3. Das minimale Kosten Netzwerkflussproblem

Ein gerichteter Graph $G = (V, E, u, c, b)$ mit

$c: E \to \mathbb{R}$, c_{ij} Kosten bzgl. Kante $(i,j) \in E$

$u: E \to \mathbb{R}^+$, u_{ij} Kapazität der Kante $(i,j) \in E$

$b: V \to \mathbb{R}$ mit

$$b(i) = \begin{cases} \text{Angebot des Knotens } i & \text{falls } b(i) \geq 0 \\ \text{Nachfrage des Knotens } i & \text{falls } b(i) \leq 0 \end{cases}$$

heißt (allgemeines) Flußnetzwerk

Ein Knoten $i \in V$ mit $b(i) = 0$ heißt Durchgangs=knoten.

Folgendes lineare Programm definiert das minimale Kosten Netzwerkflussproblem:

$$\text{minimiere} \sum_{(i,j) \in E} c_{ij} x_{ij} =: z(x)$$

wobei

$$(1) \quad \sum_{j:(i,j) \in E} x_{ij} - \sum_{j:(j,i) \in E} x_{ji} = b(i) \quad \forall i \in V$$

(Flussbalancierungsbedingung)

$$(2) \quad 0 \leq x_{ij} \leq u_{ij} \qquad \forall (i,j) \in E$$

(Kapazitätsbedingung)

Der Vektor $x = (x_{ij})$ definiert den <u>Fluss</u> im Netzwerk. Sei N die Knoten/Kanten-Inzidenzmatrix von G. Dann können wir obiges lineare Programm folgendermaßen schreiben:

$$\begin{aligned} \min\ & cx \\ Nx &= b \\ x &\leq u \\ x &\geq 0 \end{aligned}$$

<u>Beobachtung:</u>

Obiges lineare Programm kann nur dann eine zulässige Lösung besitzen, wenn

$$\sum_{\substack{i \in V \\ b(i) > 0}} b(i) = - \sum_{\substack{i \in V \\ b(i) < 0}} b(i) .$$

D.h., Gesamtangebot = Gesamtnachfrage.

<u>Spezialfälle:</u>

1) <u>Einzelquelle - kürzeste - Weg - Problem:</u>

<u>gegeben:</u> gerichteter Graph $G = (V, E, c)$
$c : E \to \mathbb{R}$, wobei c_{ij} die Länge der Kante (i,j) bezeichnet
$V = \{1, 2, \ldots, n\}$

<u>gesucht:</u> Ein kürzester Weg von 1 nach j $\forall j \in V$

Definiere

$$b(1) := n-1$$
$$b(i) := -1 \quad \forall\, i \in V \setminus \{1\}$$
$$u_{ij} := n \quad \forall\, (i,j) \in E$$

Eine optimale Lösung des daraus resultierende minimale Kosten Netzwerkflussproblems sendet vom Knoten 1 zu jedem anderen Knoten j den Fluss 1 entlang eines kürzesten Pfades.

2) <u>Maximale Flussproblem</u>

Sei $G = (V, E, u, s, t)$ ein Flussnetzwerk mit Quelle s und Senke t. Dann definieren wir

$$G' = (V, E', u, c, b), \text{ wobei}$$

$$E' := E \cup \{(t,s)\},$$
$$u_{ts} := \infty \;; \quad c_{ts} := -1$$
$$b(i) := 0 \quad \forall\, i \in V$$
$$c_{ij} := 0 \quad \forall\, (i,j) \in E$$

Die Lösung des korrespondierenden minimale Kosten Netzwerkflussproblems maximiert den Fluss über die Kante (t,s)

$\Rightarrow$

Dieser ist gleich dem maximalen Fluss von s nach t.

Übung

Beweisen Sie die Korrektheit unserer Reduktionen des Einzelquelle-kürzesten-Weg-Problems bzw. des maximalen Flussproblems auf das minimale Kosten Netzwerkflussproblem.

3.1 Entwicklung von Polynomzeitalgorithmen für Netzwerkflussprobleme

Üblicherweise wird die Laufzeit eines Algorithmus in Abhängigkeit zur Länge der Eingabe bestimmt.

Eingabelänge eines Netzwerkflussproblems:

$O((n+m)(\log n + \log C + \log U))$ Bits, wobei

- $n := |V|$, $m := |E|$,
- $C := \max\{c_{ij} \mid (i,j) \in E\}$ und
- $U := \max(\{u_{ij} \mid (i,j) \in E\} \cup \{|b(i)| \mid i \in V\})$

Die Laufzeit ist polynomiell, wenn sie durch ein Polynom in der Länge der Eingabe beschränkt ist. D.h., polynomielle Laufzeit kann von U und C abhängen. Ein Algorithmus ist stark polynomiell zeitbeschränkt, falls seine Laufzeit durch ein Polynom in n und m beschränkt ist, also nicht von C und U abhängt.

Annahme:

- Alle Daten sind ganzzahlig und der Algorithmus berechnet nur ganzzahlige Zwischenlösungen und Lösungen

Allgemeine Methoden:

a) geometrische Verbesserungsmethode

- verläuft in mehreren gleichartigen Phasen

Idee:

Falls die Verbesserung der aktuellen Lösung in jeder Iteration proportional zum Unterschied zur optimalen Lösung ist und jede Iteration selbst nur polynomielle Laufzeit benötigt, dann hat der Algorithmus selbst polynomielle Laufzeit.

Bezeichne H den maximalen Unterschied zweier zulässigen Lösungen.

Beispiel:

- maximales Flussproblem: $H \leq m \cdot U$
- minimale Kosten Netzwerkflussproblem: $H \leq m \cdot CU$

Lemma 3.1

Bezeichne z_k den Wert der Zielfunktion einer zulässigen Lösung eines Minimierungproblems, die in der k-ten Iteration eines Algorithmus berechnet wurde. Sei z^* der Wert der Zielfunktion einer optimalen Lösung. Falls der Algorithmus garantiert, dass stets

$$(z_k - z_{k+1}) \geq \alpha \cdot (z_k - z^*)$$

für eine Konstante $0 < \alpha < 1$, dann terminiert der Algorithmus nach $\leq \lceil \frac{\log H}{\alpha} \rceil$ Iterationen.

Beweis:

$(z_k - z^*)$ ist die nach der k-ten Iteration noch mögliche Verbesserung.

Beh.:

Nach weiteren $t := \lceil \frac{1}{\alpha} \rceil$ Iterationen hat sich die noch mögliche Verbesserung zumindest halbiert. D.h.,

$$z_{k+t} - z^* \leq \frac{1}{2}(z_k - z^*)$$

Bew. d. Beh.:

Es gilt:

$$(z_{k+t} - z^*) = (z_k - z^*) - \sum_{i=1}^{t} (z_{k+(i-1)} - z_{k+i})$$

$$\leq (z_k - z^*) - \alpha \sum_{i=1}^{t} (z_{k+(i-1)} - z^*)$$

z_{k+i} fällt monoton in i. Also gilt

$$\leq (z_k - z^*) - \alpha \cdot t\,(z_{k+t} - z^*)$$

$$\leq (z_k - z^*) - (z_{k+t} - z^*)$$

$$\Leftrightarrow (z_{k+t} - z^*) \leq \frac{1}{2}(z_k - z^*)$$

$\square$

Da H die maximal mögliche Verbesserung ist und jede Zwischenlösung ganzzahlig ist, terminiert der Algorithmus nach spätestens $(\log H) \cdot t$ Iterationen.

∎

b) Skalierungsmethode

Es gibt verschiedene Skalierungsmethoden. Wir betrachten hier die einfachste.

Bitskalierung:

Idee:

Lösung eines Problems P durch die Lösung einer Folge $P_1, P_2, \ldots, P_k$ von Problemen, wobei P_i, $1 \le i \le k$ die Daten auf den i höchstwertigsten Bits approximiert. Die Lösung eines jeden nachfolgenden Problems liefert eine bessere Approximation der Lösung des ursprünglichen Problems bis schließlich $P_k = P$.

Für $2 \le i \le k$ dient eine optimale Lösung für P_{i-1} als Startlösung für das Problem P_i.

Wir werden später Skalierungsmethoden für Netzwerkflussprobleme entwickeln.

3.2 Basiseigenschaften von Netzwerkflüssen

Ziel:

Die Herleitung von zum Max-Flow Min-Cut Theorem analogen Sätzen für das minimale Kosten Netzwerkflussproblem.

Bisher:

- Definition des Netzwerkflussproblems über einen Fluss auf Kanten des Netzwerkes. Algorithmen zur Berechnung eines maximalen Flusses haben zulässigen Fluss durch Vergrößerung des Flusses entlang von Pfaden verbessert.

Jetzt:

- Zulässiger Fluss (d.h. alle Nachfragen und alle Angebote werden erfüllt) soll entlang von Pfaden und Kreisen verbilligt werden.

Gegeben, einen zulässigen Fluss in einem Flussnetzwerk möchten wir diesen auf alle Pfade und Kreise des Flussnetzwerkes verteilen.

Seien

- x_{ij} Fluss auf der Kante $(i,j) \in E$,
- P eine Aufzählung aller Pfade in G und
- Q eine Aufzählung aller Kreise in G.

Wir ordnen jedem Pfad $p \in P$ und jedem Kreis $q \in Q$ einen Flussanteil $h(p)$ bzw. $f(q)$ derart zu, so dass $\forall (i,j) \in E$ gilt:

$$x_{ij} = \sum_{p:(i,j)\in p} h(p) + \sum_{q:(i,j)\in q} f(q)$$

$\forall (i,j) \in E \ \forall p \in P \ \forall q \in Q$ definieren wir

$$\delta_{ij}(p) := \begin{cases} 1 & \text{falls } (i,j) \in p \\ 0 & \text{sonst} \end{cases}$$

und

$$\delta_{ij}(q) := \begin{cases} 1 & \text{falls } (i,j) \in q \\ 0 & \text{sonst} \end{cases}$$

Dann gilt $\forall (i,j) \in E$:

$$x_{ij} = \sum_{p\in P} \delta_{ij}(p) \cdot h(p) + \sum_{q\in Q} \delta_{ij}(q) f(q)$$

Demzufolge kann man den Flussvektor x durch den Pfadflussvektor h und den Kreisflussvektor f definieren. Eine derartige Repräsentation heißt <u>Pfad/Kreisfluss - Repräsentation</u> des Flusses.

<u>Satz 3.1</u> (Flusszerlegungseigenschaft)

a) Jeder gerichtete Pfad/Kreisfluss hat eine eindeutige Repräsentation als Kantenfluss.

b) Jeder nichtnegative Kantenfluss x kann durch einen gerichteten Pfad/Kreisfluss (nicht notwendigerweise eindeutig) mit folgenden Eigenschaften repräsentiert werden:

i) Jeder Pfad mit positiven Fluss verbindet einen Angebots- mit einem Nachfrage=knoten.

ii) Höchstens $n+m$ Pfade und Kreise haben einen $\neq$ Nullfluss, wovon höchstens m Kreise sind.

Beweis:

a) folgt direkt aus

$$x_{ij} = \sum_{p \in P} \delta_{ij}(p)\, h(p) + \sum_{q \in Q} \delta_{ij}(q)\, f(q)$$

$\forall\ (i,j) \in E.$

b) Wir beweisen die Aussage konstruktiv, indem wir eine gewünschte Flusszerlegung explizit definieren:

- Sei i_0 ein beliebiger Angebotsknoten. Dann existiert eine Kante (i_0, i_1) mit $x_{i_0 i_1} > 0$. Falls
  - i_1 Nachfrageknoten, dann ist der aktuelle Pfad p konstruiert. Stop.
  - Andernfalls impliziert die Flussbalancierungsbedingung bzgl. i_1, dass

eine Kante (i_1, i_2) mit $x_{i_1 i_2} > 0$ existiert.
Wiederhole obige Betrachtungsweise, bis

- Nachfrageknoten erreicht und somit der aktuelle Pfad p konstruiert sind oder
- ein bereits besuchter Knoten erreicht ist, d.h., ein Kreis q konstruiert wurde.

Nun modifizieren wir das aktuelle Flusswerte = werte bzgl. des gefundenen Pfades p bzw. des gefundenen Kreises q.

<u>1. Fall:</u> Pfad p vom Angebotsknoten i_0 zum Nachfrageknoten i_k wurde konstruiert.

Definiere:

$$h(p) := \min \{ b(i_0), -b(i_k), \min\{x_{ij} \mid (i,j) \in p\}\}$$

und

$$b(i_0) := b(i_0) - h(p),$$
$$b(i_k) := b(i_k) + h(p) \text{ und}$$
$$x_{ij} := x_{ij} - h(p) \quad \forall (i,j) \in p.$$

<u>2. Fall:</u> Kreis q wurde konstruiert.

Definiere

$$f(q) := \min \{ x_{ij} \mid (i,j) \in q \} \text{ und}$$
$$x_{ij} := x_{ij} - f(q) \quad \forall (i,j) \in q.$$

Wiederhole das Ganze auf dem neudefinierten Flussnetzwerk mit dem neudefinierten Fluss bis das Netzwerk keinen Angebots- und daher auch keinen Nachfrageknoten mehr enthält.

- Wähle Durchgangsknoten i mit mindestens einer ausgehenden Kante (i,j), so dass $x_{ij} > 0$ und wiederhole die obige Betrachtung mit dem Startknoten i.

$\Rightarrow$

Der Algorithmus findet einen Kreis q.

Wiederhole das Ganze, bis der neudefinierte Fluss der Nullfluss ist, d.h., $x_{neu} = 0$.

Beobachtung:

1) Jedesmal, wenn obiger Algorithmus einen Pfad findet, reduziert sich die Anzahl der Angebots/Nachfrageknoten oder eine Kante erhält Nullfluss.

2) Jedesmal, wenn ein Kreis konstruiert wird, erhält mindestens eine Kante Nullfluss.

$\Rightarrow$

$$\# \text{ Pfade } + \# \text{ Kreise mit } \neq \text{ Nullfluss } \leq n+m$$

und

$$\# \text{ Kreise mit } \neq \text{ Nullfluss } \leq m.$$

Ziel:

Verallgemeinerung der Flußzerlegungseigenschaft auf negative Kantenflüsse.

Seien

P eine Aufzählung der ungerichteten Pfade in G und

Q eine Aufzählung der ungerichteten Kreise in G.

Wir definieren für jeden ungerichteten Pfad p einen Anfangs- und einen Endknoten und assoziieren mit p die Orientierung vom Anfangs- zum Endknoten. Analog enthält jeder Kreis $q \in Q$ eine Orientierung.

Eine Kante e des Flussnetzwerkes, die auf p bzw q in der Orientierung von p bzw. q gerichtet ist, heißt Vorwärtskante bzgl. p bzw q. Ist eine Kante auf p bzw. q gegen die Orientierung gerichtet, dann heißt diese Rückwärtskante bzgl. p bzw. q.

Ein Pfadfluss auf p ist ein Fluss mit Wert $h(p)$ auf jeder Vorwärtskante und $-h(p)$ auf jeder Rückwärtskante. Analog wird Kreisfluss auf einem Kreis $q \in Q$ definiert.

$\forall (i,j) \in E \; \forall p \in P \; \forall q \in Q$ definieren wir

$$\delta_{ij}(p) := \begin{cases} 1 & \text{falls } (i,j) \text{ Vorwärtskante auf } p \\ -1 & \text{falls } (i,j) \text{ Rückwärtskante auf } p \\ 0 & \text{sonst} \end{cases}$$

und

$$\delta_{ij}(q) := \begin{cases} 1 & \text{falls } (i,j) \text{ Vorwärtskante auf } q \\ -1 & \text{falls } (i,j) \text{ Rückwärtskante auf } q \\ 0 & \text{sonst} \end{cases}$$

<u>Satz 3.2</u> (Flusszerlegungseigenschaft verallgemeinert)

a) Jeder ungerichtete Pfad/Kreisfluss hat eine eindeutige Repräsentation als Kantenfluss.

b) Jeder Kantenfluss x kann durch einen ungerichteten Pfad/Kreisfluss (nicht notwendigerweise eindeutig) mit folgenden Eigenschaften repräsentiert werden:

i) Jeder Pfad mit positiven Fluss verbindet einen Angebots- mit einem Nachfrageknoten.

ii) Auf jedem Pfad bzw. Kreis erscheint jede Kante mit positiven Fluss als Vorwärts- und jede Kante mit negativen Fluss als Rückwärtskante.

iii) Höchstens $n+m$ Pfade und Kreise haben keinen Nullfluss, wovon höchstens m Kreise sind.

<u>Beweis</u>:

Übung

■

Als nächstes werden wir eine Methode zur Verbesserung der aktuellen Lösung entwickeln. Dabei werden wir die Flusszerlegungseigenschaft anwenden.

Bezeichnung:

Sei $G = (V, E, u, c, b)$ ein Flussnetzwerk und x ein Fluss in G. Ein Kreis q in G mit $f(q) > 0$ heißt x-augmentierender Kreis, falls

$$0 \leq x_{ij} + \delta_{ij}(q)\, f(q) \leq u_{ij} \quad \forall (i,j) \in q.$$

Interpretation:
Der Fluss bleibt zulässig, falls ein positiver Flussanteil $\leq f(q)$ rund um den Kreis geschickt wird.

Die Kosten $c(q)$ eines augmentierenden Kreises q sind definiert durch

$$c(q) := \sum_{(i,j) \in E} \delta_{ij}(q) \cdot c_{ij}.$$

Interpretation:
$c(q)$ gibt die Kostenänderung an, falls eine Einheit Fluss um den Kreis geschickt wird.

Demnach ist die Kostenänderung bzgl. eines x-augmentierenden Kreises q stets

$$\leq c(q) \cdot f(q).$$

Seien x und y zwei beliebige zulässige Lösungen eines Netzwerkflussproblems, gegeben durch $G = (V, E, u, c, b)$. D.h.,

$$\begin{aligned} Nx &= b & \quad & & Ny &= b \\ x &\leq u & & \text{und} & y &\leq u \\ x &\geq 0 & & & y &\geq 0. \end{aligned}$$

Dann gilt für den Differenzvektor $z := y - x$

$$Nz = Ny - Nx = 0.$$

(Beachte: z kann negative Kantenflüsse enthalten.)

Also folgt aus den Flußzerlegungseigenschaften, dass z durch Kreisflüsse repräsentiert werden kann. D.h.,

$\exists\ r \leq m$ Kreisflüsse $f(q_1), f(q_2), \ldots, f(q_r)$, so dass für jede Kante $(i,j) \in E$ gilt:

$$z_{ij} = \delta_{ij}(q_1) f(q_1) + \delta_{ij}(q_2) f(q_2) + \ldots + \delta_{ij}(q_r) f(q_r)$$

Wegen $y = x + z$ gilt für jede Kante $(i,j) \in E$:

$$(*) \quad 0 \leq y_{ij} = x_{ij} + \left[\delta_{ij}(q_1) f(q_1) + \ldots + \delta_{ij}(q_r) f(q_r)\right] \leq u_{ij}$$

Satz 3.2 b ii) $\Rightarrow$

Eine Kante ist in jedem Kreis, der sie enthält, Vorwärtskante oder in jedem Kreis, der sie enthält

Rückwärtskante. Also haben alle Summanden $\delta_{ij}(q_\ell) f(q_\ell)$, $1 \le \ell \le r$ in $(*)$ dasselbe Vorzeichen.

$\Longrightarrow$

Für jeden Kreis q_k für jede Kante $(i,j) \in q_k$ gilt

$$0 \le x_{ij} + \delta_{ij}(q_k) f(q_k) \le u_{ij}.$$

Interpretation:

Falls wir irgendeinen dieser Kreisflüsse $f(q_k)$ auf x addieren, dann bleibt die resultierende Lösung zulässig.

$\Longrightarrow$

Jeder Kreis $q_1, q_2, \dots, q_r$ ist ein x-augmentierender Kreis.

Weiterhin gilt:

$$\sum_{(i,j)\in E} c_{ij} y_{ij} = \sum_{(i,j)\in E} c_{ij} x_{ij} + \sum_{(i,j)\in E} c_{ij} z_{ij}$$

$$= \sum_{(i,j)\in E} c_{ij} x_{ij} + \sum_{(i,j)\in E} c_{ij} \left(\sum_{k=1}^{r} \delta_{ij}(q_k) f(q_k) \right)$$

$$= \sum_{(i,j)\in E} c_{ij} x_{ij} + \sum_{k=1}^{r} c(q_k) \cdot f(q_k).$$

Insgesamt haben wir folgenden Satz bewiesen:

<u>Satz 3.3</u> (augmentierende Kreiseigenschaft)

Seien x und y zwei beliebige zulässige Lösungen eines Netzwerkflussproblems $G = (V, E, u, c, b)$ und $m := |E|$. Dann ist y gleich x zuzüglich dem Fluss auf $\leq m$ x-augmentierenden Kreisen. Die Kosten von y sind gleich den Kosten von x plus den Kosten des Flusses auf diesen x-augmentierenden Kreisen.

Aus Satz 3.3 ergibt sich leicht folgendes Optimalitätskriterium:

<u>Satz 3.4</u> (Optimalitätskriterium)

Ein zulässiger Fluss ist genau dann minimal, wenn er keinen augmentierenden Kreis mit negativen Kosten enthält.

<u>Beweis</u>:

Übung

■

Betrachte folgende LP-Formulierung des (allgemeinen) minimale Kosten Netzwerkflussproblems, die die Anforderung von Mindestflüssen auf Kanten beinhaltet:

$$\min \sum_{(i,j)\in E} c_{ij} x_{ij} \quad , \text{ wobei}$$

$$(*) \quad \sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = b(i) \quad \forall i \in V$$

$$(**) \quad l_{ij} \leq x_{ij} \leq u_{ij} \qquad \forall (i,j) \in E.$$

Annahme 1: $l_{ij} = 0$ und $c_{ij} \geq 0 \ \forall (i,j) \in E$.

Ziel:

Transformation des allgemeinen Problems in ein äquivalentes Problem, das die Annahme 1 erfüllt.

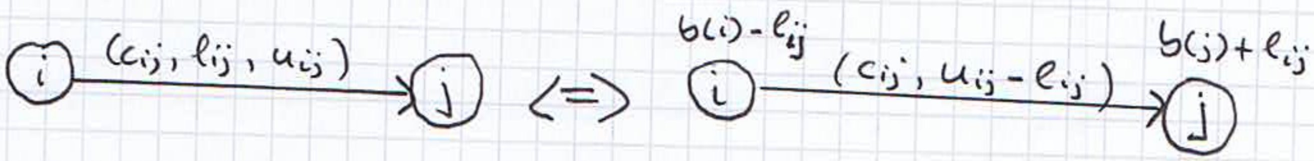
1. Transformation:

Entfernung unterer Schranken l_{ij} mit $l_{ij} > 0$.

- Ersetze in der Problemstellung x_{ij} durch $x'_{ij} + l_{ij}$

Effekt: x'_{ij} hat untere Schranke 0.

Netzwerkinterpretation:



Es wird l_{ij} Fluss über die Kante (i,j) geschickt und lediglich der über l_{ij} hinausgehende Fluss gemessen.

2. Transformation

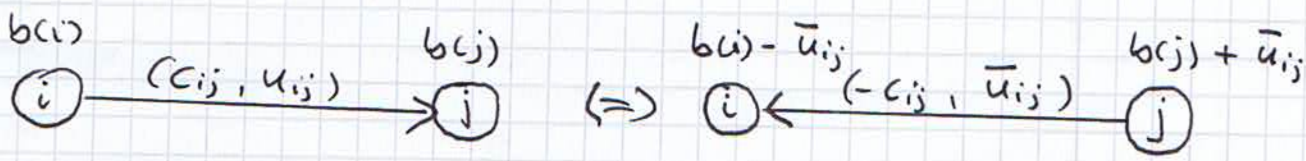
Elimination negativer Kosten c_{ij}

Sei

$$\bar{u}_{ij} := \begin{cases} u_{ij} & \text{falls } u_{ij} < \infty \\ \text{Obere Schranke für den Fluss auf } (i,j) & \text{sonst} \end{cases}$$

Ersetze x_{ij} durch $\bar{u}_{ij} - x_{ji}$ und ersetze die Kante (i,j) mit Kosten c_{ij} durch die Kante (j,i) mit Kosten $c_{ji} := -c_{ij}$

Netzwerkinterpretation



Der neue Fluss x_{ji} misst die Größe desjenigen Flusses, der von dem Fluss $\bar{u}_{ij}$ auf (i,j) wieder zurückgenommen wird.

Übung:
Beweisen Sie die Korrektheit der beiden Transformationen.

Seien

$$C := \max \{ c_{ij} \mid (i,j) \in E \} \quad \text{und}$$

$$U := \max \{ \max \{ |b(i)| \mid i \in V \}, \max \{ u_{ij} \mid (i,j) \in E \text{ und } u_{ij} < \infty \} \}.$$

Annahme 2: (Zulässigkeitsannahme)

Es existiert eine zulässige Lösung.

Übung:

Zeigen Sie, dass für ein beliebiges minimale Kosten Netzwerkflussproblem leicht mit Hilfe des maximalen Netzwerkflussproblems eine zulässige Lösung berechnet werden kann, falls solche existiert, bzw. festgestellt werden kann, dass keine zulässige Lösung existiert.

Annahme 3: (Zusammenhangsannahme)

$\forall i, j \in V$ gilt: Es existiert ein gerichteter Pfad unendlicher Kapazität zwischen i und j.

- Diese Annahme kann leicht mittels Hinzufügen der zusätzlichen Kanten $(v,j), (j,v)$ $\forall j \in V$, $v \notin V$ mit großen Kosten und unendlicher Kapazität erfüllt werden.

Falls vorher die Annahme 2 erfüllt war, dann kommt keine der zusätzlichen Kanten in einer optimalen Lösung vor.

Sei $G = (V, E, u, c, b)$ ein Flussnetwerk und x ein zulässiger Fluss in G. Das Restnetwerk $G_x = (V, E_x, r, c', b)$ ist definiert durch

$$E_x := \{(i,j) \mid (i,j) \in E \text{ und } r_{ij} := u_{ij} - x_{ij} > 0\}$$
$$\cup \{(j,i) \mid (i,j) \in E \text{ und } r_{ji} := x_{ij} > 0\}.$$

Die Kante (i,j) hat Kosten c_{ij} und <u>Restkapazität</u> $r_{ij} := u_{ij} - x_{ij}$. Die Kante (j,i) hat Kosten $-c_{ij}$ und Restkapazität $r_{ji} := x_{ij}$.

<u>Übung:</u>
Falls bereits im ursprünglichen Netzwerk für $i,j \in V$ sowohl (i,j) als auch (j,i) existieren, dann können im Restnetzwerk parallele Kanten vorkommen. Transformieren Sie das Netzwerk in ein äquivalentes Netzwerk, so dass das korrespondierende Restnetzwerk keine parallele Kanten enthält.

<u>Beobachtung:</u>
Jeder gerichtete Kreis Kreis im Restnetzwerk G_x ist ein augmentierender Kreis in G und umgekehrt.

<u>Satz 3.5</u> (negativer Kreis - Optimalitätskriterium)
Ein zulässiger Fluss x ist genau dann optimal, wenn das Restnetzwerk keinen gerichteten Kreis negativer Kosten enthält.

<u>Beweis:</u>

Übung

■

<u>Ziel:</u>
Herleitung weiterer Optimalitätskriterien.

$\forall i \in V$ bezeichne $\pi(i) \in \mathbb{R}$ das Potential des Knotens i. Für eine gegebene Menge π von Knotenpotentiale definieren wir für $(i,j) \in E$ die reduzierte Kosten c_{ij}^{π} der Kante (i,j) durch

$$c_{ij}^{\pi} := c_{ij} - \pi(i) + \pi(j).$$

Lemma 3.2

a) Für einen beliebigen Pfad P von Knoten k zum Knoten l gilt:

$$\sum_{(i,j)\in P} c_{ij}^{\pi} = \sum_{(i,j)\in P} c_{ij} - \pi(k) + \pi(l)$$

b) Für einen beliebigen gerichteten Kreis W gilt

$$\sum_{(i,j)\in W} c_{ij}^{\pi} = \sum_{(i,j)\in W} c_{ij}.$$

Beweis:

Übung

■

Satz 3.6 (reduzierte Kosten-Optimalitätskriterium)

Eine zulässige Lösung x^* ist genau dann eine optimale Lösung des minimale Kosten Netzwerkflussproblems, wenn es eine Menge π von Knotenpotentiale mit $c_{ij}^{\pi} \geq 0 \ \forall (i,j) \in G_{x^*}$ existiert.

Beweis:

Wir werden die Äquivalenz des reduzierte Kosten-Optimalitätskriteriums und des negativen Kreis-Optimalitätskriteriums beweisen.

Annahme:

x^* erfüllt das reduzierte Kosten-Optimalitäts-kriterium.

Dann gilt:

$$\sum_{(i,j)\in W} c_{ij}^{\pi} \geq 0 \quad \forall \text{ gerichtete Kreise } G_{x^*}.$$

Lemma 3.2 b) $\Rightarrow$

$$\sum_{(i,j)\in W} c_{ij}^{\pi} = \sum_{(i,j)\in W} c_{ij} \geq 0 \quad \forall \text{ gerichtete Kreise } W \text{ in } G_{x^*}.$$

Also enthält G_{x^*} keinen gerichteten Kreis negativer Kosten.

Annahme:

x^* ist eine zulässige Lösung mit G_{x^*} enthält keinen gerichteten Kreis negativer Kosten.

Ziel:

Definition von Knotenpotentiale π, so dass $c_{ij}^{\pi} \geq 0 \quad \forall (i,j) \in G_{x^*}$.

$\forall j \in V$ sei $d(j)$ die kürzeste Pfaddistanz des Knotens j vom Knoten 1 in G_{x^*}.

G_{x^*} enthält keinen gerichteten Kreis negativer Kosten

$\Rightarrow$

$$d(j) \leq d(i) + c_{ij} \quad \forall (i,j) \in G_{x^*}.$$

$$\Leftrightarrow \quad c_{ij} - (-d(i)) + (-d(j)) \geq 0$$

Definiere $\forall\, i \in V$:

$$\pi(i) := -d(i).$$

Dann gilt $c_{ij}^{\pi} \geq 0 \quad \forall (i,j) \in G_{x^*}$.

$\blacksquare$

Satz 3.6 charakterisiert einen <u>optimalen Fluss</u> x als einen Fluss, so dass

$$c_{ij}^{\pi} \geq 0 \quad \forall (i,j) \in G_x.$$

für irgendeine Menge π von Knotenpotentialen.

Analog könnte man eine <u>optimale Menge von Knotenpotentialen</u> π als eine Menge von Knotenpotentiale, so dass

$$c_{ij}^{\pi} \geq 0 \quad \forall (i,j) \in G_x$$

für irgendeinen zulässigen Fluss x in G definieren.

Dies hat folgende <u>ökonomische Interpretation</u>:

Bezeichne c_{ij} die Kosten für den Transport von einer Einheit Fluss vom Knoten i zum Knoten j über die Kante (i,j).

Interpretiere $\mu(i) := -\pi(i)$ als die Kosten um eine Einheit Fluss am Knoten i zu erhalten.

Dann sind

$$c_{ij} + \mu(i)$$

die Kosten, um eine Einheit Fluss zum Knoten i und dann über die Kante (i,j) zum Knoten j zu transportieren.

Betrachte das reduzierte Kosten-Optimalitätskriterium:

$$c_{ij}^{\pi} = c_{ij} - \pi(i) + \pi(j) \geq 0$$

$$\Longleftrightarrow \quad \mu(j) \leq c_{ij} + \mu(i).$$

Dies impliziert, dass eine Einheit Fluss nach j zu transportieren nicht teurer sein kann, als eine Einheit nach i und dann über die Kante (i,j) nach j zu transportieren.

<u>Satz 3.7</u> (komplementärer Schlupf-Optimalitätskriterium)
Eine zulässige Lösung x^* ist genau dann eine optimale Lösung des minimale Kosten Netzwerk-

flussproblems, wenn für eine Menge π von Knotenpotentiale die reduzierte Kosten und Flusswerte für jede Kante $(i,j) \in E$ folgende Bedingungen erfüllen:

i) $c_{ij}^{\pi} > 0 \Rightarrow x_{ij}^{*} = 0$

ii) $0 < x_{ij}^{*} < u_{ij} \Rightarrow c_{ij}^{\pi} = 0$

iii) $c_{ij}^{\pi} < 0 \Rightarrow x_{ij}^{*} = u_{ij}$.

<u>Beweis:</u>

Wir beweisen die Äquivalenz des reduzierte Kosten-Optimalitätskriteriums und des komplementären Schlupf-Optimalitätskriteriums.

„$\Rightarrow$"

<u>Annahme:</u>

Für ein Paar (x^{*}, π) gilt

$$c_{ij}^{\pi} \geq 0 \quad \forall \, (i,j) \in G_{x^{*}}.$$

Betrachte $(i,j) \in E$ beliebig. Wir unterscheiden drei Fälle:

<u>1. Fall:</u> $c_{ij}^{\pi} > 0$

Dann gilt wegen $c_{ji}^{\pi} = -c_{ij}^{\pi} < 0$

$(j,i) \notin G_{x^{*}}$

$\Rightarrow x_{ij}^{*} = 0$.

<u>2. Fall:</u> $0 < x^*_{ij} < u_{ij}$

$\Rightarrow (i,j), (j,i) \in G_{x^*}$.

$\Rightarrow c^{\pi}_{ij} \geq 0$ und $c^{\pi}_{ji} \geq 0$

Also impliziert $c^{\pi}_{ij} = - c^{\pi}_{ji}$

$$c^{\pi}_{ij} = c^{\pi}_{ji} = 0.$$

<u>3. Fall:</u> $c^{\pi}_{ij} < 0$

$\Rightarrow (i,j) \notin G_{x^*}$

$\Rightarrow x^*_{ij} = u_{ij}$.

„$\Leftarrow$"

Übung

■

Beide Komponenten eines Paares (x, π) mit

- x ist ein zulässiger Fluss,
- π ist eine Menge von Knotenpotentiale und
- $c^{\pi}_{ij} \geq 0 \quad \forall\, (i,j) \in G_x$

sind optimal. Für die Berechnung eines kosten-optimalen Flusses kennen wir bereits das zugehörige Optimierungsproblem. Es stellt sich nun die Frage, wie das zugehörige Optimierungsproblem zur Berechnung einer optimalen Menge von Knotenpotentiale aussieht. Wir werden dieses als lineares Programm formulieren.

Definition ⇒

Für jede optimale Lösung des Optimierungsproblems existiert ein Fluss x, so dass

$$\bar{c}_{ij}^{\pi} \geq 0 \quad \forall (i,j) \in G_x .$$

Problem:
Obige Randbedingungen hängen vom konkreten Fluss x, der unbekannt ist, ab. Besser wären Randbedingungen, die für alle $(i,j) \in E$ erfüllt sind.

Idee:
Für alle $(i,j) \in E$ arbeite derart eine Variable α_{ij} in die Randbedingungen ein, so dass für optimale π stets Werte für α_{ij}, bezüglich denen obige Randbedingungen erfüllt sind, existieren.

Durchführung:

Betrachte folgende modifizierte Randbedingungen

$$\begin{aligned} \bar{c}_{ij}^{\pi} + \alpha_{ij} &\geq 0 \\ \alpha_{ij} &\geq 0 \end{aligned} \qquad \forall (i,j) \in E .$$

Für optimales π und zugehörigen Fluss x haben folgende Werte für α_{ij}, $(i,j) \in E$ die gewünschte Eigenschaft:

$$\alpha_{ij} := \begin{cases} 0 & \text{falls } \bar{c}_{ij}^{\pi} \geq 0 \\ -\bar{c}_{ij}^{\pi} & \text{falls } \bar{c}_{ij}^{\pi} < 0 . \end{cases}$$

Nehmen wir also an, dass

$$(*) \quad \begin{aligned} c_{ij}^{\pi} + \alpha_{ij} &\geq 0 \\ \alpha_{ij} &\geq 0 \end{aligned} \qquad \forall (i,j) \in E$$

die Randbedingungen unseres Optimierungsproblems sind. Unschön ist noch, dass die Variablen π als Index vorkommen. Daher formen wir $(*)$ äquivalent um zu:

$$(**) \quad \pi(i) - \pi(j) - \alpha_{ij} \leq c_{ij} \qquad \forall (i,j) \in E$$

Offen ist noch die Definition einer geeigneten Ziel=funktion. Schön wäre es, wenn im Optimum die Werte der Zielfunktionen beider Optimierungs=problemen gleich wären.

Multiplikation von $(**)$ mit x_{ij} und Auf=summation über alle Kanten in E ergibt:

$$\sum_{(i,j)\in E} (\pi(i) - \pi(j))\, x_{ij} - \sum_{(i,j)\in E} \alpha_{ij} x_{ij} \leq \sum_{(i,j)\in E} c_{ij} x_{ij}$$

$$\Leftrightarrow \sum_{i \in V} \underbrace{\Big(\sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} \Big)}_{b(i)} \pi(i) - \sum_{(i,j)\in E} \alpha_{ij} x_{ij} \leq \sum_{(i,j)\in E} c_{ij} x_{ij}$$

$$\Leftrightarrow \sum_{i \in V} b(i)\, \pi(i) - \sum_{(i,j)\in E} \alpha_{ij} x_{ij} \leq \sum_{(i,j)\in E} c_{ij} x_{ij}.$$

Wegen $x_{ij} \leq u_{ij}$, $\alpha_{ij} \geq 0$ $\forall (i,j) \in E$ ist die linke Seite obiger Ungleichung minimal für $x_{ij} = u_{ij}$ $\forall (i,j) \in E$. Also impliziert obige Ungleichung

$$(***) \quad \sum_{i \in V} b(i)\pi(i) - \sum_{(i,j)\in E} u_{ij}\alpha_{ij} \leq \sum_{(i,j)\in E} c_{ij}x_{ij}$$

Demnach ist es sinnvoll, die Zielfunktion

$$\sum_{i \in V} b(i)\pi(i) - \sum_{(i,j)\in E} u_{ij}\alpha_{ij}$$

zu maximieren.

Insgesamt haben wir folgendes duales lineare Programm erhalten (Beachte: $c_{ij}^{\pi} = c_{ij} - \pi(i) + \pi(j)$):

$$\text{maximiere} \quad w(\pi,\alpha) := \sum_{i \in V} b(i)\pi(i) - \sum_{(i,j)\in E} u_{ij}\alpha_{ij}$$

$$\begin{aligned} \text{wobei} \quad \pi(i) - \pi(j) - \alpha_{ij} &\leq c_{ij} && \forall (i,j) \in E \\ \alpha_{ij} &\geq 0 && \forall (i,j) \in E \\ \pi(i) &\gtrless 0 && \forall i \in V. \end{aligned}$$

$\pi(i) \gtrless 0$ bedeutet, dass wir keine Restriktionen bezüglich $\pi(i)$ haben.

Die Ungleichung (***) impliziert, dass der Wert der Zielfunktion des dualen linearen Programms stets eine untere Schranke für den Wert der Zielfunktion des primalen linearen Programms ist.

<u>Ziel</u>: Beweis, dass im Optimum beide Werte gleich sind.

Betrachte hierin den Vektor π fest.

Frage:

Welche Wahl von α_{ij}, $(i,j) \in E$ maximiert den Wert der Zielfunktion?

$(*) \Rightarrow$

α_{ij}, $(i,j) \in E$ müssen derart gewählt werden, dass

$\alpha_{ij} \geq -c_{ij}^{\pi}$ und $\alpha_{ij} \geq 0 \quad \forall (i,j) \in E$.

Da der Koeffizient von α_{ij} in der Zielfunktion $-u_{ij} \leq 0$ ist, gilt:

Je kleiner der Wert von α_{ij}, desto größer der Wert der Zielfunktion.

$\Longrightarrow$

Im Optimum nehmen alle α_{ij}, $(i,j) \in E$ den Wert

$$\alpha_{ij} := \max\{0, -c_{ij}^{\pi}\}$$

an. Somit können wir in der Zielfunktion die Variablen α_{ij} mittels Substitution von $\max\{0, -c_{ij}^{\pi}\}$ für α_{ij} eliminieren und erhalten dann die Ziel=funktion:

$$(****) \quad w(\pi) := \sum_{i \in V} b(i)\pi(i) - \sum_{(i,j) \in E} u_{ij} \cdot \max\{0, -c_{ij}^{\pi}\}$$

Nun können wir beweisen, dass im Optimum beide Werte gleich sind.

Satz 3.8 (Dualitätssatz)

Sei $G = (V, E, u, c, b)$ ein Flussnetzwerk, das eine zulässige Lösung besitzt. Dann gibt es ein primal-dual Paar (x^*, π) von Lösungen mit $z(x^*) = w(\pi)$.

Beweis:

Sei x^* eine optimale Lösung für G.

Satz 3.7 $\Rightarrow$

x^* erfüllt zusammen mit einem Vektor π die komplementären Schlupfbedingungen.

Beh.:

Für diese Lösungen gilt

$$-c_{ij}^{\pi} x_{ij}^{*} = \max\{0, -c_{ij}^{\pi}\} u_{ij} \quad \forall (i,j) \in E.$$

Bew. d. Beh.:

Wir unterscheiden drei Fälle:

1) $c_{ij}^{\pi} > 0$:

Satz 3.7 $\Rightarrow$ $x_{ij}^{*} = 0$

Also gilt:

$$-c_{ij}^{\pi} x_{ij}^{*} = 0 = \max\{0, -c_{ij}^{\pi}\} u_{ij}$$
$$(\text{da } -c_{ij}^{\pi} < 0)$$

2) $c_{ij}^{\pi} = 0$:

Dann gilt $-c_{ij}^{\pi} x_{ij}^{*} = 0 = \max\{0, -c_{ij}^{\pi}\} u_{ij}$.

3) $\underline{c_{ij}^{\pi} < 0}$:

Satz 3.7 $\Rightarrow$ $x_{ij}^* = u_{ij}$

Also gilt:

$$-c_{ij}^{\pi} x_{ij}^* = \max\{0, -c_{ij}^{\pi}\} u_{ij}$$

$\square$

Betrachten wir nun die Zielfunktion (****) des dualen Problems:

$$w(\pi) = \sum_{i \in V} b(i)\pi(i) - \sum_{(i,j) \in E} \max\{0, -c_{ij}^{\pi}\} u_{ij}$$

obige Beh. $\Rightarrow$

$$w(\pi) = \sum_{i \in V} b(i)\pi(i) + \sum_{(i,j) \in E} c_{ij}^{\pi} x_{ij}^*$$

$$= \sum_{(i,j) \in E} \left[(\pi(i) - \pi(j)) x_{ij}^* + (c_{ij} - \pi(i) + \pi(j)) x_{ij}^* \right]$$

$$= \sum_{(i,j) \in E} c_{ij} x_{ij}^* = z(x^*)$$

$\blacksquare$

<u>Bemerkung</u>:

Satz 3.6 + Satz 3.7 + Satz 3.8 $\Rightarrow$

Wir können aus einer optimalen primalen Lösung eine optimale duale Lösung durch

- einmaliges Lösen des Einzelquelle-Kürzesten-Weg-Problems berechnen.

Ziel:

Konstruktion einer optimalen primalen Lösung aus einer optimalen dualen Lösung mittels

- einmaliges Lösen des maximalen Flussproblems.

Sei π eine optimale duale Lösung und c^π der Vektor der reduzierten Kosten.

Idee:

Definition eines Netzwerkes $G^* = (V, E^*, l^*, u^*, c, b)$ so dass ein beliebiger zulässiger Fluss in G^* das komplementäre Schlupf-Optimalitäts=kriterium erfüllt.

Durchführung:

- Die Knotenmarkierungen b in G^* und b in G sind identisch.

- E^* nebst unteren und oberen Schranken l^*, u^* sind wie folgt definiert.

  - $\forall (i,j) \in E: \; c_{ij}^\pi > 0$

    enthält E^* die Kante (i,j) mit $l_{ij}^* = u_{ij}^* = 0$.

  - $\forall (i,j) \in E: \; c_{ij}^\pi < 0$

    enthält E^* die Kante (i,j) mit $l_{ij}^* = u_{ij}^* = u_{ij}$

- $\forall (i,j) \in E : c_{ij}^{\pi} = 0$

enthält E^* die Kante (i,j) mit
$l_{ij}^* = 0$ und $u_{ij}^* = u_{ij}$.

Übung:
Führen Sie die Transformation auf das maximale Flussproblem zu Ende. Entwerfen Sie insbesondere einen Algorithmus, der, gegeben ein Flussnetzwerk und eine optimale duale Lösung, eine optimale primale Lösung ausgibt.

3.3 Algorithmen zur Lösung des minimalen Kosten Netzwerkflussproblems

3.3.1 Nichtpolynomielle Algorithmen

3.3.1.1 Negative Kreisalgorithmen

Idee:
Ausgehend von einer zulässigen primalen Lösung wird sukzessive, unter Beibehaltung der primalen Zulässigkeit, diese verbessert. Hierzu werden im Restnetzwerk G_x gerichtete Kreise negativer Kosten identifiziert und auf diesen Flüsse augmentiert. Wenn keine gerichteten Kreise negativer Kosten mehr im Restnetzwerk existieren, dann terminiert das Verfahren. Aus Satz 3.5 folgt dann, dass die gefundene Lösung optimal ist.