statt der DKS: $y_i = 0$ oder $a_i^T \cdot x = b_i$

hatten wir dort

relaxierte DKS-Bedingungen: $y_i = 0$ oder $a_i^T \cdot x \leq \beta \cdot b_i$

(mit $\beta = \ell$). Diese ergeben einen β-approximativen Primal-Dual Algorithmus.

Wir verallgemeinern diese Beobachtungen:

Theorem:

Wenn für die Lösungen x von (P) und y von (D)

gelten die primalen Komplementäre-Slackness (PKS) Bedingungen:

$$x_j > 0 \implies y^T \cdot a^j = c_j$$

und die relaxierten dualen Komplementäre Slackness (DKS) Bedingungen

$$y_i > 0 \implies a_i^T \cdot x \leq \beta \cdot b_i$$

für irgendein $\beta > 0$, dann ist die Lösung x (und auch y) β-approximativ.

Warum? mit analogem Beweis wie beim Komplementären Slackness oben

$$\boxed{c^T \cdot x - y^T \cdot \beta \cdot b} = (c^T - y^T \cdot A) \cdot x + y^T \cdot (A \cdot x - \beta \cdot b) =$$

$$= \sum_{j=1}^{n} \underbrace{(c_j - y^T \cdot a^j) \cdot x_j}_{=0} + \sum_{i=1}^{m} \underbrace{y_i \cdot (a_i^T \cdot x - \beta b_i)}_{\leq 0} \leq 0$$

$$\implies \quad c^T \cdot x \leq \beta \cdot y^T \cdot b \leq \beta \cdot OPT_{frac.} \leq \beta \cdot OPT_{ganz}$$

(wobei OPT_{ganz} das primale ganzzahlige Optimum bedeutet)