

5. FLOWS OVER TIME / DYNAMISCHE FLÜSSE

Bisher haben wir nur die Rushhour betrachtet. Diese zeichnet sich dadurch aus, dass wir in einem Zeitraum auf dem ganzen Weg p eine konstante Flussrate f_p haben. Eine statische Betrachtung ist somit angemessen. Zu anderen Tageszeiten ist das Modell nicht zutreffend, d.h. man muss das zeitliche Verhalten des Flusses mit modellieren. Damit erhalten wir einen Flow over time. Dieser Fluss ist dann nicht nur für Verkehr sondern auch für viele logistische Probleme relevant.

Zunächst werden wir konstante Fahrzeiten betrachten und dann später flussabhängige Fahrzeiten. Bei flussabhängigen Fahrzeiten können Ereignisse wie Stau und leere Straßen berücksichtigt werden.

5.1. Maximale $s - t$ Flüsse bei konstanten Fahrzeiten. Sei G ein Graph. Seien s und t Knoten in G , die uns einen Start- und Zielknoten angeben. Für jede Kante a gibt es eine Kapazität u_a und eine Fahrzeit τ_a , die zum durchqueren der Kante benötigt wird. In dieser Betrachtung ist τ_a konstant, also unabhängig vom Flussaufkommen auf der Kante. Insgesamt betrachten wir einen Zeithorizont von 0 bis T . Ein *dynamischer Fluss* f mit Zeithorizont T ist gegeben durch Funktionen $f_a : [0, T[\rightarrow \mathbb{R}_0^+$ für alle Kanten $a \in A$, wobei $f_a(\theta)$ die *Flussrate* darstellt, die in die Kante zum Zeitpunkt θ rein geht. Zur Zeit $\theta \geq \tau_a$ kommt $f(\theta - \tau_a)$ am Kopf der Kante an (Abb. 48).

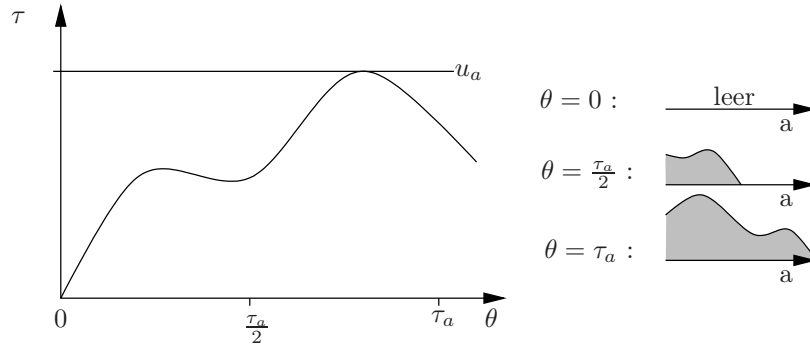


ABBILDUNG 48. Sei $f_a(\theta)$ die Flussrate der Kante a . Betrachtet man den Flussverlauf auf der Kante, so ergibt sich ein „umgekehrtes Bild“ der Flussrate.

Ein dynamischer Fluss heißt zulässig, wenn

- (1) die Kapazitätsbedingung $f_a(\theta) \leq u_a$ für alle $\theta \in [0, T[$ erfüllt ist.
- (2) die Flusserhaltungsgleichung gilt, d.h. kein Defizit an einem Knoten gestattet ist. Genauer bedeutet das, dass der Einfluss bis zum Zeitpunkt θ größer oder gleich als Ausfluss bis Zeitpunkt θ für alle θ . Formal:

$$\sum_{a \in \delta^-(v)} \int_{\tau_a}^{\theta} f_a(\xi - \tau_a) d\xi \geq \sum_{a \in \delta^+(v)} \int_0^{\theta} f_a(\xi) d\xi.$$

Ein echtes Größer kann vorkommen, wenn Warten in Knoten erlaubt („Reservoir“), die Gleichheit ist immer dann gefordert, wenn kein Warten erlaubt ist.

- (3) die Flusserhaltungsgleichung mit Gleichheit für $\theta = T$ und $f_a(\theta) = 0$ für $\theta > T - \tau_a$ erfüllt ist. Anschaulich bedeutet das, dass der Fluss das Netzwerk nach T verlassen hat.

Der Wert eines zulässigen dynamischen Flusses f ist der Nettoausfluss aus der Quelle s im gesamten Zeitraum $[0, T[$ bzw. der Nettoeinfuss in die Senke t im Zeitraum $[0, T[$. Formal erhalten wir

$$\begin{aligned} \text{value}(f) &= \sum_{a \in \delta^+(s)} \int_0^T f_a(\theta) d\theta - \sum_{a \in \delta^-(s)} \int_{\tau_a}^T f_a(\theta - \tau_a) d\theta \\ &= - \sum_{a \in \delta^+(t)} \int_0^T f_a(\theta) d\theta + \sum_{a \in \delta^-(t)} \int_{\tau_a}^T f_a(\theta - \tau_a) d\theta. \end{aligned}$$

Im Folgenden betrachten wir das *maximale dynamische $s-t$ Flussproblem*. Dabei ist ein Digraph $G = (V, A)$ mit $s, t \in V$, Kantenkapazitäten u_a und Fahrzeiten τ_a für jede Kante und ein Zeithorizont T gegeben. Gesucht ist jetzt ein zulässiger dynamischer s, t -Fluss f mit Zeithorizont T und einem maximalem Flusswert $\text{value}(f)$.

In unserer bisherigen Definition haben wir Flüsse über Flussraten f_a definiert, die über ein stetiges Zeitintervall gegeben waren: $f_a(\theta) : [0, T[\rightarrow \mathbb{R}_0^+$. Eine andere Möglichkeit wäre, die Flussraten über diskrete Zeit zu definieren. Dabei wird die Zeitachse in kleine Intervalle eingeteilt. Eine diskrete Flussrate $g_a(\theta)$ gibt zu jedem diskreten Zeitpunkt eine Flussmenge an, die über die Kante a geschickt wird (Abb. 49).

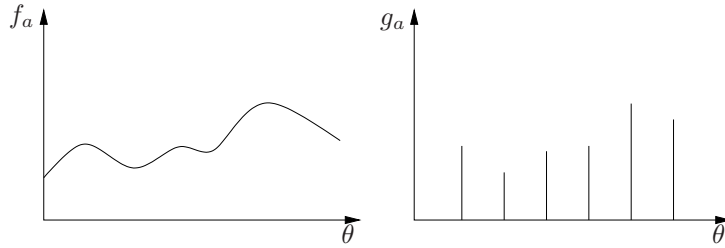


ABBILDUNG 49. Stetige und diskrete Flussraten.

Im Folgenden interessiert uns die Beziehung zwischen dem diskreten und stetigen Modell. Wir betrachten Flussraten über das Intervall $[0, T[$. Sei $g_a(\theta)$ eine diskrete Flussrate, die zu den Zeitpunkten θ_i Flusseinheiten u_i über die Kante a schickt. Eine stetige Flussrate $f_a(\theta)$, die den gleichen Flusswert wie $g_a(\theta)$ schickt und $g_a(\theta)$ gut approximiert, ist mit

$$f_a(\theta) = \frac{u_i}{\theta_{i+1} - \theta_i} \text{ für } \theta \in [\theta_i, \theta_{i+1}[$$

gegeben (Abb. 50).

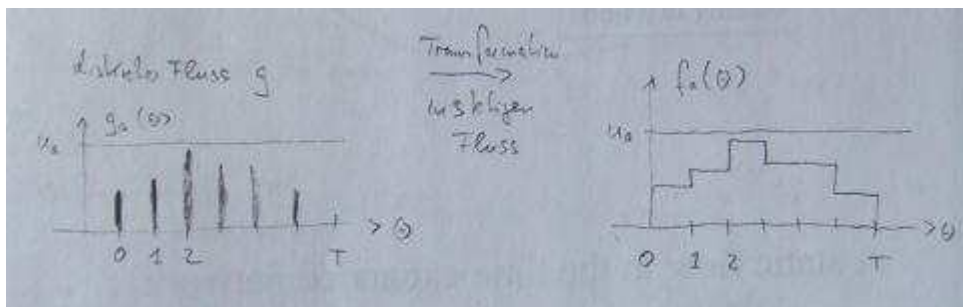


ABBILDUNG 50. Transformation einer diskreten Flussrate $g_a(\theta)$ zu einer stetigen $f_a(\theta)$.

Um eine stetige Flussrate $f_a(\theta)$ in eine diskrete umzuwandeln, müssen wir das Zeitintervall $[0, T[$ in Intervalle $[\theta_i, \theta_{i+1}[$ einteilen. Setzen wir $g_a(\tau_i) = \int_{\theta_i}^{\theta_{i+1}} f_a(\theta) d\theta$, so erhalten wir eine diskrete Flussrate über die Kante a , wobei $\text{value}(f_a(\theta)) = \text{value}(g_a(\theta))$ gilt und $g_a(\theta)$ für eine sehr feine Diskretisierung $f_a(\theta)$ sehr gut approximiert (Abb. 51).

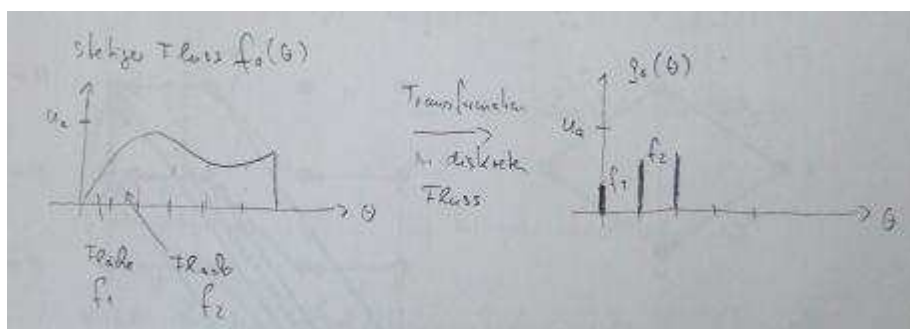


ABBILDUNG 51. Transformation einer stetigen Flussrate $f_a(\theta)$ zu einer diskreten Flussrate $g_a(\theta)$.

Note 5.1. Bei hinreichend feiner Diskretisierung der Zeit, kann jedes stetige dynamische Flussproblem als diskretes interpretiert werden und umgekehrt mit beliebig kleinem Approximationsfehler. Vorausgesetzt ist dabei, dass die f_a vernünftig z.B. stetig, sind.

Der Vorteil der diskreten Zeit besteht durch die Betrachtung zeitexpandierter Netzwerke. In einem *Zeitexpandierten Netzwerk* haben wir Zeitschichten $\theta_0 = 0, \theta_1, \dots, T$, die der diskreten Zeit entspricht. In jeder Zeitschicht haben wir eine Kopie der Knotenmenge von G . Eine Kante zwischen zwei Knoten v_{θ_i} und w_{θ_j} existiert, wenn die Kante (v, w) existiert und die Fahrzeit $\tau_{(v,w)}$ der Kante (v, w) der Zeit $\theta_j - \theta_i$ entspricht. Eine Kante zwischen den Knoten v_{θ_i} und $v_{\theta_{i+1}}$ besteht, wenn die Knoten auch Zwischenspeicher sind. Besser versteht man die Definition anhand eines Beispiels.

Example 5.2. Der zeitexpandierte Graph G^T vom Graph G mit $T = 5$ und Zeitschritten 1 hat 5 Zeitschichten, d.h. $5 \cdot n = 20$ Knoten (Abb. 52). Kanten verlaufen innerhalb von einer Zeitschicht, wenn die Zeit zum Überqueren dieser Kante Null beträgt. Vertikale Kanten existieren in diesem Fall, weil wir die Knoten auch als Zwischenspeicher nutzen können.

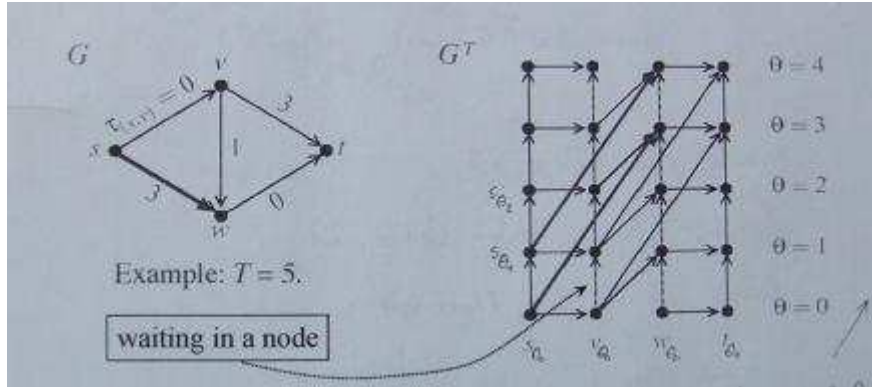


ABBILDUNG 52. Der zeitexpandierte Graph G^T von G mit $T = 5$.

Der Vorteil zeitexpandierter Netze ist, dass ein maximaler dynamischer s, t -Fluss mit Zeithorizont T einem maximalen statischen s, t -Fluss in G^T entspricht. An sich ist also die statische Flusstheorie auch für dynamische Flüsse anwendbar z.B. maxFlow - min Cut. Aber G^T wird für feine Zeitskalen riesig, weswegen polynomielle Algorithmen bzgl. G angewandt auf G^T nur pseudopolynomial sind.

Gehen wir wieder zurück zum maximalen dynamischen s, t -Flussproblem mit konstanten τ_a . Im Jahr 1958 haben Ford-Fulkerson einen polynomialen Algorithmus für dieses Problem gefunden, den wir auch später bei nicht konstanten τ_a benötigen. Im Folgenden betrachten wir die grobe Idee des Ford-Fulkerson Algorithmus (4):

Algorithm 4 Ford-Fulkerson für das maximale dynamische s, t Flussproblem

1. Schritt: Berechne einen statischen min-cost Fluss x in $G + (t, s) =: \overline{G}$ bzgl. Kosten τ_a und Kapazitäten u_a für $a \in G$ und $\tau_{(t,s)} := -T$ und $u_{(t,s)} := \infty$ für die Kante (t, s) . Dies entspricht eine Zirkulation in $\overline{G}$ zu suchen, die die Zielfunktion $\sum_{a \in G} \tau_a x_a - T \cdot \text{value}(x)$ minimiert, wobei die Balancen $b(v) = 0$ sind. Dies geht in polynomieller Zeit, mit den Methoden aus ADM I.
 2. Schritt: Wähle eine beliebige Kreisdekomposition von x bzgl. $\overline{G}$. Diese Zerlegung ist in polynomieller Zeit konstruierbar und wir erhalten maximal m Pfade. Bei allen möglichen Kreisen interessieren uns nur Kreise, die die Rückwärtskante (t, s) enthalten. Diese definieren s, t -Pfade $p \in \mathcal{P}$, wobei $\mathcal{P}$ die Menge aller Wege in G von s nach t bezeichnet.
 3. Schritt: Nutze die Pfade $p \in \mathcal{P}$ zur Konstruktion eines zeitlich wiederholten dynamischen Flusses f . Der Fluss f schickt Fluss mit konstanter Rate x_p entlang p im Intervall $[0, T - \tau_p]$. Der Zeitpunkt $T - \tau_p$ ist der letzte Zeitpunkt, zum dem Fluss von s über p nach t gelangen kann.
 4. Schritt: Dies ergibt einen dynamischen s, t -Fluss mit Zeithorizont T . Die Flussraten $f_a : [0, T[\rightarrow \mathbb{R}_0^+$ ergeben sich polynomial aus den maximal m Wegraten f_p und den Zeiten τ_a . Jede Flussrate ist eine stückweise konstante Funktion mit maximal $2m + 2$ Sprüngen. D.h. alle Flussraten haben polynomielle Outputgrößen in der Inputgröße und können in polynomialer Zeit konstruiert werden. Damit sind wir polynomial.
-

Betrachten wir den Algorithmus 4 und seine Konstruktionen an einem Beispiel.

Example 5.3. Wir wollen einen maximalen dynamischen s, t -Fluss für den Graphen G berechnen (Abb. 53).

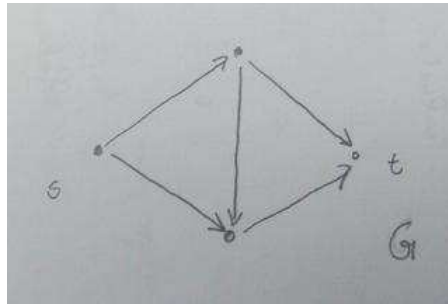


ABBILDUNG 53. Anwendung des Ford-Fulkerson Algorithmus auf den Graphen G .

1. Schritt: Berechne einen statischen min-cost Fluss x in $G + (t, s) =: \overline{G}$ bzgl. Kosten τ_a und Kapazitäten u_a für $a \in G$ und $\tau_{(t,s)} := -T$ und $u_{(t,s)} := \infty$ für die Kante (t, s) (Abb. 54).

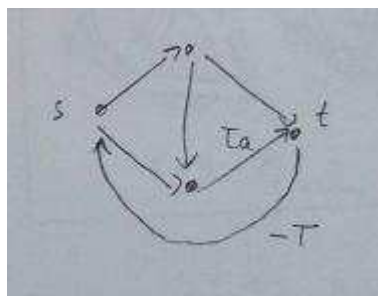


ABBILDUNG 54. Der Graph G wird im ersten Schritt zu $\overline{G}$ umgewandelt.

2. Schritt: Wähle eine beliebige Kreisdekomposition von x bzgl. $\overline{G}$ (Abb. 55).

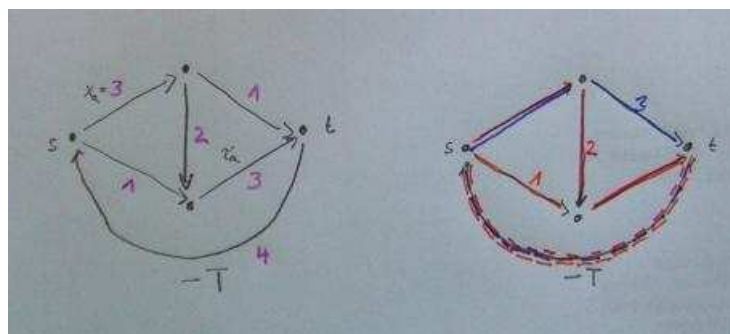


ABBILDUNG 55. Der Fluss x auf der rechten Seite kann in drei Kreise/Pfade aufgeteilt werden. Die Flussraten x_p betragen: $x_{\text{rot}} = 2$, $x_{\text{orange}} = 1$ und $x_{\text{blau}} = 1$.

3. Schritt: Nutze die Pfade $p \in \mathcal{P}$ zur Konstruktion eines zeitlich wiederholten dynamischen Flusses f (Abb. 56).

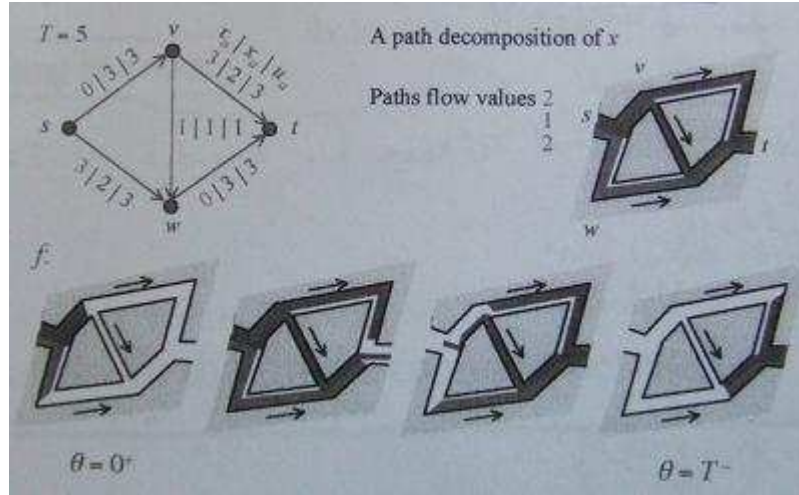


ABBILDUNG 56. Das Verhalten des dynamischen Flusses über die Zeit $[0, T]$.

4. Schritt: Die Flussraten $f_a : [0, T[\rightarrow \mathbb{R}_0^+$ ergeben sich polynomial aus den maximal m Wegraten f_p und den Zeiten τ_a (Abb. 57).

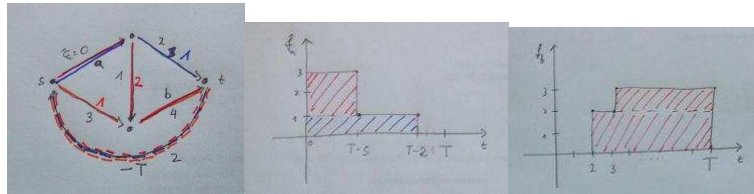


ABBILDUNG 57. Für die Kante a und die Kante b ergeben sich folgende Fahrzeitfunktionen.

Im 4. Schritt haben wir behauptet, dass der dort definierte Fluss f zulässig ist. Dies wollen wir jetzt zeigen:

Lemma 5.4. *Der Algorithmus von Ford Fulkerson konstruiert einen zulässigen zeitlich wiederholten dynamischen s, t Fluss.*

Beweis. Wir müssen die Flusserhaltung, Kapazitätsgleichung und Durchlauf des Flusses bis zum Zeitpunkt T zeigen.

- (1) *Flusserhaltung:* Fluss wird immer entlang von Wegen geschickt. Betrachten wir einen Zeitpunkt θ und einen Knoten v . Fließt über einen Weg p zum Zeitpunkt θ schon Fluss über den Knoten v , so verlässt genauso viel Fluss den Knoten in diesem Augenblick. D.h. der Weg nichts an der Flusserhaltung. Die Flusserhaltung gilt also mit Gleichheit, da wir keine Zwischenspeicherung von Fluss im Knoten haben.
- (2) *Kapazitätsgleichung:* Zu einem Zeitpunkt θ kann maximal so viel Fluss in die Kante einfließen wie über alle Wege, die diese Kante benutzen. D.h. $f_a(\theta) \leq \sum_{p \ni a} x_p = x_a \leq u_a$, wobei x_a dem Flusswert des stetigen Modells aus Schritt 2 entspricht.
- (3) Nach Zeit T ist das Netzwerk leer, da der letzte Fluss auf p zur Zeit $T - \tau_p$ abgeschickt wird.

□

Wir haben also einen speziellen zeitlich wiederholten dynamischen Fluss definiert. Es gilt $\text{value}(f) = \sum_p x_p(T - \tau_p)$, weil wir für die Zeitspanne $T - \tau_p$ bei einer Rate von x_p Fluss über den Weg p verschicken. Weiter umformen können wir dies zu

$$\begin{aligned} \text{value}(f) &= \sum_p x_p (T - \tau_p) \\ &= T \cdot \sum_p x_p - \sum_p \tau_p x_p \\ &= T \cdot \text{value}(x) - \sum_p \left(\sum_{a \in P} \tau_a \right) x_p \\ &= T \cdot \text{value}(x) - \sum_{a \in A} \left(\sum_{p \ni a} x_p \right) \tau_a \\ &= T \cdot \text{value}(x) - \sum_{a \in A} \tau_a x_a \end{aligned}$$

Im 1. Schritt hatten wir ein min-cost Flow Problem mit der Zielfunktion $\sum_{a \in A} \tau_a x_a - T \cdot \text{value}(x)$ gelöst. Da dieser Wert gleich dem negativen Wert von f ist, maximiert f den Flusswert über alle zeitlich wiederholten dynamischen s, t -Flüssen mit Zeithorizont T . Weiter wissen wir, dass $\tau_p \leq T$ für alle $p \in \mathcal{P}$ gilt.

Im Folgenden wollen wir zeigen, dass der Fluss f nicht nur optimal bzgl. aller zeitlich wiederholten dynamischen s, t -Flüssen ist, sondern auch bzgl. aller anderen dynamischen s, t -Flüsse. Dazu nutzen wir ein min-cut-max-flow Argument für den dynamischen Fall.

Definition 5.5. Ein *dynamischer s, t -Schnitt mit Zeithorizont T* ist gegeben durch Werte $\pi(v) \in [0, \infty[$ für alle Knoten $v \in V$ mit $\pi(s) = 0$ und $\pi(t) \geq T$. Diese Werte $\pi(v)$ definieren eine aufsteigende Folge $\mathcal{C}$ von Mengen

$$X(\theta) := \{v \in V | \pi(v) \leq \theta\}.$$

wobei für $\theta \in [0, T[$ die Mengen $X(\theta)$ und $V - X(\theta)$ einen s, t -Schnitt definieren.

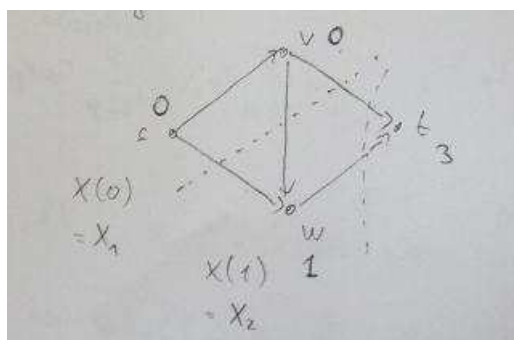


ABBILDUNG 58. Mit $\pi(s) = 0$, $\pi(v) = 0$, $\pi(w) = 1$ und $\pi(t) = 3 = T$ ergeben sich die Mengen $X(0) = \{s, v\}$ und $X(1) = \{s, v, w\}$, die dann die Schnitte $\delta(X(0)) = \{(v, t), (v, w), (s, w)\}$ und $\delta(X(1)) = \{(v, t), (w, t)\}$ definieren.

Für alle Mengen $X(\theta)$ gilt, dass $s \in X(\theta)$ und $t \notin X(\theta)$ für $\theta \in [0, T[$. Weiter stellt man fest, dass die Folge $\mathcal{C}$ nur endlich viele verschiedene Mengen X_i enthält und $X_1 \subset X_2 \subset \dots \subset X_k$ gilt.

Die *Kapazität* eines dynamischen s, t - Schnittes $\mathcal{C}$ ist definiert als

$$\text{cap}(\mathcal{C}) := \sum_{\substack{a \in A \\ a=(v,w)}} \max\{\pi(w) - \pi(v) - \tau_a, 0\} \cdot u_a.$$

Veranschaulichen kann man sich die Definition folgendermaßen: Sei $a = (v, w)$ eine Kante. Dann ist $\pi(w) - \pi(v)$ die Zeit, in der die Kante a in einem dynamischen Schnitt enthalten ist. In der Zeitspanne $\pi(w) - \pi(v) - \tau_a$ kann Fluss die Kante a überqueren, während a im Schnitt ist. Da dies negativ werden kann, wählen wir das Maximum mit 0. Maximal kann in dieser Zeit u_a , der Schranke für Flussraten, pro Zeiteinheit geschickt werden.

Lemma 5.6. Für jeden dynamischen s, t -Fluss f und jeden dynamischen s, t -Schnitt $\mathcal{C}$ zum selben Zeithorizont T gilt

$$\text{value}(f) \leq \text{cap}(\mathcal{C}).$$

Beweis. Betrachte $\mathcal{C}$ und die Folge $X_1, \dots, X_k$ der Schnitte zu $\mathcal{C}$. Jeder Schnitt X_i erzeugt die Kantenmenge $\delta^+(X_i)$, die Kanten, die aus X_i in die Menge $V - X_i$ gehen (Abb. 59).

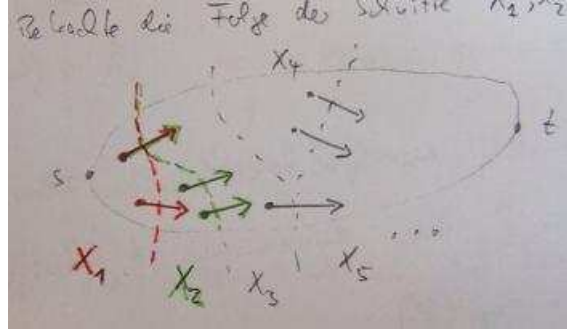


ABBILDUNG 59. Jeder Schnitt X_i definiert eine Kantenmenge $\delta^+(X_i)$.

Das Zeitintervall $[0, T[$ wird durch die „Gültigkeitsbereiche“ der Schnitte X_i überdeckt. Mit Gültigkeitsbereich von X_i ist das Intervall $[\theta_{i-1}, \theta_i[$ mit $\theta_i = \max\{\pi(v) | v \in X_i\}$ gemeint.

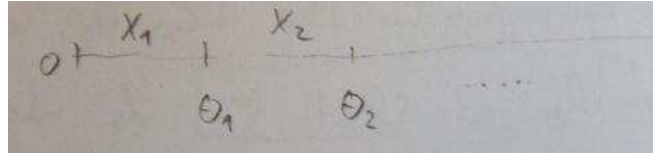


ABBILDUNG 60. Das Intervall $[0, T[$ wird durch die Gültigkeitsbereiche der Schnitte X_i überdeckt.

Damit erhalten wir für den Fluss f , dass in jedem Intervall $[\theta_{i-1}, \theta_i]$ höchstens $\sum_{a \in \delta^+(X_i)} u_a (\theta_i - \theta_{i-1} - \tau_a)$ geschickt werden kann. Es gilt also

$$\begin{aligned} \text{value}(f) &= \sum_i \text{Flussmenge über Kante in } \delta^+(X_i) \text{ in } [\theta_{i-1}, \theta_i[\\ &\leq \sum_{\substack{a \in A \\ a=(v,w)}} \max\{\pi(w) - \pi(v) - \tau_a, 0\} \cdot u_a \\ &= \text{cap}(\mathcal{C}). \end{aligned}$$

□

Im folgenden Beispiel verdeutlicht die Übereinstimmung der Überdeckung von $[0, T[$ und der Definition des Schnittes.

Example 5.7. Wir betrachten den folgenden Graphen mit dem Schnitt $\mathcal{C}$, der durch die Folge $\{X_1, X_2\}$ gegeben ist (Abb. 61).

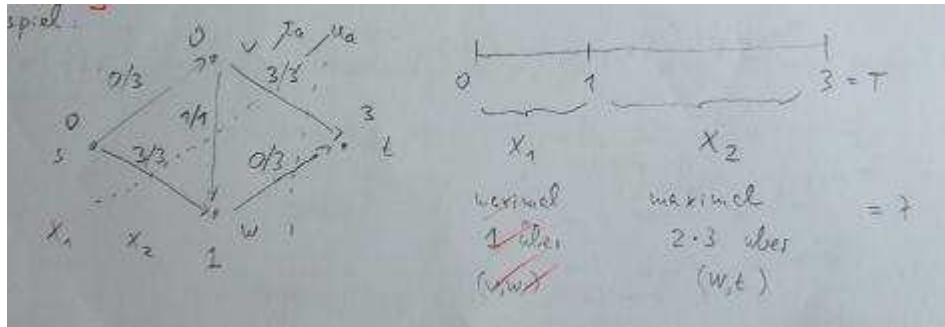


ABBILDUNG 61. Der Gültigkeitsbereich der X_i überdeckt $[0, T[$.

Wir erhalten $\theta_1 = 1$ und $\theta_2 = 3 = T$. Über die Kantenmenge $\delta^+(X_1) = \{(v, t), (v, w), (s, w)\}$ kann in der Zeit $[0, 1[$ kein Fluss fließen, weil das Überqueren der Kanten (s, w) und (v, t) drei Zeiteinheiten benötigt und für das Überqueren von (v, w) eine Zeiteinheit. Der Fluss, der über $\delta^+(X_1)$ geschickt werden kann, ist also maximal 0.

Betrachten wir nun den Fluss, der über $\delta^+(X_2) = \{(v, t), (w, t)\}$ in einem Zeitraum von $\theta_2 - \theta_1 = 2$ Zeiteinheiten fließen kann. Zum Überqueren der Kante (v, t) braucht eine Flusseinheit drei Zeiteinheiten, d.h. hier kann kein Fluss fließen. Zum Überqueren der Kante (w, t) wird keine Zeiteinheit benötigt. Es kann also zwei Zeiteinheiten die maximale Flussrate 3 fließen. Wir erhalten damit einen Fluss von $2 \cdot 3 = 6$.

Betrachten wir jetzt die Definition von $\mathcal{C}$, dann erhalten wir

Kante	$\pi(w) - \pi(v) - \tau_a$	Geschickt
(s, v)	$0 - 0 - 0$	nichts
(s, w)	$1 - 0 - 3$	nichts
(v, w)	$1 - 0 - 1$	nichts
(v, t)	$3 - 0 - 3$	nichts
(w, t)	$3 - 1 - 0$	$2 \cdot 3 = 6$

Beide Berechnungsweisen ergeben also den gleichen Wert.

Zurück zum Algorithmus von Ford-Fulkerson.

Theorem 5.8 (Ford-Fulkerson '58). *Es gelten die folgenden drei Aussagen:*

- (1) *Der Algorithmus konstruiert einen maximalen dynamischen s, t -Fluss zum Zeithorizont T .*
- (2) *Die Laufzeit wird dominiert durch die Berechnung des statischen min-cost-Flusses x .*
- (3) *Der optimale dynamische Fluss ist zeitlich wiederholt und kommt ohne Speicherung in Knoten aus.*

Beweis. Wir wollen hier nur die 1. Aussage beweisen. Wir hatten schon gezeigt, dass f ein zeitlich wiederholter dynamischer s, t -Fluss zum Zeithorizont T ist. Im

Folgenden wollen wir einen dynamischen s, t -Schnitt $\mathcal{C}$ zu f konstruieren, für den $\text{cap}(\mathcal{C}) = \text{value}(f)$ gilt.

Die Konstruktion des Schnittes $\mathcal{C}$ aus dem Fluss f ergibt sich folgendermaßen: Setze $\pi(v) := (\text{Länge eines kürzesten Weges von } s \text{ nach } v \text{ bzgl. } \tau_a \text{ im Residualgraphen } (G + (t, s))_x = \overline{G}_x)$, wobei x der min-cost Flow bzgl. τ_a für $a \in G$ und $-T$ für (t, s) ist. Diesen Fluss hatten wir im 1. Schritt berechnet. Da x optimal ist, enthält $\overline{G}_x$ keine negativen Zyklen. Also gilt $\pi(s) = 0$.

Die Kante (t, s) ist in $\overline{G}_x$, da die Kapazität unbeschränkt ist. Da keine negativen Zyklen in $\overline{G}_x$ enthalten sind, darf es keinen Weg von s nach t in G_x geben, dessen Länge kürzer als T ist. Also gilt $\pi(t) \geq T$.

Ist der Fluss $x > 0$, so ist auch die Kante $(s, t) \in \overline{G}_x$ mit $\tau_{(s,t)} = T$. Damit gilt $\pi(t) = T$. Der Fall $x = 0$ ist uninteressant. Die Werte $\pi(v)$ bilden tatsächlich einen dynamischen Schnitt.

In den folgenden zwei Claims zeigen wir, wie das Flussverhalten von f auf einer Kante $a = (v, w)$ ist, solange die Kante sich im Schnitt befindet.

Claim 5.9. Falls $[\pi(v), \pi(w) - \tau_a] \neq \emptyset$ gilt, dann ist $x_a = u_a$ und $f_a(\theta) = u_a$ für alle $\theta \in [\pi(v), \pi(w) - \tau_a]$.

Claim 5.9 sagt folgendes aus: wenn es tatsächlich Zeitpunkte gibt, zu denen Fluss von v nach w fließen kann, während die Kante a im Schnitt ist, dann ist zu diesem Zeitpunkt die Flussrate gleich der Kapazität.

Beweis Claim 5.9: 1. Schritt: z.z. $x_a = u_a$.

Wir nehmen an, dass $x_a < u_a$ gilt. Dann gilt $(v, w) \in \overline{G}_x$. Dann ist der kürzeste Weg von s nach v mit der Kante (v, w) auch ein Weg von s nach w mit der Länge $\pi(v) + \tau_a$. Da wir $\pi(v)$ als die Länge des kürzesten Weges von s nach v definiert haben, gilt $\pi(w) \leq \pi(v) + \tau_a$ und somit $\pi(v) \geq \pi(w) + \tau_a$ (Abb. 62). Damit ist das Intervall $[\pi(v), \pi(w) - \tau_a]$ leer und wir erhalten einen Widerspruch zu unserer Annahme.

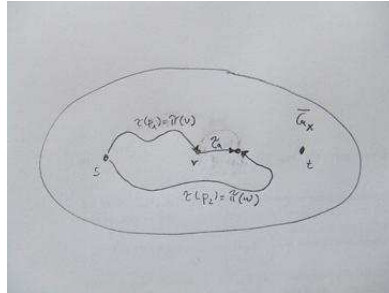


ABBILDUNG 62. Die Länge von $p_1 + (v, w)$ ist nicht kürzer als die von p_2 .

2. Schritt: z.z. $\forall \theta \in [\pi(v), \pi(w) - \tau_a]$ gilt $f_a(\theta) = u_a$.

Wir nehmen an, es existiert ein $\theta \in [\pi(v), \pi(w) - \tau_a]$ mit $f_a(\theta) < u_a = x_a$. Dann gibt es einen Weg p_0 in der Pfadzerlegung von x mit $a \in p_0$ und es gilt entweder:

- (1) der erste Fluss entlang p_0 kommt in v nach Zeit $\pi(v)$ an oder
- (2) der letzte Fluss durch p_0 verlässt w vor $\pi(w)$.

Nehmen wir an, es würde kein solcher Weg existieren. Wenn auf allen Wegen durch a der Fluss in v bereits zu $\pi(v)$ ankommt, gilt:

$$f_a(\pi(v)) = \sum_{p \ni a} x_p = x_a = u_a.$$

ABBILDUNG 64. Es gilt $\tau[(s, t) + p'_0] < \pi(w)$.

Die Kante (t, s) ist im Residualgraphen $\overline{G}_x$ enthalten, da wir annehmen, dass der Fluss x größer ist als Null. Diese Kante (t, s) hat die Länge T . Vom Knoten s schicken wir über jeden Knoten so lange Fluss, so dass er noch zum Zeitpunkt T in t ankommt. Da der letzte Fluss den Knoten w aber vor dem Zeitpunkt $\pi(w)$ verlassen hat, muss er noch eine Länger Strecke als $T - \pi(w)$ zurücklegen. Es gilt also

$$(12) \quad \tau(p_0[w, t]) > T - \pi(w).$$

Über den Weg p_0 fließt Fluss. Also existieren die Rückwärtskanten von $p_0[w, t]$ (Abb. 64). Damit haben wir einen weiteren Weg von s über den Knoten t zum Knoten w . Nach der Definition von $\pi(w)$ wissen wir $\pi(w) \leq T - \tau(p_0[w, t])$ und wir können weiter abschätzen:

$$\pi(w) \leq T - \tau(p_0[w, t]) \stackrel{(12)}{<} T - [T - \pi(w)] = \pi(w).$$

Damit erhalten wir einen Widerspruch.

Damit haben wir Claim 5.9 gezeigt. $\triangle$

Bevor wir zeigen, dass $\text{value}(f) = \text{cap}(\mathcal{C})$ gilt, brauchen wir noch ein zweites Claim.

Claim 5.10. Falls $[\pi(w) - \tau_a, \pi(v)] \neq \emptyset$ gilt, dann ist $x_a = 0$ und $f_a(\theta) = 0$ für alle $\theta \in [\pi(w) - \tau_a, \pi(v)]$.

Beweis Claim 5.10: Wir nehmen an $x_a > 0$. Dann ist die Kante (w, v) in $\overline{G}_x$ enthalten. Wir benutzen wieder das kürzeste Wege Argument aus dem ersten Teil des Beweises zu Claim 5.9. Sei p_0 ein Weg von s nach v mit $\tau(p_0) = \pi(v)$ und p_1 ein Weg von s nach w mit $\tau(p_1) = \pi(w)$. Der Weg p_0 mit der Kante (w, v) bildet dann einen Weg in $\overline{G}_x$ von s nach v . Damit gilt $\pi(v) \leq \pi(w) - \tau_a$, da die Kante (w, v) im Residualgraph die Kosten $-\tau_a$ hat. Damit wäre aber das Intervall $[\pi(w) - \tau_a, \pi(v)]$ leer. Das ist ein Widerspruch zu unserer Annahme.

Aus $x_a = 0$ folgt, dass die Kante a in keinem Weg der Pfadzerlegung von x enthalten ist. Also wird auch im dynamischen Fluss nichts über die Kante geschickt und somit ist die Flussrate $f_a(\theta) = 0$ für alle $\theta \in [\pi(w) - \tau_a, \pi(v)]$. $\triangle$

Aus Claim 5.9 und 5.10 erhalten wir folgende Situation: Für eine Kante (v, w) gilt einer der beiden Fälle:

1. Fall: $[\pi(v), \pi(w) - \tau_a]$ ist nicht leer. Dann fließt während der ganzen Zeitspanne $[\pi(v), \pi(w) - \tau_a]$ Fluss mit der Flussrate u_a aber diese Kante (Claim 5.9).

2. Fall: $[\pi(w) - \tau_a, \pi(v)]$ ist nicht leer. Dann fließt kein Fluss über diese Kante (Claim 5.10).

Betrachten wir nun den Flusswert von f :

$$\begin{aligned}
\text{value}(f) &= T \cdot \text{value}(x) - \sum_a x_a \tau_a \\
&= T \cdot \text{value}(x) - \sum_{\substack{a=(v,w) \in A \\ \pi(v) < \pi(w) - \tau_a}} u_a \tau_a \\
&= T \cdot \sum_{p \in \mathcal{P}} x_p - \sum_{\substack{a=(v,w) \in A \\ \pi(v) < \pi(w) - \tau_a}} u_a \tau_a \\
&\stackrel{(*)}{=} \sum_{p \in \mathcal{P}} x_p \cdot \left[\sum_{a=(v,w) \in p} \pi(w) - \pi(v) \right] - \sum_{\substack{a=(v,w) \in A \\ \pi(v) < \pi(w) - \tau_a}} u_a \tau_a \\
&= \sum_{a \in A} (\pi(w) - \pi(v)) \cdot \left(\sum_{\substack{p \in \mathcal{P} \\ p \ni a}} x_p \right) - \sum_{\substack{a=(v,w) \in A \\ \pi(v) < \pi(w) - \tau_a}} u_a \tau_a \\
&= \sum_{a \in A} (\pi(w) - \pi(v)) x_a - \sum_{\substack{a=(v,w) \in A \\ \pi(v) < \pi(w) - \tau_a}} u_a \tau_a \\
&= \sum_{\substack{a=(v,w) \in A \\ \pi(v) < \pi(w) - \tau_a}} (\pi(w) - \pi(v)) u_a - \sum_{\substack{a=(v,w) \in A \\ \pi(v) < \pi(w) - \tau_a}} u_a \tau_a \\
&= \sum_{a \in A} \max\{\pi(w) - \pi(v) - \tau_a, 0\} \cdot u_a \\
&= \text{cap}(\mathcal{C}).
\end{aligned}$$

Zu (*): T lässt sich durch die Teleskop Summe $T = T - 0 = \sum_{a \in \mathcal{P}} \pi(w) - \pi(v)$ darstellen.

Wir haben also einen Schnitt $\mathcal{C}$ gefunden, dessen Wert gleich dem Flusswert von f ist. Damit ist f optimal. $\square$

In diesem Abschnitt haben wir also im einfachsten Fall, eine Quelle, eine Senke und konstante Fahrzeiten, einen maximalen dynamischen Fluss in polynomialer Zeit berechnet. Der berechnete Fluss war zeitlich wiederholt. Im folgenden Abschnitt werden wir schwierigere Probleme betrachten.

5.2. Komplexität von dynamischen Flussproblemen mit konstanten Fahrzeiten τ_a . In diesem Kapitel betrachten wir die Komplexitätslandschaft für statische und dynamische Probleme (Tab. 3). Dabei geht es um die Probleme s, t -Fluss, transshipment, min-cost und quickest multi-commodity Fluss.

Zunächst erläutern wir die einzelnen Problemstellungen. Beim s, t -Fluss Problem geht es darum auf einem Netzwerk von einer Quelle zu einer Senke eine Maximale Anzahl von Fluss zu schicken, ohne die Kapazitäten auf den einzelnen Kanten zu überschreiten. Das transshipment Problem behandelt den gleichen Fall, nur dass dieses mal mehrere Quellen und Senkenpaare vorhanden sind. Im min-cost Flow Problem ist zu jedem Knoten ein Balance und auf jeder Kante Kosten gegeben. Gesucht ist jetzt ein Fluss, der die Balancen ausgleicht und dabei Kosten minimal ist. Beim quickest multi-commodity Problem gibt es mehrere Quellen- Senke Paare, zwischen denen eine bestimmte Anzahl an Fluss geroutet werden muss. Ziel beim Routen ist es einen Fluss zu finden, der die zum Routen benötigte Zeitspanne minimiert.

Im statische Fall kann das transshipment Problem als s, t -Fluss Problem modelliert werden. Dabei wird eine Super-Quelle und eine Super-Senke eingeführt. Jeder Quellknoten wird nun durch eine Kante mit der Super-Quelle verbunden. Gleiches

wird für die Senke Knoten gemacht. Insgesamt kann man das s, t -Flussproblem mit Hilfe von kürzesten Wege Berechnungen, also in polynomieller Zeit, lösen.

Im dynamischen Fall müssen die beiden Probleme gesondert betrachtet werden, da der „Trick“ der mit Super-Quelle und der Super-Senke nicht mehr funktioniert. Trotzdem können beide Probleme in polynomieller Zeit gelöst werden. Den Fall des s, t -Flusses haben wir im letzten Abschnitt betrachtet. Das dynamische transshipment Problem haben Hoppe und Tardos dann im Jahr 2000 mit Hilfe von submodularen Funktionen gelöst.

Betrachten wir noch die beiden letzten Probleme: im statischen Fall ist das min-cost Problem durch die Betrachtung des Residualgraphens und der Berechnung von kürzesten Wege in polynomieller Zeit lösbar. Im dynamischen Fall hingegen ist das Problem, wie Klinz und Woeginger 1995 gezeigt haben, NP-schwer. Allerdings kann es über zeitexpandierte Graphen in pseudopolynomieller Zeit gelöst werden. Auch das quickest multi-commodity Problem ist im dynamischen Fall NP-schwer (Fleischer und Skutella 2002/03) und im statischen Fall polynomiell. Im statischen Fall kann das Problem nur mit Hilfe von LP's gelöst werden. Ein kombinatorischer Algorithmus ist bisher noch nicht bekannt. Das dynamische Problem kann natürlich wieder über zeitexpandierte Graphen und mit LP Formulierungen in pseudopolynomieller Zeit gelöst werden. Es gibt auch Fully Polynomial Time Approximation Scheme (FPTAS) für diese Problem.

	s,t-Flow	transshipment	min-cost	quickest multi-commodity
statisch	poly		poly	poly (LP)
dynamisch	poly (statischer min-cost Fluss): Ford & Fulkerson 1958	poly (minimiere submodulare Funktionen): Hoppe & Tardos 2000	pseudopoly NP-schwer: Klinz & Woeginger 1995	pseudopoly (LP) NP-schwer: Fleischer & Skutella 2002/03

TABELLE 3. Die Komplexitätslandschaft für statische und dynamische Probleme.

5.3. $(2+\epsilon)$ -Approximation des quickest multi-commodity Flow Problems.

Im Folgenden wollen wir eine $(2+\epsilon)$ -Approximation für das quickest multi-commodity Flow Problem betrachten. Betrachten wir die Definition des Problems: Im *quickest multi-commodity Flow Problem* (QMC) ist ein Digraph G mit Quelle-Senke Paaren (s_i, t_i) und Demand d_i gegeben. Pro Kante a haben wir außerdem Kosten c_a , die beim überqueren der Kante auftreten, eine konstante Fahrzeit τ_a , die ein Fluss zum überqueren der Kante benötigt, und eine Kapazitätsbeschränkung u_a , die die maximale Belastung der Kante angibt. Weiter sei eine Kostenschranke C gegeben. Gesucht ist dann ein Fluss f , der unter allen zulässigen Flüssen, die die Kostenschranke einhalten, am wenigsten Zeit zum Routen des Demands benötigt. Die *Kosten* eines dynamischen Flusses f sind definiert als

$$C(f) := \sum_{a \in A} \int_0^T c_a \cdot f_a(\theta) d\theta.$$

Das Hauptwerkzeug, das wir betrachten werden, ist der *statische gemittelte Fluss* x zu einem dynamischen Fluss f mit Horizont T , definiert durch $x_a := \frac{1}{T} \int_0^T f_a(\theta) d\theta$. Als erstes wollen wir im folgenden Lemma überprüfen, dass x wohldefiniert ist.

Lemma 5.11. *Der statisch gemittelte Fluss x zu f erfüllt die Kapazitätsbedingung und die Flusserhaltung, d.h. er ist ein Fluss. Ferner gilt*

$$\text{value}(x) = \frac{1}{T} \text{value}(f) \text{ und } c(x) = \frac{1}{T} c(f).$$

Beweis. Sei f ein Fluss. Dann gilt $f_a(\theta) \leq u_a$ für alle $\theta \in [0, T[$. Daraus folgt

$$x_a = \frac{1}{T} \int_0^T f_a(\theta) d\theta \leq \frac{1}{T} \int_0^T u_a d\theta = u_a.$$

Also erfüllt x die Kapazitätsbedingung.

Betrachten wir nun die Flusserhaltung. Am Ende von T muss f die Flusserhaltung mit Gleichheit erfüllen. Betrachte den Knoten v . Sei a eine Kante, die in den Knoten v führt. Dann kann auf a frühestens nach τ_a Zeiteinheiten Fluss in den Knoten v fließen. Bei den Kanten, die von v weggehen, kann schon ab dem Zeitpunkt 0 Fluss fließen. Es gilt also zum Zeitpunkt T

$$(13) \quad \sum_{a \in \delta^-(v)} \int_{\tau_a}^T f(\theta - \tau_a) d\theta = \sum_{a \in \delta^+(v)} \int_0^T f_a(\theta) d\theta$$

für alle $v \in V$. Da $\frac{1}{T} x_a = \int_{\tau_a}^T f(\theta - \tau_a) d\theta = \int_0^T f_a(\theta) d\theta$ gilt, erhalten wir durch einsetzen in Gleichung (13) die Flusserhaltung von x_a .

Den Flusswert können wir analog zur Flusserhaltung berechnen. Dazu müssen wir statt einen beliebigen Knoten die Quelle s betrachten. Es gilt dann

$$\begin{aligned} \text{value}(x) &= \sum_{a \in \delta^+(s)} x_a - \sum_{a \in \delta^-(s)} x_a \\ &= \sum_{a \in \delta^+(s)} \frac{1}{T} \int_0^T f_a(\theta) d\theta - \sum_{a \in \delta^-(s)} \frac{1}{T} \int_{\tau_a}^T f(\theta - \tau_a) d\theta \\ &= \frac{1}{T} \text{value}(f) \end{aligned}$$

Betrachten wir noch die Kosten:

$$C(f) = \sum_{a \in A} \int_0^T f_a(\theta) c_a d\theta = \sum_{a \in A} c_a T x_a = TC(x).$$

□

Lemma 5.12. *Der statisch gemittelte Fluss x zu f mit Zeithorizont T ist längenbeschränkt mit Schranke T , d.h. es gibt eine Pfadzerlegung $\mathcal{P}$ für jedes (s_i, t_i) mit $\tau_p \leq T$ für $p \in \mathcal{P}$.*

Beweis. Wir gehen von einem dynamischen Fluss mit dem Zeithorizont T aus. D.h. der dynamische Fluss schafft es, allen Fluss in der Zeit T von der Quelle zur Senke zu schicken. Bei einer Pfadzerlegung von x müssen wir also nur darauf achten, auf welchen Wegen f fließt. Für jeden solchen Weg p gilt dann $\tau_p \leq T$. □

Bei der Pfadzerlegung wird nicht verlangt, dass es eine polynomiale Pfadzerlegung sein muss. Bei komplizierten f , die z.B. jede Sekunde ihren Weg ändern, kann die Pfadzerlegung auch aus exponentiell vielen Wegen bestehen.

Sei f ein dynamischer Fluss. Dann gilt für den statisch gemittelte Fluss x , dass er tatsächlich ein Fluss ist und Kosten und Flusswerte von x und f zusammenhängen. Außerdem ist x ein längenbeschränkter Fluss. An sich wollen wir aber einen dynamischen Fluss berechnen und gehen deswegen genau andersherum vor. Wir berechnen einen statisch gemittelten Fluss der längenbeschränkt ist und der bzgl.

Kostenwert und Flusswert gut ist. Danach wandeln wir diesen in einen dynamischen Fluss um. Wir können zeigen, dass wir damit eine gute Approximation eines optimalen Flusses für unser Problem erhalten.

Diese Idee haben Fleischer und Skutella 2002 in einem Algorithmus zur Konstruktion eines schnellsten dynamischen Multi-Commodity Flusses umgesetzt, der die Demands mit Kosten kleiner gleich C erfüllt. Der Algorithmus liefert eine $(2+\epsilon)$ -Approximation des schnellsten Multi-Commodity Flusses, was wir später zeigen werden (Alg. 5).

Algorithm 5 $(2 + \epsilon)$ -Approximation QMC

1. Schritt Errate den optimalen Zeithorizont T^* (binäre Suche)
2. Schritt Berechne einen T^* -längenbeschränkten statischen Multi-Commodity Fluss $(x_p)_{p \in \mathcal{P}}$ mit

$$\text{value}(x) = \frac{1}{T^*}d \text{ und } C(x) \leq \frac{1}{T^*}C.$$

Wie ein solcher Fluss berechnet werden kann, betrachten wir später.

3. Schritt Konstruiere, wie bei Ford Fulkerson, aus x einen dynamischen Fluss f , in dem entlang $p \in \mathcal{P}$ ein zeitlich wiederholter Fluss mit Raten x_p im Zeitraum $[0, T^*]$ losgeschickt wird.
 4. Schritt Warte bis aller Fluss in den Senken angekommen ist.
-

Theorem 5.13 (Fleischer und Skutella '02). *Der durch den Algorithmus konstruierte Fluss f ist ein dynamischer Fluss mit Zeithorizont $\leq 2T^*$, der alle Demands erfüllt und dessen Kosten höchstens C betragen.*

Beweis. Die Flusserhaltung und Kapazitätsbedingung ergeben sich wie bei Ford Fulkerson.

Wir müssen jetzt zeigen, dass aller Demand durch den Fluss f erfüllt ist. Dazu berechnen wir den Wert von f bzgl. einer Commodity i :

$$\text{value}_i(f) = \sum_{p \in \mathcal{P}_i} T^* x_p = T^* \text{value}_i(x) = T^* \cdot \frac{1}{T^*} d_i = d_i.$$

Weiter ergibt sich für die Kosten C von f

$$\begin{aligned} C(f) &= \sum_{p \in \mathcal{P}} c_p T^* x_p \\ &= \sum_{p \in \mathcal{P}} \left(\sum_{a \in p} c_a \right) T^* x_p \\ &= T^* \sum_{a \in A} c_a \sum_{p \ni a} x_p \\ &= T^* \sum_{a \in A} c_a x_a \\ &= T^* C(x) \\ &\leq C, \end{aligned}$$

also wird die Kostenschranke auch eingehalten.

Bleibt noch die Betrachtung des Zeithorizonts. In den ersten T^* Zeiteinheiten wird Fluss von s losgeschickt. Die Wege $p \in \mathcal{P}$ sind längenbeschränkt bzgl. T^* , d.h. der Fluss braucht maximal T^* Zeiteinheiten von der Quelle zur Senke. Damit ist das Netzwerk zum Zeitpunkt $2T^*$ leer. $\square$

Wir müssen jetzt noch die Frage aus dem 2. Schritt behandeln, wie man einen T^* -längenbeschränkten Multi-commodity Fluss konstruiert bzw. wie man auch feststellt, dass ein solcher nicht existiert. Allerdings ist dieses Problem schwach NP schwer, weil es als Spezialfall das Kosten minimale längenbeschränkten kürzeste Wege Problem enthält.

Für das CSP-Problem hatten wir ein FPTAS kennen gelernt: d.h. falls ein T -längenbeschränkter statischer Fluss existiert, dann kann man einen $T + \epsilon$ längenbeschränkten Fluss x' mit Kosten $C(x') \leq C(x)$ in polynomieller Zeit in der Inputgröße und $\frac{1}{\epsilon}$ konstruieren.

Diese FPTAS können wir dann zur Konstruktion unseres gesuchten Flusses benutzen:

Zusammengefasst erhalten wir ein FPTAS für das Ausgangsproblem indem wir zunächst das zeitexpandierte Netzwerk kondensieren, d.h. wir skalieren das zeitexpandierte Netzwerk so, dass T/Δ polynomial ist, und runden die Fahrzeiten. Danach lösen wir das statische Flussproblem im kondensierten Graphen. Dafür hatten wir ein FPTAS im letzten Kapitel kennengelernt. Die Lösung interpretieren wir als dynamischen Fluss, den wir etwas glätten um die Kapazitäten einzuhalten, mit den ursprünglichen Fahrzeiten. Im folgenden Theorem wollen wir dies Ergebnis festhalten.

Theorem 5.14. [Fleischer & Skutella 03] Wählt man $\Delta := \frac{\epsilon^2 T}{n}$, so erhält man ein kondensiertes zeitexpandiertes Netzwerk mit $\frac{n}{\epsilon^2}$ Zeitscheiben, das einen $(1 + \epsilon)$ -approximativ schnellsten Mehrgüterfluss mit beschränkten Kosten liefert.

5.4. Dynamische Fluss-Probleme mit flussabhängigen Fahrzeiten. Es ist sehr schwierig dynamische Flussprobleme mit flussabhängigen Fahrzeiten aus der Praxis mathematisch zu behandeln, da viele Problem mathematisch nicht exakt greifbar sind. In der Praxis werden deswegen Ergebnisse meist aus Simulationen gezogen. Diese Simulationen sind schon sehr ausgereift, u.a. kann ausgefeiltes Fahrverhalten abgebildet werden. Eine weiterer Grund für die Schwierigkeit mit flussabhängigen Fahrzeiten ist, dass die Technik der zeitexpandierten Graphen nicht zur Verfügung steht. Die zeitexpandierten Graphen G^T basieren auf festen Fahrzeiten. Im dynamischen Fall würde sich G^T andauernd ändern. Allgemein stellt sich die Frage, ob einfache Optimierungsmodelle nicht in die Simulation einzubringen sind.

In der Arbeitsgruppe COGA wurden in den letzten Jahren zwei vereinfachte Modell behandelt:

- (1) Einfluss abhängige Fahrzeiten [Köhler, Langkan, Skutella]
- (2) Last abhängige Fahrzeiten [Köhler, Skutella]

Das Modell der Einfluss abhängigen Fahrzeiten geht von der folgenden Vereinfachung aus: Sei $x_a(\theta)$ die Flussrate, die zum Zeitpunkt θ in die Kante a rein fließt. Dann benötigt der Fluss, der bei θ die Kante betritt, $\tau_a(x_a(\theta))$ Zeit zum durchfahren der Kante. Damit haben alle Flusseinheiten, die zum Zeitpunkt θ die Kante a betreten die gleiche Fahrzeit.

Ein Nachteil des Problems ist, dass die FIFO-Eigenschaft verletzt wird (Abb. 65).

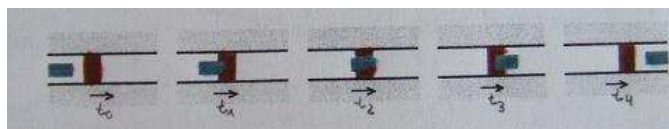


ABBILDUNG 65. Der blaue Block überholt den roten auf der Kante, obwohl der Eintrittszeitpunkt des roten Blocks vor dem des blauen lag.

Betrachten wir eine Straße im Querschnitt und zu unterschiedlichen Zeitpunkten. Nehmen wir im ersten Bild an, dass wir einen riesigen Flussblock (rot) haben, der sehr viel Zeit braucht, um durch die Kante zu kommen. Nehmen wir an, danach kommt ein kleinerer (blauer) Flussblock, der die Kante schneller durchquert. Irgendwann wird dann der blaue Block den roten auf der Kante überholen. In der Praxis kann man diese Verhalten normalerweise nicht feststellen.

Im zweiten Modell werden die Fahrzeiten auf einer Kante Last abhängig. Dabei gibt $y_a(\theta)$ die Flussmenge (keine Rate) auf der Kante a zum Zeitpunkt θ . Die Flusseinheiten auf a zum Zeitpunkt θ fahren dann mit der Geschwindigkeit $\frac{1}{\tau_a(y_a(\theta))}$. D.h. zu jedem Zeitpunkt θ haben alle Flusseinheiten auf a dieselbe Geschwindigkeit, die sich jedoch mit θ ändert.

Der Nachteil an dem Modell ist, dass später einfahrender Fluss die Geschwindigkeit des Flusses vor ihm auf der Kante bestimmt. (Abb. 66)

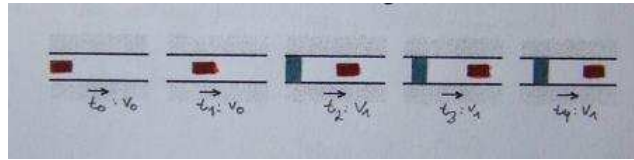


ABBILDUNG 66. Bevor im Zeitpunkt t_2 der blaue Block die Kante betritt, kann der rote Block mit der Geschwindigkeit $v_0 > v_1$ fahren.

Trotz dieser Nachteile kann in den einzelnen Modellen optimiert werden. Es zeigt sich, dass die dabei ermittelten Routen denen der Simulation überlegen sind.

Die Ergebnisse der beiden Arbeiten in einer Übersicht (Tab. 4).

Einfluss abhängige Fahrzeiten:	Last-abhängige Fahrzeiten:
schnellste s, t Fluss Problem ist stark NP-schwer	schnellste s, t Fluss Problem ist APX-schwer d.h. wir haben kein approximations Schema (außer $P = NP$)
schnellste s, t Fluss Problem hat eine einfache $(2 + \epsilon)$ Approximation mit zeitlich wiederholten Flüssen	schnellste s, t Fluss Problem hat eine $(2 + \epsilon)$ Approximation mit zeitlich wiederholten Flüssen
die Approximation ist zu einem FPTAS mit unbrauchbaren Flüssen ausbaubar	

TABELLE 4. Die Ergebnisse für das schnellste s, t - Flussproblem aus den beiden Arbeiten.

Beide Ansätze wurden implementiert und sind auch in der Praxis brauchbar.

Im Folgenden wollen wir die Konstruktion eines schnellen dynamischen s, t Flusses f bei Einfluss-abhängigen Fahrzeiten betrachten, der uns eine $(2 + \epsilon)$ Approximation für schnellste dynamische s, t Flüsse mit Einfluss-abhängigen Fahrzeiten liefert. Dazu unterteilen wir das Problem in drei Fälle.

Fall A: die Fahrzeiten sind konstant.

Nutze binäre Suche bzgl. Zeithorizont T und löse für jedes T ein maximales dynamisches s, t - Flussproblem gemäß 5.1. Wenn der Zeithorizont ganzzahlig ist, dann endet die binäre Suche. Ist das nicht der Fall erhalten wir eine ϵ - Approximation.

Allerdings konnten Fleischer und Tardas 1998 zeigen: Wenn die Fahrzeiten τ_a und der Demand d ganzzahlig sind, dann ist der optimale Zeithorizont eine rationale Zahl $\frac{p}{q}$ mit $q \leq \text{Kapazität eines minimalen } s, t\text{-Schnittes}$. Also ist eine exakte binäre Suche in polynomieller Zeit möglich. Dieses fasst das nachfolgende Theorem zusammen.

Theorem 5.15. *Für das schnellste s, t Fluss-Problem mit konstanten Fahrzeiten existiert ein optimaler Fluss, der zeitlich wiederholt ist. Ein solcher Fluss kann in polynomieller Zeit ermittelt werden.*

Damit hätten wir den ersten der drei Fälle abgeschlossen. Gehen wir jetzt zu Fall B.

Fall B: die Fahrzeitfunktionen sind stückweise konstant und monoton steigend (Abb. 67).

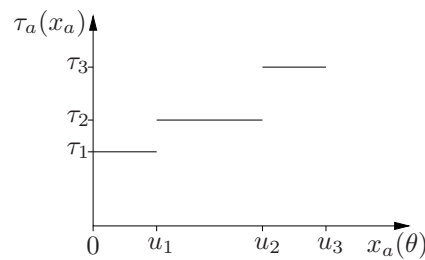


ABBILDUNG 67. Im Fall B werden nur stückweise konstante, monoton steigende Fahrzeiten behandelt.

Betrachten wir nun einen Algorithmus, der uns eine 2er Approximation des Problems liefert.

Algorithm 6 2er Approximation für Fall B

1. Schritt: Konstruiere für jede Kante a aus der Funktion $\tau_a(x_a)$ einen so genannten Bogengraphen G_a^B .
2. Schritt: Ersetze jede Kante a durch ihren Bogengraphen G_a^B und ermittle im resultierenden Graphen G^B einen schnellsten s, t -Fluss f^B zum Demand d . Das können wir wie im Fall A machen, da jetzt alle Fahrzeiten konstant sind. Damit erhalten wir sofort, dass dieser Schritt polynomial in Abhängigkeit der Inputgröße ist.

Für den Zeithorizont T^B des Flusses f^B ergibt sich die Relation $T^B \leq T^{\text{opt}}$, wobei T^{opt} der Zeithorizont des optimalen Flusses im Ursprungsproblem ist. Dies gilt, da jeder Fluss aus G auch in G^B zulässig ist. Allerdings ist nicht jeder Fluss aus G^B auch zulässig in G , da in G^B Aufspaltung erlaubt ist.

Weiter wissen wir, dass beim Fluss f^B alle Kanten von unten nach oben gesättigt sind. Wäre dies nicht der Fall, könnte man den Fluss nach unten verschieben und somit die Fahrzeit verbessern.

Als letztes gilt noch, dass f^B zeitlich wiederholt ist und aus einem statischen Fluss x^B in G^B gewonnen wurde.

Leider ist f^B im Allgemeinen nicht zulässig in G .

3. Schritt: Konstruktion eines in G zulässigen zeitlich wiederholten Flusses f aus f^B .

- reroute x^B , so dass er in keinem G_a^B mehrere Bögen benutzt. Dazu schiebe allen Fluss in G_a^B auf den höchsten Fluss-führenden Bogen. Sei x der resultierende statische Fluss. Dieser ist als Fluss in G interpretierbar, da er nur einen Bogen pro Kante benutzt. Der Fluss x ist trivialerweise ein statischer Fluss in G . Da x nur Wege nutzt, die auch von x^B benutzt wurden, gilt $\tau_p \leq T^B$ für $p \in \mathcal{P}$ mit $x_p > 0$.
- Nutze diese Wege $p \in \mathcal{P}$ (in G interpretiert) zur Konstruktion eines zeitlich wiederholten Flusses f : Sende zeitlich wiederholt entlang der Wege $p \in \mathcal{P}$. Bis zum Zeithorizont T' , der so gewählt wird, dass aller Demand d bei t zum Zeitpunkt T' angekommen ist. Es gilt also

$$\begin{aligned} d = \text{value}(f) &= \sum_{p \in \mathcal{P}} (T' - \tau_p) x_p \\ &= \text{value}(x) T' - \sum_{a \in A} \tau_a x_a. \end{aligned}$$

Damit ist T' durch $T' = \frac{d + \sum \tau_a x_a}{\text{value}(x)}$ leicht ermittelbar.

Betrachten wir an einem Beispiel die Konstruktionen des Bogengraphen und des Rerouten von x^B .

Example 5.16. Im 1. Schritt wird für jede Kante a ein Bogengraph G_a^B konstruiert. Dies kann man sich durch die Abb. 68 veranschaulichen.

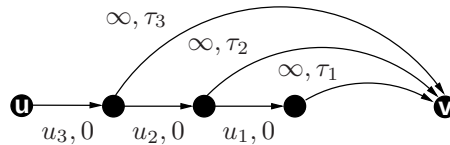


ABBILDUNG 68. Bogengraph G_a^B der Kante a mit der Fahrzeitfunktion aus Abb. 67.

Im 3. Schritt wird dann der Fluss x^B so umgeleitet, dass er in keinem G_a^B mehrere Bögen benutzt. Dazu schiebe allen Fluss in G_a^B auf den höchsten Fluss-führenden Bogen (Abb. 69).

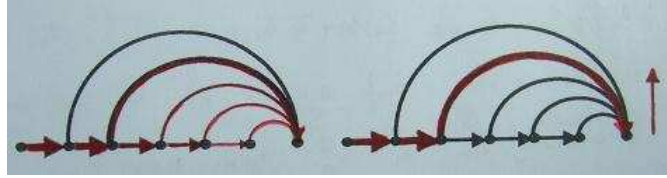


ABBILDUNG 69. Der Fluss x^B wird umgeroutet zu einem zulässigen Fluss x in G .

Im nächsten Satz wollen wir zeigen, dass die Konstruktion von f uns eine $(2 + \epsilon)$ Approximation an den schnellsten dynamischen s, t Fluss liefert.

Theorem 5.17. *Der Fluss f löst das schnellste dynamische s, t - Fluss-Problem mit stückweise konstanten Fahrzeiten in der Zeit $T' \leq 2T^{\text{opt}}$. Dabei ist f zeitlich wiederholt und polynomial ermittelbar.*

Beweis. Wir definieren $\text{value}(T)$ als den Wert von f bei Zeithorizont T . Es gilt also

$$\text{value}(T) := T \text{value}(x) - \sum_{a \in A} \tau_a x_a.$$

Diese Funktion ist monoton steigend.

Um zu zeigen, dass f eine $2 + \epsilon$ Approximation darstellt, reicht es zu zeigen, dass $\text{value}(2T^{\text{opt}}) \geq d$ gilt. D.h. in der Zeit $2T^{\text{opt}}$ wird der gesamte Demand vom Fluss f durch das Netzwerk geroutet. Betrachten wir also $\text{value}(2T^{\text{opt}})$:

$$\begin{aligned} \text{value}(2T^{\text{opt}}) &\geq \text{value}(2T^B) && [\text{da } \text{value}(T) \text{ monoton in } T \text{ ist}] \\ &= 2T^B \text{value}(x) - \sum_{a \in A^B} \tau_a x_a \\ &= T^B \text{value}(x) + T^B \sum_{p \in \mathcal{P}^B} x_p - \underbrace{\sum_{a \in A} \tau_a x_a}_{= \sum_{p \in \mathcal{P}^B} \tau_p x_p} \\ &= T^B \text{value}(x) + \sum_{p \in \mathcal{P}^B} \underbrace{(T^B - \tau_p)}_{\geq 0} x_p \\ &\geq T^B \text{value}(x) \\ &= T^B \text{value}(x^B) \\ &\geq T^B \text{value}(x^B) - \sum_{a \in A} \tau_a x_a^B \\ &= \text{value}(f) = d \end{aligned}$$

Damit hätten wir die Behauptung gezeigt. $\square$

Bleibt noch der letzte Fall C.

Fall C: die Fahrzeitfunktionen sind monoton und linksseitig stetig.

Wir approximieren die Fahrzeitfunktion durch stückweise konstante Funktionen. Bei der Approximation unterteilen wir die y Achse in kleine Intervalle $[a_i, a_{i+1}]$ mit $a_i = (1 + \eta)^i \delta$, die also geometrisch wachsen. (Abb. 70)

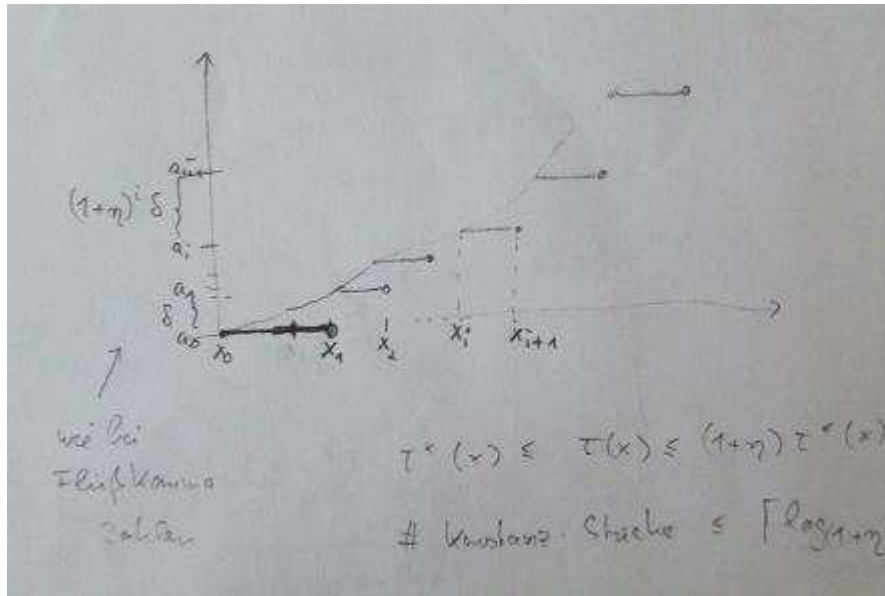


ABBILDUNG 70. Approximation von τ durch τ^* .

Danach betrachten wir den Schnittpunkt (a_i, x_i) der Gerade a_i , die parallel zur x -Achse verläuft, mit der Fahrzeitfunktion τ . Auf der x -Achse erhalten wir dann ebenfalls eine Unterteilung in Intervalle $[x_i, x_{i+1}]$. Die Approximation $\tau^*(x)$ an τ hat dann die Form $\tau^*(x) = a_i$ für $x \in [x_i, x_{i+1}]$. Weiter gilt

$$\tau^* \leq \tau(x) \leq (1+\eta)\tau^*(x) + \delta \quad \forall x,$$

d.h. die Anzahl der Konstanz-Stücke $\leq \left\lceil \log_{(1+\eta)} \frac{\tau_a(u_a)}{\delta} + 1 \right\rceil$. Wenden wir die Konstruktion aus Fall B auf τ^* an, dann erhält man nach einigen Umformungen eine $2 + \epsilon$ Approximation.