Universität Karlsruhe
Fakultät für Informatik
ITI Wagner

Algorithmen für Routenplanung
Sommersemester 2009
Daniel Delling, Thomas Pajor

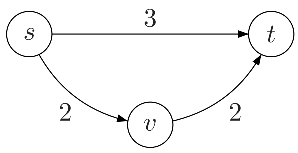
Erstes Theorie-Übungsblatt - Musterlösung

Aufgabe 1: Bidirektionale Suche

(a) Folgender kleiner Graph liefert ein Gegenbeispiel. Der Ablauf des Algorithmus ist wie folgt.

Vorwärtssuche: `settle`(s)

- `insert`$(v,2)$
- `insert`$(t,3)$



Rückwärtssuche: `settle`(t)

- `insert`$(s,2)$
- `insert`$(v,3)$

Vorwärtssuche: `settle`(v)

Rückwärtssuche: `settle`(v)

- `stop`. Ausgegebener Pfad ist $P = [s, v, t]$ mit Länge 4.

Der kürzeste Weg P' ist jedoch $P' = [s, t]$ mit Länge 3.

(b) Wir beweisen die Aussage durch Widerspruch. Sei also $M \in V$ der durch das Abbruchkriterium induzierte Knoten mit $M \in \overrightarrow{S} \cap \overleftarrow{S}$. Angenommen P' enthält einen Knoten $v \in V$ mit $v \notin \overrightarrow{S} \cup \overleftarrow{S}$. Aus der Korrektheit von Dijkstra's Algorithmus folgt

$$\begin{aligned} u \in \overrightarrow{S} \quad &\Leftrightarrow \quad \text{dist}(s, u) \leq \text{dist}(s, M), \\ u \in \overleftarrow{S} \quad &\Leftrightarrow \quad \text{dist}(u, t) \leq \text{dist}(M, t). \end{aligned}$$

Damit gilt

$$\begin{aligned} &\text{dist}(s, v) > \text{dist}(s, M) \text{ und} \\ &\text{dist}(v, t) > \text{dist}(M, t). \end{aligned}$$

Also folgt sofort $\text{len}\, P < \text{len}\, P'$ was ein Widerspruch dazu ist, dass P' der kürzeste Weg ist.

(c) Folgender Algorithmus implementiert eine korrekte bidirektionale Suche auf unserem Abbruchkriterium.

Die Notation sei wie folgt. Für die Warteschlangen Q, die Distanzen dist und die Menge abgearbeiteter Knoten S bezeichne $\overline{Q}$, $\overline{\text{dist}}$ und $\overline{S}$ die jeweiligen Größen für die entgegengesetzte Suche. Ist also $Q = \overrightarrow{Q}$, so bezeichnet $\overline{Q} = \overleftarrow{Q}$. Die Abwechslungsstrategie der beiden Suchen wird über `chooseDirection()` gesteuert und kann beliebig gewählt werden.

Algorithmus 1 : Bidirektionale Suche

Eingabe : Graph $G = (V, E, \text{len})$ und Start- und Zielknoten $s, t \in V$
Ausgabe : Der kürzeste Weg von s nach t

$\overrightarrow{Q}, \overleftarrow{Q} \leftarrow$ Vorwärts- und Rückwärtsschlangen
$\overrightarrow{\text{dist}}, \overleftarrow{\text{dist}} \leftarrow \infty$
$\overrightarrow{S}, \overleftarrow{S} \leftarrow \emptyset$

$\overrightarrow{Q}$`.insert(`s,0`)`, $\overrightarrow{\text{dist}}(s) \leftarrow 0$
$\overleftarrow{Q}$`.insert(`t,0`)`, $\overleftarrow{\text{dist}}(t) \leftarrow 0$

solange nicht $\overrightarrow{Q}$`.empty()` **oder nicht** $\overleftarrow{Q}$`.empty()` **tue**
- $Q \leftarrow$ `chooseDirection()`
- $v \leftarrow Q$`.deleteMin()`
- $S \leftarrow S \cup \{v\}$
- **Wenn** $v \in \overline{S}$
  - **gib aus** Pfad $[s, \ldots, M, \ldots, t]$ der Länge μ
  - **stop**
- **für alle** *ein- bzw. ausgehenden Kanten* $e = (v, w)$ *bzw.* $e = (w, v)$ **tue**
  - **Wenn** $w \in \overline{Q}$ **oder** $w \in \overline{S}$
    - **Wenn** $\mu > \text{dist}(v) + \text{len}(e) + \overline{\text{dist}}(w)$
      - $\mu \leftarrow \text{dist}(v) + \text{len}(e) + \overline{\text{dist}}(w)$
      - $M \leftarrow w$
  - **Wenn** $\text{dist}(w) = \infty$
    - Q`.insert(`w, $\text{dist}(v) + \text{len}(e)$`)`
    - $\text{dist}(w) \leftarrow \text{dist}(v) + \text{len}(e)$
  - **sonst, wenn** $\text{dist}(v) + \text{len}(e) < \text{dist}(w)$
    - Q`.decreaseKey(`w, $\text{dist}(v) + \text{len}(e)$`)`
    - $\text{dist}(w) \leftarrow \text{dist}(v) + \text{len}(e)$

gib aus Kein Pfad gefunden.

(d) Wir beweisen die Aussage in dem wir zeigen dass gilt

$$|\overrightarrow{S} \cap \overleftarrow{S}| = 1 \Rightarrow \text{minKey}(\overrightarrow{Q}) + \text{minKey}(\overleftarrow{Q}) \geq \mu.$$

Sei also $v \in \overrightarrow{S} \cap \overleftarrow{S}$ der durch unser Abbruchkriterium induzierte Knoten. Aus dem Ablauf von Dijkstra's Algorithmus folgt

$$\begin{aligned} &\text{minKey}(\overrightarrow{Q}) \geq \text{dist}(s, v) \text{ und} \\ &\text{minKey}(\overleftarrow{Q}) \geq \text{dist}(v, t). \end{aligned}$$

Damit folgt unmittelbar

$$\begin{aligned} \text{minKey}(\overrightarrow{Q}) + \text{minKey}(\overleftarrow{Q}) &\geq \text{dist}(s, v) + \text{dist}(v, t) \\ &\overset{(b)}{\geq} \mu. \end{aligned}$$

Aufgabe 2: A^*, Aufwärmübung

(a) Bevor wir die Aussage beweisen zeigen wir zunächst ein Lemma.
Lemma 1. *Sei $P := [v_1, \ldots, v_k]$ ein Pfad in G. Dann gilt in G_π*

$$\operatorname{len}_\pi P = \operatorname{len} P - \pi(v_1) + \pi(v_k).$$

Lösung. Der Beweis erfolgt durch nachrechnen:

$$\begin{aligned}
\operatorname{len}_\pi P &= \sum_{i=1}^{k-1} \operatorname{len}_\pi(v_i, v_{i+1}) \\
&= \sum_{i=1}^{k-1} \big(\operatorname{len}(v_i, v_{i+1}) - \pi(v_i) + \pi(v_{i+1})\big) \\
&= \Big(\sum_{i=1}^{k-1} \operatorname{len}(v_i, v_{i+1})\Big) - \pi(v_1) + \pi(v_k) \\
&= \operatorname{len} P - \pi(v_1) + \pi(v_k)
\end{aligned}$$

$\square$

Beide Algorithmen arbeiten gleich, genau dann wenn sie die Knoten in der gleichen Reihenfolge abarbeiten. Für zwei Knoten $u, v \in V$ bezeichne $u \prec v$ dass u vor v abgearbeitet wird. Es gilt:

$$\begin{aligned}
\text{DIJKSTRA:} \quad & u \prec v \quad \Leftrightarrow \quad \operatorname{dist}_\pi(s,u) < \operatorname{dist}_\pi(s,v) \\
A^*\text{:} \quad & u \prec v \quad \Leftrightarrow \quad \operatorname{dist}(s,u) + \pi(u) < \operatorname{dist}(s,v) + \pi(v)
\end{aligned}$$

Damit ist nun

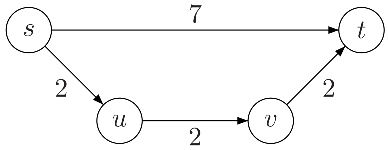
$$\begin{aligned}
& \operatorname{dist}_\pi(s,u) < \operatorname{dist}_\pi(s,v) \\
\overset{\text{lem}}{\Leftrightarrow} & \operatorname{dist}(s,u) - \pi(s) + \pi(u) < \operatorname{dist}(s,v) - \pi(s) + \pi(v) \\
\Leftrightarrow & \operatorname{dist}(s,u) + \pi(u) < \operatorname{dist}(s,v) + \pi(v).
\end{aligned}$$

(b) DIJKSTRA's Algorithmus ist korrekt wenn der Graph keine negativen Zyklen enthält. Wir stellen also sicher, dass alle reduzierten Kosten ≥ 0 sind. Also

$$\begin{aligned}
\operatorname{len}_\pi(u,v) \geq 0 &\Leftrightarrow \operatorname{len}(u,v) + \pi(u) - \pi(v) \geq 0 \\
&\Leftrightarrow \operatorname{len}(u,v) + \pi(u) \geq \pi(v).
\end{aligned}$$

Aufgabe 3: Bidirektionaler A^*

(a) Wir zeigen die Behauptung mittels eines Gegenbeispiels.



Die Potentiale seien wie folgt: Die Vorwärtspotentiale seien $\overrightarrow{\pi}(x) = 0$ für alle Knoten $x \in V$. Die Rückwärtspotentiale seien jedoch $\overleftarrow{\pi}(s) = 0$, $\overleftarrow{\pi}(u) = 2$, $\overleftarrow{\pi}(v) = 4$ und $\overleftarrow{\pi}(t) = 6$.

Der bidirektionale A* Algorithmus arbeitet nun wie folgt.

Vorwärtssuche: `settle`(s)

- `insert`$(u,\ 2+0)$
- `insert`$(t,\ 7+0)$
  Da t bereits von der Rückwärtssuche entdeckt wurde, update $\mu := 7$, $M := t$.

Rückwärtssuche: `settle`(t)

- `insert`$(v, 2+4)$
- `insert`$(s, 7+0)$

Es ist nun

$$\underbrace{\text{minKey}(\overrightarrow{Q})}_{=2} + \underbrace{\text{minKey}(\overleftarrow{Q})}_{=6} \geq \underbrace{\mu_{\overrightarrow{\pi}}}_{=7}$$

stop.

Es wird der Weg $[s, M, t] = [s, t]$ mit Länge 7 ausgegeben. Der kürzeste Weg ist jedoch $[s, u, v, t]$ mit Länge 6.

(b) Wir betrachten zum Beweisen der Aussage einen klassischen bidirektionalen Dijkstra Algorithmus auf den Graphen $G_{\overrightarrow{\pi}}$ und $G_{\overleftarrow{\pi}}$ mit den reduzierten Kosten $\text{len}_{\overrightarrow{\pi}}$ und $\text{len}_{\overleftarrow{\pi}}$.

Es gilt $\overrightarrow{\pi} + \overleftarrow{\pi} = \text{const}$ genau dann wenn für alle Knoten $u, v \in V$ gilt

$$\overrightarrow{\pi}(u) + \overleftarrow{\pi}(u) = \overrightarrow{\pi}(v) + \overleftarrow{\pi}(v).$$

Damit folgt

$$\begin{aligned}
&\overrightarrow{\pi}(u) - \overleftarrow{\pi}(v) = \overrightarrow{\pi}(v) - \overleftarrow{\pi}(u) \\
\Rightarrow &\overrightarrow{\text{dist}}(v,u) + \overrightarrow{\pi}(u) - \overrightarrow{\pi}(v) = \overrightarrow{\text{dist}}(v,u) + \overleftarrow{\pi}(v) - \overleftarrow{\pi}(u) \\
\Rightarrow &\overrightarrow{\text{dist}}(v,u) + \overrightarrow{\pi}(u) - \overrightarrow{\pi}(v) = \overleftarrow{\text{dist}}(v,u) + \overleftarrow{\pi}(v) - \overleftarrow{\pi}(u) \\
\Rightarrow &\text{len}_{\overrightarrow{\pi}}(v,u) = \text{len}_{\overleftarrow{\pi}}(v,u).
\end{aligned}$$

Die Graphen $G_{\overrightarrow{\pi}}$ und $G_{\overleftarrow{\pi}}$ haben also die gleichen reduzierten Kosten auf allen Kanten. Mit der Korrektheit des Bidirektionalen Dijkstra Algorithmus auf dem Abbruchkriterium $\mu \leq \text{minKey}(\overrightarrow{Q}) + \text{minKey}(\overleftarrow{Q})$ folgt somit die Korrektheit des Bidirektionalen A* Algorithmus bezüglich dieses Kriteriums.

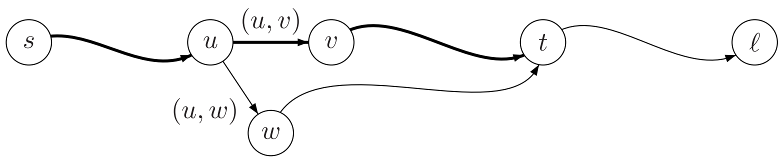
Aufgabe 4: Der ALT-Algorithmus

(a) Um die Aussage zu beweisen betrachten wir wieder den Graphen mit reduzierten Kosten G_π unter Dijkstra's Algorithmus. Um zu beweisen dass Dijkstra's Algorithmus nur Knoten entlang des kürzesten Weges abarbeitet, beweisen wir zwei Aussagen.

1. Ist $(u, v) \in E$ eine Kante auf dem kürzesten s-t-Weg, dann gilt $\text{len}_\pi(u, v) = 0$.

2. Ist $u \in V$ ein Knoten auf dem kürzesten s-t-Weg und $(u, w) \in E$ eine *abzweigende* Kante, das heißt (u, w) liegt nicht auf dem kürzesten s-t-Weg, so ist $\text{len}_\pi(u, w) > 0$.

Damit ergibt sich $\text{dist}_\pi(s, t) = 0$ und $\text{dist}_\pi(s, w) > 0$ für alle Knoten $w \in V$ die nicht auf dem kürzesten s-t-Weg liegen. Das heißt DIJKSTRA's Algorithmus auf G_π arbeitet nur Knoten entlang des kürzesten Weges ab.



Um den Beweis zu vereinfachen zeigen wir zunächst folgende Hilfsaussage.
Lemma 2. *Sei $u \in V$ und der Knoten t liegt auf dem kürzesten Weg von u nach ℓ. Dann ist das Potential von u gerade $\pi(u) = \text{dist}(u, t)$.*

Lösung. Wir rechnen nach:

$$\begin{aligned} \pi(u) &= \max(\underbrace{\text{dist}(u, \ell) - \text{dist}(t - \ell)}_{=\text{dist}(u,t)}, \text{dist}(\ell, t) - \text{dist}(\ell, u)) \\ &= \text{dist}(u, t) \end{aligned}$$

Der zweite Eintrag der Maximumsbildung kann nicht weiter zur Verbesserung des Potentials beitragen, da $\text{dist}(u, t)$ bereits die bestmögliche untere Schranke (Dreiecksungleichung) für den Abstand von u nach t ist. □

Wir zeigen nun die zwei Aussagen:

1. Nachrechnen liefert für Kanten $(u, v) \in E$ auf dem kürzesten s-t-Weg:

$$\begin{aligned} \text{len}_\pi(u, v) &= \text{len}(u, v) - \pi(u) + \pi(v) \\ &= \text{len}(u, v) - \text{dist}(u, t) + \text{dist}(v, t) \\ &= \text{len}(u, v) - \text{len}(u, v) \\ &= 0 \end{aligned}$$

2. Die Aussage $\text{len}_\pi(u, w)$ für abzweigende Kanten $(u, w) \in E$ vom kürzesten Weg zeigen wir durch Widerspruch. Nehmen wir also an, dass $\text{len}_\pi(u, w) = 0$ gilt, das heißt

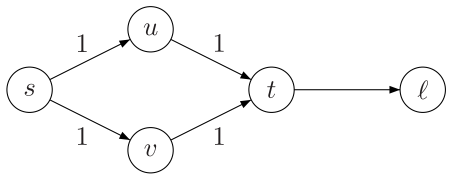
$$\begin{aligned} & 0 = \text{len}_\pi(u, w) \\ \Rightarrow\quad & 0 = \text{len}(u, w) - \pi(u) + \pi(w) \\ \Rightarrow\quad & 0 = \text{len}(u, w) - \text{dist}(u, t) + \pi(w) \\ \Rightarrow\quad & \text{dist}(u, t) = \text{len}(u, w) + \pi(w) \end{aligned}$$

Da für $\pi(w)$ gelten muss $\pi(w) \leq \text{dist}(w, t)$ erhalten wir für den Fall $\pi(w) = \text{dist}(w, t)$ dass $\text{dist}(u, t) = \text{len}(u, w) + \text{dist}(w, t)$ gilt, das heißt der Weg $[u, w, \ldots, t]$ hat die gleiche Länge wie der Weg $[u, v, \ldots, t]$. Somit existieren zwei kürzeste Wege von u nach t was nach Aufgabenstellung ausgeschlossen ist.

Bleibt also nur der Fall $\pi(w) < \mathrm{dist}(w,t)$. Damit erhalten wir allerdings $\mathrm{dist}(u,t) > \mathrm{len}(u,w)+\mathrm{dist}(w,t)$, das heißt der Weg über $[u,w,\ldots,t]$ ist kürzer als der Weg $[u,v,\ldots,t]$ was ein Widerspruch zu der Annahme ist, dass (u,v) auf dem kürzesten Weg liegt.

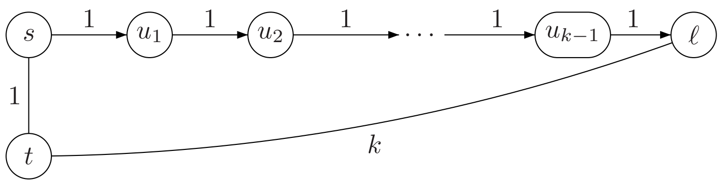
Insgesamt erhalten wir also dass DIJKSTRA's Algorithmus auf G_π nur Kanten entlang des kürzesten s-t-Weges abarbeitet.

(b) Betrachte folgenden Beispielgraph.



Da die kürzesten Wege nicht eindeutig sind ergeben sich für die Kanten (s,u) und (s,v) jeweils reduzierte Kosten 0. Es ist also möglich, dass der ALT Algorithmus sowohl u als auch v abarbeitet.

(c) Wir geben eine Familie $\mathcal{G}$ von Graphen an die die geforderten Eigenschaften erfüllt. Für jedes $k \in \mathbb{N}$ konstruieren wir einen Graphen G_k mit $\Theta(k)$ Knoten wie in der folgenden Abbildung illustriert.

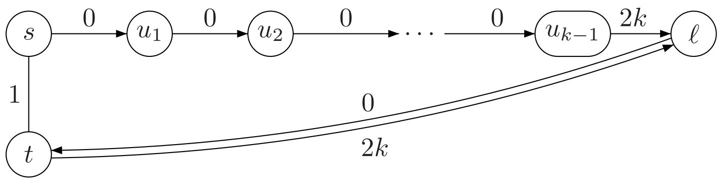


Der Quellknoten s kann mit u_0 und die Landmarke ℓ mit u_k identifiziert werden.

Wir berechnen nun die reduzierten Kosten von G_k. Die Potentiale ergeben sich wie folgt:

$$\begin{aligned}
\pi(u_i) &= \max(\underbrace{\underbrace{\mathrm{dist}(u_i,\ell)}_{=k-i} - \underbrace{\mathrm{dist}(t,\ell)}_{=k}}_{=-i}, \underbrace{\underbrace{\mathrm{dist}(\ell,t)}_{=k} - \underbrace{\mathrm{dist}(\ell,u_i)}_{=k+i+1}}_{=-i-1}) \\
&= -i \\
\pi(\ell) &= \max(\underbrace{\underbrace{\mathrm{dist}(\ell,\ell)}_{=0} - \underbrace{\mathrm{dist}(t,\ell)}_{=k}}_{=-k}, \underbrace{\underbrace{\mathrm{dist}(\ell,t)}_{=k} - \underbrace{\mathrm{dist}(\ell,\ell)}_{=0}}_{=k}) \\
&= k \\
\pi(s) &= \pi(u_0) \\
&= 0 \\
\pi(t) &= \max(\mathrm{dist}(t,\ell) - \mathrm{dist}(t,\ell), \mathrm{dist}(\ell,t) - \mathrm{dist}(\ell,t)) \\
&= 0
\end{aligned}$$

Mit $\mathrm{len}_\pi(u,v) = \mathrm{len}(u,v) - \pi(u) + \pi(v)$ ergeben sich die reduzierten Kosten auf den Graphen wie folgt.



Der Suchraum von DIJKSTRA's Algorithmus auf dem ursprünglichen Graph besteht demnach nur aus $\{s, u_1, t\}$ während der ALT Algorithmus den Suchraum $V \setminus \{\ell\}$ hat, da die reduzierten Kosten entlang aller Kanten (u_{i-1}, u_i) für $i < k$ gerade 0 sind.

Mit $|V| \in \Theta(k)$ ergibt sich somit eine Suchraumverschlechterung um einen Faktor von $\Theta(|V|)$ des ALT Algorithmus gegenüber dem einfachen DIJKSTRA.