Universität Karlsruhe
Fakultät für Informatik
ITI Wagner

Algorithmen für Routenplanung
Sommersemester 2009
Daniel Delling, Thomas Pajor

Fünftes Übungsblatt - Musterlösung

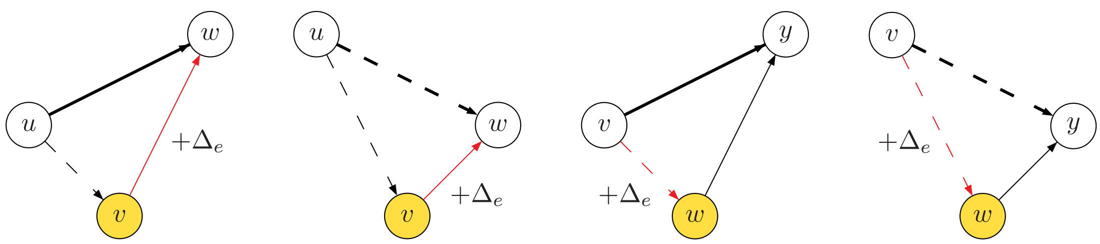
Aufgabe 1: Aufgabe 1: Dynamische Kontraktionshierarchien

Sei $G_\uparrow = (V, E_\uparrow, \mathrm{len}_\uparrow)$ der Upwardgraph und $G_\downarrow = (V, E_\downarrow, \mathrm{len}_\downarrow)$ der Downwardgraph einer Kontraktionshierarchie. Weiterhin sei $E_\Delta \subseteq E_\uparrow$ die Menge von Kanten für die sich das Gewicht von $\mathrm{len}(e)$ auf $\mathrm{len}(e) + \Delta_e$ für $\Delta_e > 0$ erhöht (zum Beispiel durch Staus oder Straßensperrungen).

(a) Sei $e \in E_\Delta$ eine Kante für die sich das Gewicht erhöht. Ein Kante e' ist in $\mathrm{SC}(e) \subseteq E_\uparrow$ enthalten, wenn sie durch die Gewichtsänderung beeinflusst wird, das heißt, wenn e Teil des Pfades $P_{e'}$ ist, der durch e' überbrückt wird.

Middle-Nodes. Zunächst wird während der Knotenreduktion eines Knotes v für eingefügte Shortcuts (u, w), die v überbrücken, der überbrückte Knoten (*middle-node*) v mit abgespeichert. Sei also $\mathrm{mid} : E \to V$ die Funktion, die einem Shortcut $e \in E$ den jeweils überbrückten Knoten $v \in V$ zuordnet. Dies lässt sich während der Vorberechnung der Kontraktionshierarchie ohne Mehraufwand durchführen.

Berechnen von $\mathrm{SC}(e)$**.** Mit folgendem rekursiven Verfahren lassen sich nun zu einer Kante $e \in E_\Delta$ die relevanten Shortcuts $\mathrm{SC}(e)$ bestimmen. Zunächst enthält $\mathrm{SC}(e)$ nur die Kante e. Um Shortcuts (u, v, w) zu identifizieren[1], wird für alle Kanten $(u, x) \in E_\uparrow$ überprüft ob $\mathrm{mid}(u, x) = v$. Ist dies der Fall, so wird (u, x) in $\mathrm{SC}(E)$ eingefügt, und das gleiche Verfahren für die Kante (u, x) rekursiv angewendet. Um hingegen Shortcuts (v, w, y) zu identifizieren, wird für alle Kanten $(x, y) \in E_\downarrow$ geschaut ob $\mathrm{mid}(x, y) = w$. Ist dies der Fall, so wird (x, y) zu $\mathrm{SC}(E)$ eingefügt, und das Verfahren für (x, y) rekursiv angewendet.



(a) (b) (c) (d)

Abbildung 1: Vier Möglichkeiten wie eine Update-Kante (rot) zu einem unmittelbaren Shortcut (dick) in Bezeihung stehen kann. Bezüglich der Kontraktionshierarchie gelte $u < v$ gdw. u unterhalb von v gezeichnet ist. Kanten aus $E_\uparrow$ sind durchgezogen, Kanten aus $E_\downarrow$ gestrichelt. Eine Aufwärtskante kann Einfluss auf eine Abwärtskante (Abbildung b) nehmen, die dann später wieder Einfluss auf eine Aufwärtskante nehmen kann (Abbildung c), daher müssen Abwärtskanten aus $E_\downarrow$ bei der Rekursion mitberücksichtigt werden.

[1] Die Notation (u, v, w) stehe für eine Shortcut-Kante (u, w), die den Knoten v überbrückt.

Auf diese Art werden alle Shortcuts identifiziert (auch solche mit mehreren Hops), die die Kante e überspringen.

(b) Sei zu $e \in E_\Delta$ die Menge $\mathrm{SC}(e)$ aus Aufgabe (a) gegeben. Die Idee besteht nun darin eine Teilmenge $U \subset V$ von Knoten zu bestimmen, für die die Kontraktion erneut durchgeführt wird. Dadurch werden die invaliden Shortcuts aus $\mathrm{SC}(e)$ neu berechnet und gleichzeitig eventuell neue Shortcuts eingefügt.

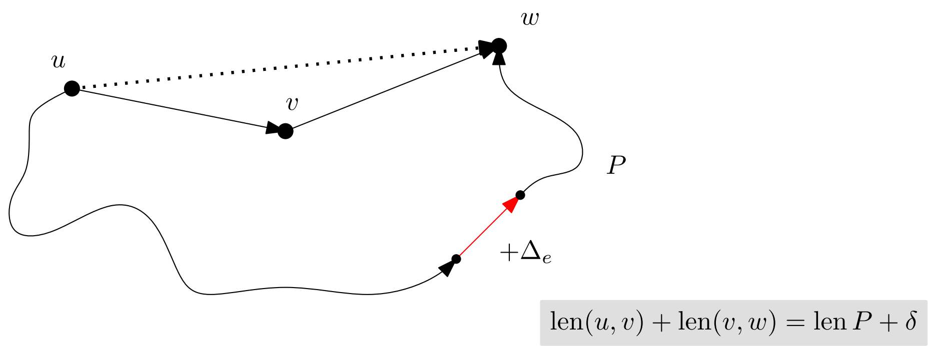
Das Vorgehen ist dabei wie folgt. Zunächst werden auf allen Kanten $e' \in \mathrm{SC}(e)$ die Gewichte $\mathrm{len}(e')$ um Δ_e erhöht. Da dies jedoch zur Konsequenz haben kann, dass die Kontraktionshierarchie nicht mehr konsistent ist, müssen wir Teile der Kontraktionshierarchie neu berechnen.

Dazu werden in einem ersten Schritt eine Menge $S \subset V$ von *Saat-Knoten* (Seed-Nodes) identifiziert, deren Elemente unmittelbar neu kontrahiert werden müssen. Da die Kontraktion von Knoten $v \in S$ Einfluss auf die Kontraktion von Knoten $w \in V$ mit $w > v$ hat, müssen alle *erreichbaren* Knoten $w \in v$ (Es muss einen Pfad von v nach w in $G_\uparrow$ geben) mit $w > v$ ebenfalls neu kontrahiert werden. Es wird also in einem zweiten Schritt zu gegebenem S eine Update-Obermenge U mit $S \subseteq U \subseteq V$ durch eine Breitensuche in $G_\uparrow$ berechnet, die die erreichbaren Knoten von S enthält. Die Kontraktion wird dann auf dem durch U induzierten Teilgraphen in der durch die Knotenordnung gegebenen Reihenfolge neu durchgeführt.

Berechnen der Saat-Knoten. Um die Menge S von Saat-Knoten zu berechnen unterscheiden wir folgende zwei Fälle.

1. Eine Kante $e = (u, v) \in E_\uparrow$ wird *eingefügt*, oder das Gewicht einer Kante *verringert* sich. Dann muss der Knoten u neu kontrahiert werden. Somit ist also u ein Saat-Knoten.
2. Das Gewicht einer Kante e wird *erhöht*. Dies ist der Fall für alle $e' \in \mathrm{SC}(e)$. Ist solch eine Kante Teil eines Pfades bei der Zeugensuche für die Kantenreduktion eines Shortcuts (u, v, w) gewesen, so ist unter Umständen der Shortcut (u, v, w) wieder einzufügen, da der Pfad über e nun möglicherweise nicht mehr kürzer als (u, v, w) ist. In diesem Fall ist also v ebenfalls ein Saat-Knoten.

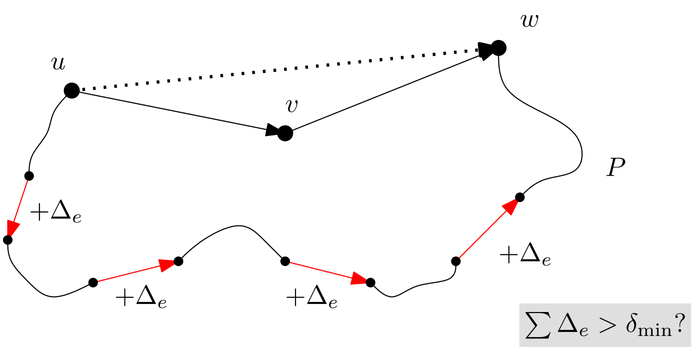
Um effizient zu einer Kante e zu bestimmen, ob sie Teil bei der Zeugensuche für die Kantenreduktion war, kann folgendermaßen vorgegangen werden. Wir speichern eine Relation $A \subseteq E \times V \times \mathbb{N}$ wobei $(e, v, \delta) \in A$ genau dann wenn es einen Shortcut (u, v, w) gibt mit $\mathrm{len}(u, v, w) = \mathrm{len}\, P + \delta$ wobei e Teil des Weges P ist. Das heißt, der Shortcut (u, v, w) wurde bei der Kantenreduktion mittels des Zeugen P gelöscht, da P um δ kürzer war. Das heißt, P darf maximal um δ länger werden um weiterhin ein Zeuge für den Shortcut (u, v, w) zu bleiben. Die Relation A kann während der Berechnung der Kontraktionshierarchie konstruiert werden.



Sei nun e eine erhöhende Kante und A_e die Teilrelation von A auf e eingeschränkt. Wir gehen für jeden Knoten v, der in A_e vorkommt folgendermaßen vor, sei v fest. Sei $\delta_{\min}$ das minimale δ unter *allen* Tupeln in A, die v enthalten. Dies ist also die kleinste Längendifferenz die dafür verantwortlich war, dass ein Shortcut, der v überbrückt gelöscht werden durfte. Dem stellen wir nun die Kantengewichtserhöhung gegenüber. Sei also

$$\sigma := \sum_{\substack{(e,v,\delta)\in A,\\ e\in E_\Delta}} \Delta_e.$$

Gilt $\sigma > \delta_{\min}$, so ist der Knoten v also möglicherweise neu zu kontrahieren, und wird zu der Menge von Saat-Knoten hinzugefügt. Die folgende Abbildung illustriert den Zusammenhang nochmal. Die roten Kanten sind dabei Kanten von Tupeln aus A, die v enthalten. Es sei angemerkt, dass mit dieser Methode auch Kanten aufsummiert werden die nicht auf dem Pfad P liegen, somit ist $\sigma > \delta_{\min}$ eine konservative Abschätzung.



(c) Zu einem Update E_Δ werden zunächst nur die Mengen $\mathrm{SC}(e)$ von invaliden Shortcuts für jedes $e \in E_\Delta$ berechnet. Es werden jedoch *keinerlei* Updates durchgeführt (Weder werden die Kantengewichte auf den Shortcuts geändert, noch werden Teiles des Graphen neu kontrahiert).

Da für alle $e \in E_\Delta$ gilt dass $\Delta_e > 0$, sind kürzeste s-t-Wege P weiterhin valide, solange sie keine Kante $e \in E_\Delta$ enthalten. Sei deshalb P ein kürzester Weg und $\mathrm{SC}(P)$ die Menge von invaliden Kanten entlang des kürzesten Weges. Dann wird iterativ ausschließlich die Teilmenge $\mathrm{SC}(P)$ upgedatet – also gemäß Aufgabe (b) die Gewichte der Kanten $e \in \mathrm{SC}(P)$ um Δ_e erhöht, sowie die relevanten Teile des Graphen rekontrahiert und schließlich $\mathrm{SC}(P)$ aus der Menge der invaliden Shortcuts gelöscht – und die s-t-Anfrage wiederholt bis der gefundene Weg P keine invaliden Shortcuts mehr enthält.

Dadurch verlagert sich der Aufwand für die Aktualisierung der Kontraktionshierarchie zur Query-Zeit. In der Praxis terminiert jedoch das Verfahren bereits nach sehr wenigen Iterationen, wodurch der Zeitverlust während der Query nicht allzu drastisch auswirkt.

Aufgabe 2: Aufgabe 2: Gemischte Schienenfunktionen

Gegeben sei der Funktionenraum der gemischten Schienenfunktionen $\hat{\mathbb{F}}$ wie auf dem Übungsblatt definiert.

(a) Abbildung 2 zeigt die Schienenfunktion mit den Interpolationspunkten wie auf dem Übungsblatt definiert. Für jeden Zug wird ein Interpolationspunkt (τ_i, w_i) in f gesetzt (angedeutet durch die Punkte in der Abbildung), dessen Abfahtszeit τ_i dem Wert auf der “x”-Achse entspricht. Der Wert $f(\tau_i)$ entspricht der reinen Fahrzeit des Zuges entlang der Kante. Die Interpolation von f zwischen zwei Punkten (t_i, w_i) und (t_j, w_j) erfolgt durch $f(\tau) = t_j - t_i + f(w_j)$

und entspricht der Wartezeit bis zur Abfahrt des nächsten Zuges j plus der reinen Fahrzeit $f(w_j)$ des nächsten Zuges.

Beachten Sie, dass wegen der Periodizität auch zwischen dem *letzten* und dem *ersten* Interpolationspunkt interpoliert werden muss.

(b) Sei $f \in \mathbb{F}$ eine reguläre Schienenfunktion, und $c \in \mathbb{R}^+$ eine Konstante.

Im Allgemeinen ist die Link-Operation für zwei beliebige Funktionen f und g (diese müssen nicht notwendigerweise stückweise linear sein) definiert durch

$$f \oplus g := f + g \circ (\mathrm{id} + f) \quad \text{bzw.} \quad (f \oplus g)(\tau) := f(\tau) + g(\tau + f(\tau)). \tag{1}$$

Interpretiert man f und g als Reisezeit-Funktionen, so bedeutet dies, dass die Reiezeit von $f \oplus g$ die konsekutive Reisezeit entlang der Funktion f und g ist, wobei der Abfahrtszeitpunkt an g dem Ankunftszeitpunkt nach f entspricht.

- Bezüglich $g := c \oplus f$ ergibt sich mit Gleichung (1) da c konstant ist

$$\begin{aligned}(c \oplus f)(\tau) &= c(\tau) + f(\tau + c(\tau)) \\ &= c + f(\tau + c).\end{aligned}$$

  Als Konsequenz lässt sich ableiten, dass für alle Interpolationspunkte in $(t_i^g, w_i^g) \in I^g$ bezüglich der Gewichte $w_i^g = w_i^f + c$ gilt, da c für alle τ auf $f(\tau + c)$ addiert wird. Bezüglich der Abfahrtszeiten t_i^g ergibt sich $t_i^g = t_i^f - c$, denn bezüglich der Abfahrtszeit des Zuges i in $c \oplus f$ muss beachtet werden, dass die konstante Reisezeit vor f *vorher* abgeleistet werden muss.
- Analog ergibt sich für $g' := f \oplus c$ aus Gleichung (1) dass

$$\begin{aligned}(f \oplus c)(\tau) &= f(\tau) + c(\tau + f(\tau)) \\ &= f(\tau) + c.\end{aligned}$$

  Damit gilt für die Interpolationspunkte $(t_i^{g'}, w_i^{g'}) \in I^{g'}$ dass $w_i^{g'} = w_i^f + c$. Die Abfahrtszeiten bleiben unverändert, das heißt $t_i^{g'} = t_i^f$, da die konstante Reisezeit erst auf die Schienenfunktion folgt. Das konstante Gewicht wird lediglich an die Zugfahrt "angehängt", und muss daher auf das jeweilige Gewicht w_i^f addiert werden.

Damit folgt insbesondere $c \oplus f \neq f \oplus c$, das heißt, die Link-Operation ist nicht symmetrisch.

□

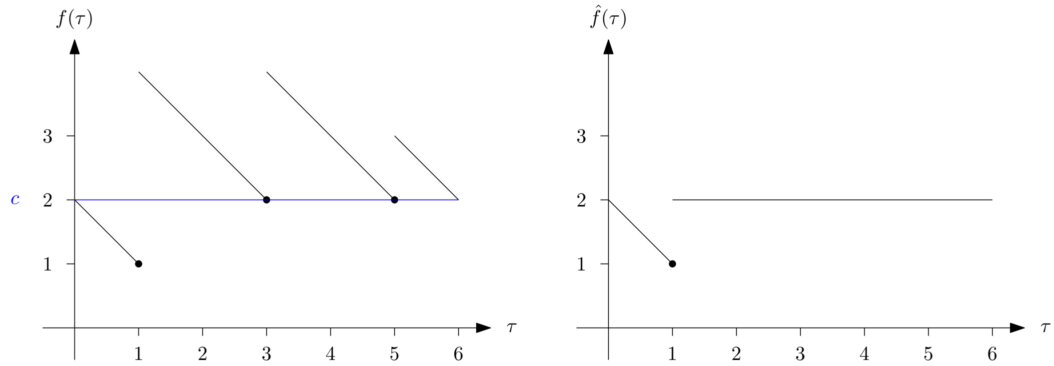


Abbildung 2: Die gemischte Schienenfunktion aus Aufgabe 2 (a).

(c) Seien $\hat{f}, \hat{g} \in \hat{\mathbb{F}}$ zwei gemischte Schienenfunktionen. Wir bezeichnen $\hat{f}$ durch $\hat{f} = (f, c)$ und $\hat{g}$ durch $\hat{g} = (g, d)$. Eine gemischte Schienenfunktion kann in einem Transportnetz entstehen wenn bei Profil-Anfragen die Minimumsbildung zwischen einer Konstanten $c \in \mathbb{R}^+$ und einer regulären Schienenfunktion $f \in \mathbb{F}$ durchgeführt wird.

Das heißt, eine gemischte Schienenfunktion $\hat{f} := (f, c) \in \hat{\mathbb{F}}$ wird für einen Zeitpunkt τ ausgewertet durch $\hat{f}(\tau) = \min(f(\tau), c)$, bzw. in Funktionalschreibweise $\hat{f} = \min(f, c)$.

Um nun die Abgeschlossenheit von $\hat{\mathbb{F}}$ bezüglich min und $\oplus$ zu zeigen, zeigen wir, dass $\min(\hat{f}, \hat{g}) \in \hat{\mathbb{F}}$ und $\hat{f} \oplus \hat{g} \in \hat{\mathbb{F}}$.

- *Minimumsbildung.*
  Aus der Distributivität von min folgt für zwei gemischte Schienenfunktionen $\hat{f}$ und $\hat{g}$ bezüglich der min-Operation in funktionaler Schreibweise

$$\begin{aligned}\min(\hat{f}, \hat{g}) &= \min\big(\min(f, c), \min(g, d)\big)\\ &= \min\big(f, g, c, d\big)\\ &= \min\big(\min(f, g), \min(c, d)\big).\end{aligned}$$

  Da f und g reguläre Schienenfunktionen aus $\mathbb{F}$ sind, und aus der Vorlesung bekannt ist, dass $\min(f, g) \in \mathbb{F}$ ist, und weiterhin $\min(c, d) \in \mathbb{R}^+$ offensichtlich gilt, ist $\min(\hat{f}, \hat{g})$ eine gemische Schienenfunktion bestehend aus einer regulären Schienenfunktion und einer Konstanten. Somit ist $\hat{\mathbb{F}}$ unter min abgeschlossen.
- *Link-Operation.* Da für eine gemischte Schienenfunktion $\hat{f} = (f, c) \in \hat{\mathbb{F}}$ gilt, dass $\hat{f} = \min(f, c)$, folgt für die Link-Operation zunächst per Definition

$$\begin{aligned}\hat{f} \oplus \hat{g} &= \hat{f} + \hat{g} \circ (\mathrm{id} + \hat{f})\\ &= \min(f, c) + \min(g \circ (\mathrm{id} + \min(f, c)), d).\end{aligned}$$

  Für alle τ mit $c \leq f(\tau)$ folgt damit

$$\begin{aligned}\hat{f} \oplus \hat{g} &= c + \min(g \circ (\mathrm{id} + c), d)\\ &= \min(c + g \circ (\mathrm{id} + c), c + d)\\ &= \min(c \oplus g, c + d).\end{aligned}$$

  Analog folgt für alle τ mit $c > f(\tau)$ dass

$$\begin{aligned}\hat{f} \oplus \hat{g} &= f + \min(g \circ (\mathrm{id} + f), d)\\ &= \min(f + g \circ (\mathrm{id} + f), f + d)\\ &= \min(f \oplus g, f \oplus d)\end{aligned}$$

  Insgesamt gilt also

$$\hat{f} \oplus \hat{g} = \min(c \oplus g, f \oplus d, f \oplus g, c + d).$$

  Aus der Vorlesung ist bekannt, dass $\mathbb{F}$ unter $\oplus$ abgeschlossen ist, also gilt $f \oplus g \in \mathbb{F}$. Weiterhin ist aus Aufgabenteil (b) bekannt, dass für die Link-Operation einer regulären Schienenfunktion mit einer Konstanten gilt, dass das Ergebnis wieder eine reguläre Schienenfunktion ist, das heißt $c \oplus g \in \mathbb{F}$ und $f \oplus d \in \mathbb{F}$. Schreiben wir die letzte Gleichung etwas um, erhalten wir

$$\hat{f} \oplus \hat{g} = \min(\underbrace{\min(c \oplus g, f \oplus d, f \oplus g)}_{=:\, h}, \underbrace{c + d}_{=:\, e}).$$

  Da $\mathbb{F}$ unter min abgeschlossen ist, gilt also $h \in \mathbb{F}$, und weiterhin natürlich $e \in \mathbb{R}^+$. Das heißt $\hat{f} \oplus \hat{g} = (h, e)$, und somit ist $\hat{f} \oplus \hat{g} \in \hat{\mathbb{F}}$. $\square$