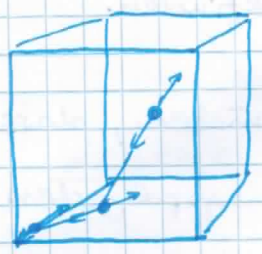


Dies ist keine starke Einschränkung, da z.B. die Standardform, (die wir bald kennenlernen) sie stets erfüllt, und jedes LP ein äquivalentes LP in Standardform hat.

Theorem: Wenn für ein LP mit n Variablen, und n linear unabhängigen Nebenbedingungen ein endliches Optimum existiert, dann gibt es eine optimale Ecke des Lösungspolyeders. (Ein Alg. braucht also nur Ecken der Lösungsmenge zu besuchen, und dies sogar nur "in Richtung" $-c$ tun!)

[Warum? (beide Definitionen einer Ecke werden benutzt)
sei P das Lösungspolyeder



- sei $x \in P$ eine optimale Lösung ($c^T \cdot x$ minimal)
- falls x eine Ecke \rightarrow fertig
- sonst gibt es ein y sodass $x+y \in P$ und $x-y \in P$

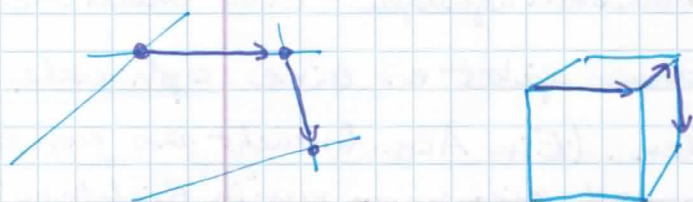
- es gilt $c^T \cdot (x+y) = c^T \cdot x = c^T \cdot (x-y)$
 oder $c^T \cdot (x+y) > c^T \cdot x > c^T \cdot (x-y)$
 oder $c^T \cdot (x+y) < c^T \cdot x < c^T \cdot (x-y)$

(je nachdem ob wir von der Hyperebene $c^T \cdot x = 0$ uns entfernen oder näher kommen)

Die beiden letzten Fälle sind ausgeschlossen, weil x schon optimal war, also ~~erfüllt~~ $x+y$ und $x-y$ sind auch optimal

- P enthält keine Gerade (nach Annahme) also entweder in Richtung y oder in Richtung $-y$ erreichen wir eine (neue) Bedingung die exakt erfüllt wird. Entweder sind wir in einer Ecke, oder wir machen weiter mit neuem y - wenn n Bedingungen exakt erfüllt werden, sind wir in einer Ecke

- Da wir das Optimum (Minimum) in einer Ecke suchen, werden wir mit dem Simplex-Algorithmus von Ecke zur Ecke wandern, und zwar von einer Ecke in eine benachbarte Ecke, solange es eine benachbarte Ecke mit kleinerem Zielwert gibt. Wir brauchen also das Konzept einer benachbarten (adjazenten) Ecke



intuitiv: die Ecken x^* und x^{**} sind benachbart wenn sie sich an einer gemeinsamen Kante befinden.

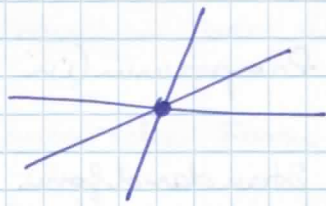
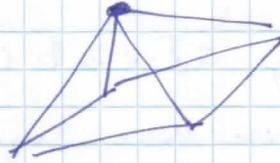
(Wie können wir algebraisch ausdrücken, dass eine Lösung x an einer Kante des Lösungspolyeders ist?)

Wie viele Nebenbedingungen werden an einer Kante als Gleichung erfüllt? \rightarrow $n-1$ Nebenbedingungen.

Definition: Zwei Ecken $x^*, x^{**} \in P$ sind benachbart oder adjazent, wenn es $n-1$ linear unabhängige Nebenbedingungen $a_i \cdot x \geq b_i$ gibt, die x^* und x^{**} beide exakt erfüllen.

(die beiden Ecken unterscheiden sich in nur einer exakt erfüllten Bedingung)

(Entartete Ecken: Die Ecke $x^* \in P$ ist entartet, wenn in x^* mehr als n Nebenbedingungen exakt erfüllt werden)

Bsp. \mathbb{R}^2  ≥ 3 Geraden \mathbb{R}^3  ≥ 4 Ebenen

entartete Ecken bereiten Schwierigkeiten für den Simplex-Algorithmus, der die entartete Ecke als viele benachbarten Ecken auffasst, und deshalb im Kreis laufen kann (siehe später)

iv.) Lineare Programme in Standardform

Definition: lineares Programm in Standardform:

minimiere $c^T \cdot x$, so dass $A \cdot x = b$, $x \geq 0$

$$\left(x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad c \in \mathbb{Q}^n \quad A \in \mathbb{Q}^{m \times n} \quad b \in \mathbb{Q}^m \right)$$

(Bemerkung: in der kanonischen Form (wenn die Vorzeichenbedingungen nicht separat sind) braucht man $m \geq n$ um n linear unabhängige Bedingungen zu haben. In der Standardform, wie wir sehen werden, $m \leq n$ (aber die Vorzeichenbedingungen gehören nicht zur Matrix A))

Behauptung: Jedes lineare Programm (in kanonischer Form) hat ein äquivalentes LP in Standardform

(\Rightarrow die Standardform ist auch allgemeiner)

Beweis: 1. Für jede Ungleichung $a_i^T \cdot x \geq b$ in der kanonischen Form, führen wir eine sog.

Slackvariable s_i ein, und ersetzen $a_i^T \cdot x \geq b$

$$\text{durch } a_i^T \cdot x - s_i = b$$

$$s_i \geq 0$$

2. die Standardform fordert $x_j \geq 0 \quad \forall j$

- falls im kanonischen LP $x_j \geq 0$ gefordert wird
ok. \checkmark

- falls im kanonischen LP $x_j \leq 0$ gefordert wird
ersetzen wir $-x_j$ durch eine

neue Variable x_j^-

$$x_j^- = -x_j$$

$$x_j \leq 0 \Leftrightarrow x_j^- \geq 0$$

- falls im kanonischen LP x_j beliebig ist,
ersetzen wir x_j durch die

Differenz zweier positiven Variablen (nichtnegativen)

$$x_j = x_j^+ - x_j^-$$

und die Bedingungen

$$x_j^+ \geq 0$$

$$x_j^- \geq 0$$

~~(...)~~

(Wann brauchen wir wieder eine neue Form?)

→ die kanonische Form ist anschaulich und intuitiv;
 die Standardform ist besser geeignet als Eingabe
 (und die Beschreibung) von Algorithmen wie der Simplex-
 Algorithmus, oder das Interior-Point Verfahren)

Beispiel 1.

minimiere $2x_1 + 4x_2$
 so dass $x_1 + x_2 \geq 3$
 $3x_1 + 2x_2 = 14$
 $x_1 \geq 0$

Konstruieren wir die äquivalente Standardform:

- nach Einführung der Slackvariable in die
 erste Bedingung, haben wir

minimiere $2x_1 + 4x_2$
 so dass $x_1 + x_2 - s_1 = 3$
 $3x_1 + 2x_2 = 14$
 $x_1 \geq 0$
 $s_1 \geq 0$

- um Nichtnegativität fordern zu können
 ersetzen wir x_2 durch $x_2^+ - x_2^-$ und erhalten

minimiere $2x_1 + 4x_2^+ - 4x_2^-$
 so dass $x_1 + x_2^+ - x_2^- - s_1 = 3$
 $3x_1 + 2x_2^+ - 2x_2^- = 14$
 $x_1 \geq 0$
 $s_1 \geq 0$
 $x_2^+ \geq 0$
 $x_2^- \geq 0$

LP23.

Beispiel 2.

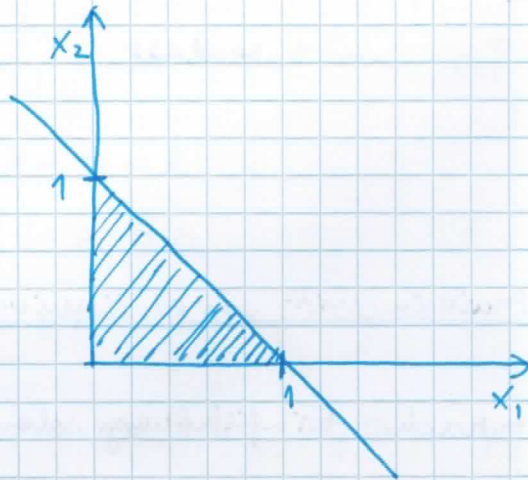
minimiere $C^T \cdot x$

so dass $x_1 + x_2 \leq 1$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

- das ist ein LP in \mathbb{R}^2 mit der Lösungsmenge



- das äquivalente LP in Standardform:
(die Nebenbedingungen)

$$x_1 + x_2 + s_1 = 1$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$$s_1 \geq 0$$

Wie ändert sich das Lösungspolyeder?

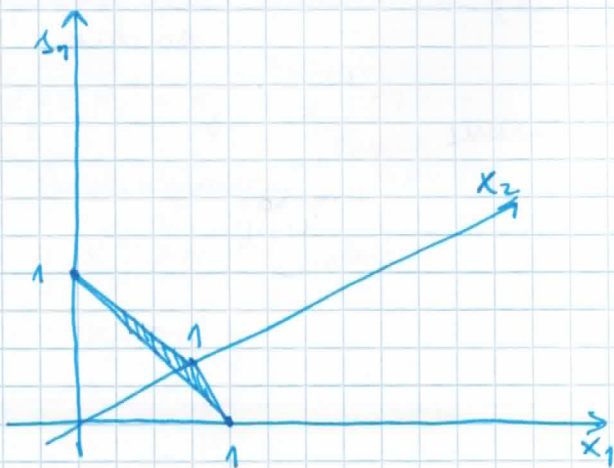
die Standardform ist ein LP in \mathbb{R}^3

die Gleichung

$$x_1 + x_2 + s_1 = 1$$

bestimmt eine affine

Ebene in \mathbb{R}^3



Beobachtung:

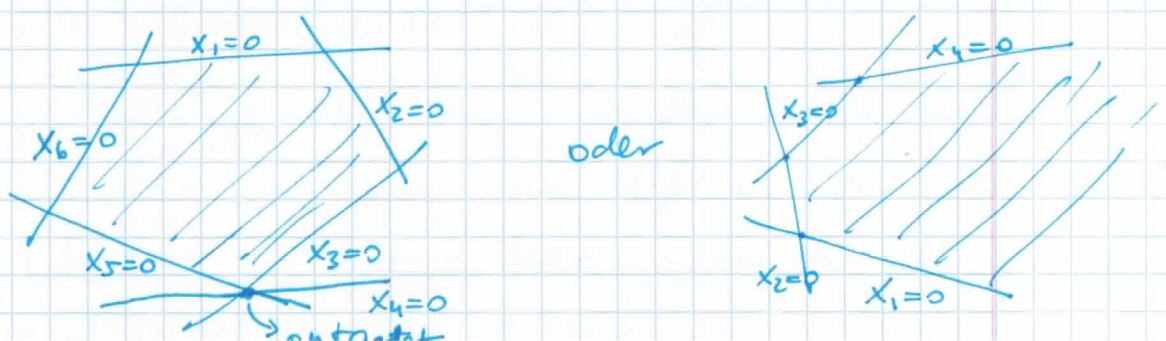
- die Hinzunahme von (jeder) s_1 zu den Variablen hat die Dimension n erhöht $\begin{pmatrix} x_1 \\ x_2 \\ s_1 \end{pmatrix}$ wird gesucht
- aber gleichzeitig (jede) Bedingung die als Gleichung erfüllt werden muss, reduziert die Dimension des Lösungsraums.
- ⇒ unser 2-dimensionaler Lösungsraum ist (eigentlich) 2-dimensional geblieben (eine affine Ebene enthält den Lösungsraum)
- (übrigens gilt das selbe nicht für die Transformation $x = x^+ - x^-$ diese erhöht die Dimension, und kann z.B. eine neue Ecke erzeugen.)

In der Standardform enthält die Lösungsmenge keine Gerade (wegen $x \geq 0$), und sie hat immer einige Ecken. Wie werden wir das Lösungspolyeder darstellen / uns vorstellen können?

Sei $Ax=b$ $x \geq 0$ ein LP in Standardform, mit n Variablen und m Gleichungen in $Ax=b$.

Das Lösungspolyeder P ist $n-m$ dimensional (falls $P \neq \emptyset$ und die a_i^T linear unabhängig sind)

(Wir denken an Beispiele mit $n-m=2$ oder $n-m=3$, und stellen uns die Lösungsmenge (für $n-m=2$) so vor:



LP25.

- die m Gleichungen werden in jeder Lösung x ^{exakt} erfüllt,
und diese bestimmen keine Kante des Polyeders

- der Lösungsraum von $x \geq 0$ ist n -dimensional;
die Bedingungen $Ax=b$ zwingen P in einen $n-m$
dimensionalen (affinen) Raum. Jede Bedingung nimmt
eine Dimension (einen Freiheitsgrad) aus n weg.

- in einer Ecke $x_j=0$ gilt für $n-m$ Komponenten j
(in einer entarteten Ecke $x_j=0$ für $>n-m$ Komponenten j)

~~und die $n-m$ Komponenten j sind~~

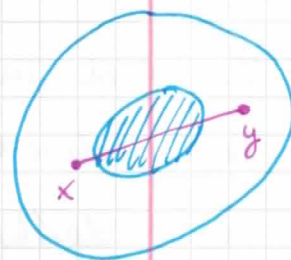
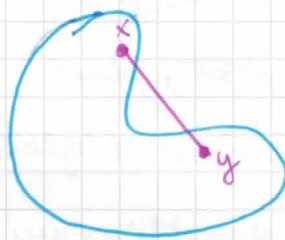
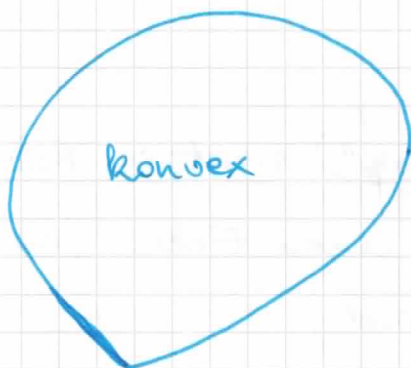
Erklärung: x^* ist eine Ecke, wenn $\geq n$ Nebenbedingungen
(und n unabhängige darunter) in x^* exakt (als Gleichung)
erfüllt werden. die m Gleichungen in $A \cdot x^* = b$
werden schon (automatisch) exakt erfüllt. Deshalb
braucht man noch $n-m$ von den n Ungleichungen
 $x_j \geq 0$ die in x^* exakt erfüllt werden. Für diese
 $n-m$ Komponenten j gilt $x_j^* = 0$ in der Ecke x^*

Konvexität:

eine Menge $S \subseteq \mathbb{R}^n$ ist konvex, wenn

für jede $x \in S$ und $y \in S$, die x und y verbindende

Strecke $\{ \lambda x + (1-\lambda)y \mid 0 \leq \lambda \leq 1 \} \subseteq S$
auch zur S gehört



nicht konvex

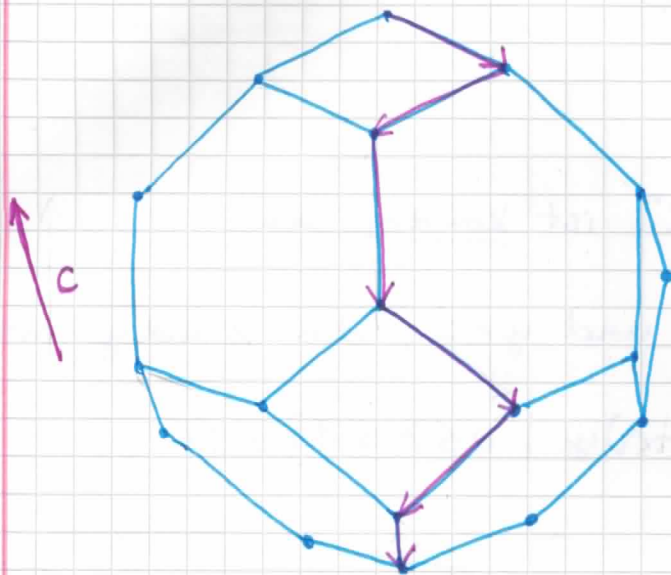
- ein Halbraum ist immer konvex
 - der Durchschnitt von konvexen Mengen ist immer konvex.
- \Rightarrow da der Lösungsraum der Durchschnitt von Halbräumen ist, ist er immer konvex

Der Simplex Algorithmus

Die Simplex Methode war der erste Algorithmus für die Lösung von linearen Programmen.

Gegeben ist ein LP in der Standardform als Eingabe.

Betrachte das Lösungspolyeder ...

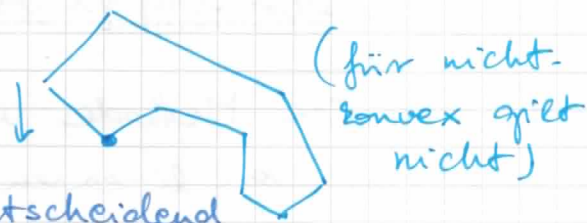


Es wird ausgenutzt, dass

- der Zielwert linear ("gleichmäßig") sinkt in Richtung $-c$ und deshalb das Minimum in einer Ecke (am weitesten in Richtung $-c$) angenommen wird;
- das Lösungspolyeder konvex ist.

Der Simplex Algorithmus ist greedy (bzw. macht lokale Suche) und läuft immer auf eine benachbarte Ecke "in Richtung $-c$ " weiter.

Aus der Konvexität des Polyeders folgt, dass falls er nicht weiterlaufen kann (d.h. er ist in einem lokalen Minimum angekommen), ist er schon in einem global minimalen Punkt.



Es kann für die Laufzeit entscheidend werden, auf welche (bessere) benachbarte Ecke der Algorithmus jeweils weiterläuft. Die Regel, die die nächste Ecke für eine beliebige aktuelle Ecke auswählt, heißt Pivot Strategie (pivoting rule)

Laufzeit?

In jeder Iteration werden Matrizen (Vektoren)
invertiert, multipliziert u. addiert. $\rightarrow \text{Poly}(n, m)$

Aber: Wie hoch ist die Anzahl der Iterationen
d.h. wieviele Ecken durchläuft ^{der} Simplex-Alg.
im Worst-Case?

Wieviele Ecken gibt es (asymptotisch) überhaupt?

z.B. der Würfel ist das Königspolyeder für

$$0 \leq x_i \leq 1 \quad i=1, 2, \dots, n$$

Wieviele Ecken hat der Würfel?

Jeder 0-1 Vektor ist eine Ecke $\Rightarrow 2^n$ Ecken

Es gibt tatsächlich worst-case Beispiele so dass
der Simplex Algorithmus jede Ecke eines verzerrten

Würfels durchläuft

Thm:

~~Der~~ Der Simplex-Algorithmus hat im Worst-Case

~~$\Omega(2^n)$~~ $\Omega(2^n)$ Iterationen, und ist somit kein
polynomieller Algorithmus. $\text{poly}(n) \cdot 2^n$ ist exponentielle
Laufzeit

Jedoch wird der Simplex-Algorithmus in der Praxis
erfolgreich verwendet!

[Genauer: für jede Pivot-Strategie gibt es Instanzen die
zu exponentieller Laufzeit führen.

Geschichte:

- 1947 Dantzig der Simplex-Algorithmus

- 1979 Khachiyan der erste polynomielle Algorithmus:
die Ellipsoid-Methode

führt letztendlich indirekt zu einem praktischen polynomiellen Algorithmus:

- 1984 Karmarkar das Innen-Point Verfahren

(gut für große Programme, wenn eine approximative Lösung ausreicht)

f.) Anwendungsbeispiele:Lineare Programmierung und Approximation

1.) VERTEX COVER

Eingabe: $G(V, E)$ mit Knotengewichten w_v Ausgabe: eine Knotenüberdeckung $C \subseteq V$ mit minimalem Gesamtgewicht der Knoten in C LP-Relaxierung des Problems:

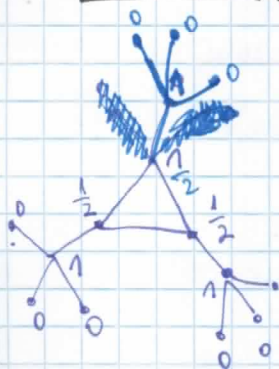
minimiere
$$\sum_{v=1}^n x_v w_v$$

so dass
$$x_v + x_u \geq 1 \quad \forall \{u, v\} \in E$$

$$0 \leq x_v \leq 1 \quad \forall v \in V$$

(beabsichtigte Bedeutung:
$$x_v = \begin{cases} 1 & \text{falls } v \in C \\ 0 & \text{falls } v \notin C \end{cases}$$
)

Für manche Probleme gibt die fraktionale Lösung (die optimale Lösung der LP-Relaxierung) Hinweise darüber, wie eine gute Lösung des ganzzahligen Problems tendenziell aussieht:

Ein Approximationsalgorithmus (deterministisches Runden)

- löse das LP;

sei $x^* = (x_1^*, x_2^*, \dots, x_n^*)$ die optimale fraktionale Lösung- setze $v \in C$ (in die Knotenüberdeckung)genau dann, wenn $x_v^* \geq \frac{1}{2}$

Behauptung 1: Diese C ist eine Knotenüberdeckung.

Wann? Für jede Kante $\{u, v\}$ gilt

$$x_u^* + x_v^* \geq 1 \Rightarrow \text{entweder } x_u^* \geq \frac{1}{2} \text{ und}$$

$$\rightarrow u \in C$$

$$\text{oder } x_v^* \geq \frac{1}{2}$$

$$\Rightarrow v \in C$$

Behauptung 2: Der Algorithmus mit deterministischem Runden ist 2-approximativ.

Beweis:

das Optimum der LP-Relaxierung ist nicht schlechter als das ganzzahlige Optimum VC_{OPT}

$$\sum_{v=1}^n w_v \cdot x_v^* \leq VC_{\text{OPT}}$$

Für jeden Knoten $v \in C$ in der Angabe gilt

$$\frac{1}{2} \leq x_v^*$$

$$1 \leq 2 \cdot x_v^*$$

$$\text{deshalb: } VC_{\text{ALG}} = \sum_{\{v: x_v^* \geq \frac{1}{2}\}} 1 \cdot w_v \leq \sum_{\{v: x_v^* \geq \frac{1}{2}\}} 2 \cdot x_v^* \cdot w_v \leq \sum_{v=1}^n 2 \cdot x_v^* \cdot w_v = 2 VC_{\text{OPT}}$$

Wert als
ganzzahlige
Lösung

Es gilt auch: jede Ecke des Lösungspolyeders des LP hat nur Einträge aus $\{0, \frac{1}{2}, 1\}$

Wann? Sei (x_1, x_2, \dots, x_n) eine fraktionale Lösung

$$\text{Sei } \epsilon > 0 \text{ so dass } 0 < x_i < \frac{1}{2} \Rightarrow 0 < x_i \pm \epsilon < \frac{1}{2}$$

$$\frac{1}{2} < x_i < 1 \Rightarrow \frac{1}{2} < x_i \pm \epsilon < 1$$

$$\text{Sei } y_i = \begin{cases} \epsilon & \text{falls } \frac{1}{2} < x_i < 1 \\ -\epsilon & \text{falls } 0 < x_i < \frac{1}{2} \\ 0 & \text{sonst} \end{cases}$$

denn gilt $x+y$ und $x-y$ sind auch Lösungen $\Rightarrow x$ war keine Ecke

2.) INDEPENDENT SET (wo Kunden (auch) nicht hilft)

Eingabe: $G(V, E)$ Ausgabe: eine unabhängige Knotenmenge $I \subseteq V$
mit maximaler Anzahl von Knoten

(I darf keine benachbarte Knoten enthalten)

LP-Relaxierung:

maximiere $\sum_{v=1}^n x_v$

so dass $x_u + x_v \leq 1 \quad \forall \{u, v\} \in E$

$0 \leq x_v \leq 1$

Wir haben gesehen: beim VC ist das ganzzahlige Optimum höchstens um Faktor 2 höher als das fractionale Optimum. Dies ist nicht notwendigerweise (d.h. für jedes Problem) so:

Beobachtung: Im Fall von INDEPENDENT SET gibt es Graphen mit fraktionalem Lösungen, die um den Faktor $\Omega(n)$ besser sind als das ganzzahlige Optimum.

Beispiel: Ein vollständiger Graph:

- das ganzzahlige Optimum: 1

$(1, 0, 0, 0, 0, \dots, 0)$

- das fractionale Optimum:

$(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2})$

mit Zielerwartung $\frac{n}{2}$

Das (Ab)wenden dieser fraktionalem Lösung ergibt keine wertvolle ganzzahlige Lösung.



5.) Ein ROUTING Problem (Anwendung in Rechnernetzen)

Eingabe: ein gerichteter Graph $\vec{G}(V, \vec{E})$
mit ausgezeichneten Knoten

$s_1 s_2 \dots s_r$ (Quellen)

$t_1 t_2 \dots t_r$ (Senken)

Ausgabe: Bestimme gerichtete Pfade $P_1 P_2 \dots P_r$, so dass

- P_i in s_i beginnt, und in t_i endet
- die maximale Belastung über alle Kanten minimal ist.

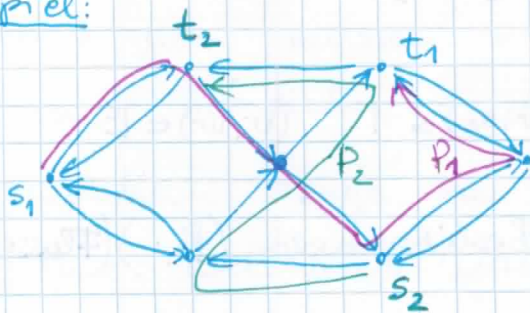
Belastung(e) = Anzahl der P_i , die über e laufen

eine IP-Formulierung des Problems

die Variablen: $x_i(e)$ für $1 \leq i \leq r$ und $e \in \vec{E}$

Bedeutung: $x_i(e) = \begin{cases} 1 & \text{wenn } P_i \text{ über } e \text{ läuft} \\ 0 & \text{sonst} \end{cases}$

Beispiel:

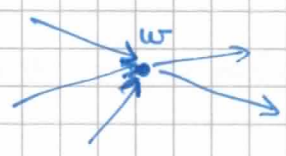


Sei i fixiert

Um auszudrücken, dass die $x_i(e)$ ($e \in E$) einem $s_i - t_i$ Pfad P_i entsprechen sollen, haben wir für jeden Knoten $w \in V$ eine Bedingung (der Pfad P_i darf Knoten wiederholt besuchen):

Wenn $w \neq s_i, t_i$, dann soll P_i den Knoten w jedes mal auch verlassen, wenn er ihn besucht:

$$\sum_{\{e | e \text{ führt in } w\}} x_i(e) - \sum_{\{e | e \text{ startet in } w\}} x_i(e) = 0 \quad \forall w \neq s_i, t_i, w \in V$$



|V| viele Flusserhaltung Bedingungen

(B_i)

$$\sum_{\{e | e \text{ startet in } s_i\}} x_i(e) - \sum_{\{e | e \text{ führt in } s_i\}} x_i(e) = 1 \quad \text{für } s_i$$

$$\sum_{\{e | e \text{ führt in } t_i\}} x_i(e) - \sum_{\{e | e \text{ startet in } t_i\}} x_i(e) = 1 \quad \text{für } t_i$$

Die (B_i) Bedingungen brauchen wir für jedes $s_i - t_i$ Knotenpaar, diese ergeben insgesamt $|V| \cdot r$ Bedingungen (für $1 \leq i \leq r, w \in V$).

Unsere Zielfunktion ist eigentlich:

Minimiere $\max_e \underbrace{\sum_{i=1}^r x_i(e)}_{\text{Belastung über Kante } e}$ Diese ist aber

keine lineare Zielfunktion! Wir führen eine neue Variable W ein die der maximalen Belastung entspricht.

LP 49/a

Die Zielfunktion ist dann

Minimiere W

mit den weiteren Bedingungen (E Bedingungen):

$$\sum_{i=1}^r x_i(e) \leq W \quad \forall e \in E$$

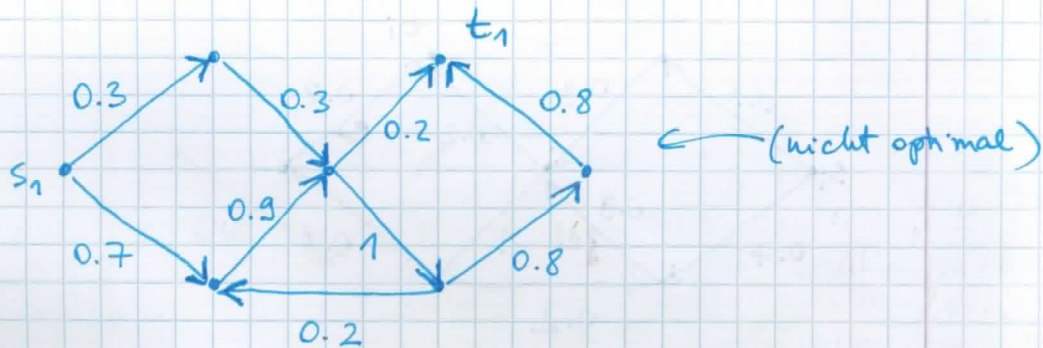
$$x_i(e) \in \{0, 1\}$$

LP-Relaxierung: $0 \leq x_i(e) \leq 1 \quad \forall e \in E$

In der LP-Relaxierung die Bedingungen (B_i) für Flusshaltung fordern, dass von s_i nach t_i ein Fluss von Wert 1 entsteht (eine fraktionale Lösung heißt also 'Fluss'):

(Definition): ein Fluss von s_i nach t_i ist eine Funktion $f: E \rightarrow \mathbb{R}_{0,+}$ (hier $f = x_i$) so dass die Flusserhaltung (B_i) gilt (es gibt meistens auch Kapazitätsschranken $f(e) \leq c(e)$ für jede Kante e).
(im allgemeinen Flussproblem)

Beispiel: der folgende Fluss erfüllt die Bedingungen (B_1) (eine fraktionale Lösung der $x_i(e)$ Werte)



Eine (optimale) Lösung der LP-Relaxierung bestimmt also jeweils einen Fluss x_i von s_i nach t_i .

Die Idee: bestimmen wir eine ganzzahlige Lösung, dh. jeweils einen einzigen Pfad P_i aus dieser optimalen fraktionalen Lösung durch randomisiertes Runden.

→ wir müssen zufällig einen Pfad P_i aus mehreren möglichen Pfaden auswählen. ABER: aus welcher Menge von Pfaden?

Mit welchen Wahrscheinlichkeiten?

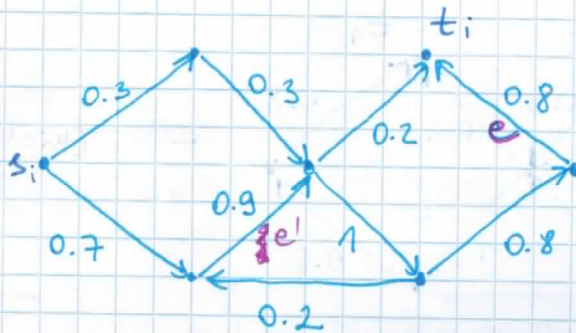
Wie entstehen mögliche Pfade und Wahrscheinlichkeiten aus einer fraktionalen Lösung?

Pfad-Zerlegung (~~Path~~ path-decomposition)

Beispiel: Wir bestimmen eine Pfad-Zerlegung der obigen fraktionellen Lösung. (Wir lassen Fixiere den Index $i \neq 1$ ~~von der Notation weg.~~)

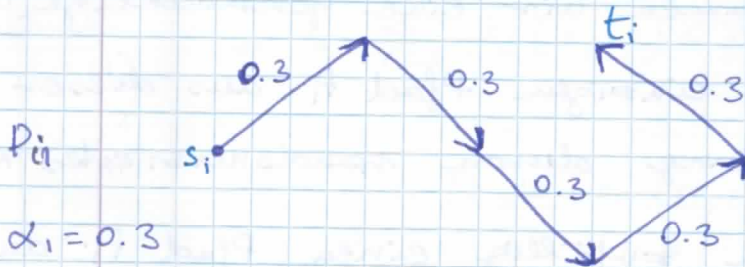
~~in P_i~~

- Betrachten wir den Teilgraph aller Kanten mit $x_i(e) > 0$.

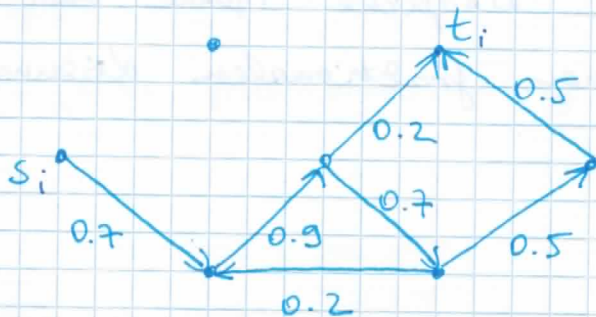


- Wir nehmen einen Pfad P_1 von s_i nach t_i und α_1 den minimalen Wert des Flusses entlang des Pfades.

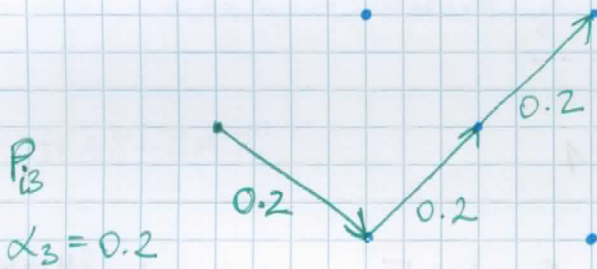
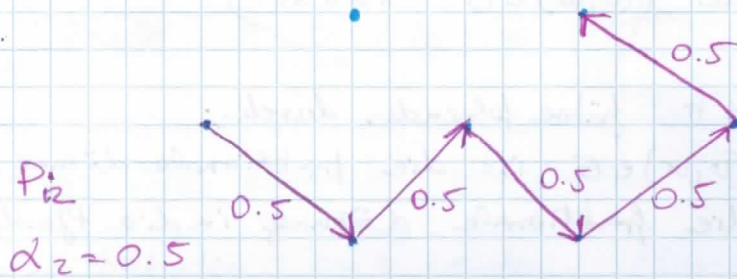
(es gibt einen s_i-t_i Pfad wegen der Flusserkhaltung)



Schließlich reduzieren wir die entsprechenden Flusswerte um $0.3 = \alpha_1$ und erhalten einen Fluss mit Wert $1 - \alpha_1$ von s_i nach t_i .

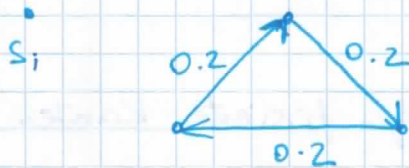


- wir iterieren diesen Prozess solange es einen $s_i - t_i$ Pfad mit positivem Flusswert gibt
 \Leftrightarrow solange $\alpha_1 + \alpha_2 + \dots < 1$



Es gilt: $\alpha_1 + \alpha_2 + \alpha_3 = 1$

- der gebliebene Graph enthält keinen $s - t$ Pfad



- Wieviele Iterationen gibt es?

\rightarrow in jeder Runde wird mindestens eine Kante aus dem Graphen entfernt, und keine Kante kommt hinzu.

\Rightarrow es gibt höchstens $|E|$ Iterationen.

- Wie soll der Algorithmus randomisiert einen Pfad von s_i nach t_i auswählen? \rightarrow mit Prob. α_j wird P_j gewählt.

- die Prob., dass Kante e benutzt wird ist dann $0.5 + 0.3 = 0.8$

- die erwartete Belastung von e wegen P_i ist somit

$1 \cdot 0.8 + 0 \cdot 0.2 = 1 \cdot x_i(e)$ (allgemein $\leq x_i(e)$) \rightarrow z.B. für $e!$ Prob = $0.7 < x_i(e)$

Algorithmus für ROUTING (randomisiertes Runden) ^{LP 53.}

→ löse die LP-Relaxierung; seien die Werte einer optimalen fraktionalem Lösung $x_i^*(e)$ (für $e \in E$, $1 \leq i \leq r$)

→ für jede $1 \leq i \leq r$ führe Folgendes durch:

- $x_i^*(e)$ ($e \in E$) ist die fraktionalem Lösung (ein Fluss) für $s_i - t_i$

- zerlege diese fraktionalem Lösung in Pfade

$P_{i,1}$ $P_{i,2}$ $P_{i,3}$... mit den entsprechenden Fluss-Werten $\alpha_{i,1}$ $\alpha_{i,2}$ $\alpha_{i,3}$...

- es gilt $\sum_j \alpha_{i,j} = 1$

- Wähle den $s_i - t_i$ Pfad P_i randomisiert, so dass

$P_i := P_{i,j}$ mit Wahrscheinlichkeit $\alpha_{i,j}$

→ Da wir eine Pfad-Zerlegung des Flusses $x_i^*(e)$ ($e \in E$) haben, gilt für jede einzelne Kante $e \in E$, dass

$$\sum_{\{j \mid e \in P_{i,j}\}} \alpha_{i,j} \leq x_i^*(e)$$

↑
Flusswerte aller Pfade, die die Kante e benutzen

→ Die Wahrscheinlichkeit, dass der zufällig gewählte P_i eine konkrete Kante e enthält, ist genau

$$\text{Prob}(P_i \text{ benutzt } e) = \sum_{\{j \mid e \in P_{i,j}\}} \alpha_{i,j} \leq x_i^*(e)$$

(die Flusswerte =) die Wahrscheinlichkeiten der Pfade, die die Kante e benutzen

→ Die erwartete Belastung über Kante e wegen des Pfades P_i (!)

ist (die durch P_i verursachte
 → Belastung falls P_i benutzt e ist 1, sonst ist es 0)

$$1 \cdot \text{Prob}(P_i \text{ benutzt } e) + 0 \cdot \text{Prob}(P_i \text{ benutzt } e \text{ nicht}) =$$

$$= \text{Prob}(P_i \text{ benutzt } e) \leq x_i^*(e)$$

→ Die erwartete Belastung über e wegen aller Pfade

$P_1, P_2, P_3, \dots, P_i, \dots, P_r$ ist

$$\leq x_1^*(e) + x_2^*(e) + \dots + x_r^*(e) \leq W_{\text{OPT}}^{\text{frac}} \leq W_{\text{OPT}}$$

[Genauer: die Zufallsvariablen werden definiert: (e fixiert)

$$y_i = \begin{cases} 1 & \text{falls } P_i \text{ benutzt } e \\ 0 & \text{sonst} \end{cases} \quad (\text{Indikatorvariablen})$$

$$y = \sum_{i=1}^r y_i \quad \text{ist die Belastung über Kante } e \text{ insgesamt}$$

(y ist auch Zufallsvariable).

$$\text{Es gilt } \mathbb{E}[y] = \sum_{i=1}^r \mathbb{E}[y_i] \leq \sum_{i=1}^r x_i^*(e) \leq W_{\text{OPT}}^{\text{frac}} \leq W_{\text{OPT}} \quad]$$

Thm: Die maximale erwartete Belastung über alle Kanten e

(max über Kanten) ist somit $\leq W_{\text{OPT}}$.

Bemerkung: Die erwartete maximale Belastung über Kanten

(Erwartung über jeweils einer schlechtesten Kante) kann

trotzdem groß sein. Aber, mit $\text{Prob} > 1 - \varepsilon$ ist die

$$\text{maximale Belastung} \leq W_{\text{OPT}} + \sqrt{3 W_{\text{OPT}} \cdot \ln \frac{n^2}{\varepsilon}}$$