Karlsruher Institut für Technologie

Institut für Theoretische Informatik
Lehrstuhl Prof. Dr. D. Wagner

Theorie-Übungsblatt 2

Algorithmen für Routenplanung, Sommer 2010

Ausgabe 26. April 2010
Abgabe keine

Problem 1: Bidirektionale Suche **

Sei $G = (V, E, \text{len})$ ein gerichteter, gewichteter Graph mit eindeutigen kürzesten Wegen, das heißt für je zwei Paare $s, t \in V$ existiert genau ein kürzester s-t-Weg. Bei Bidirektionaler Suche werden nun für eine s-t-Anfrage abwechselnd eine Vorwärts-Suche beginnend bei s und eine Rückwärts-Suche beginnend bei t durchgeführt.

Ein zur Vorlesung alternatives Abbruchkriterium sei wie folgt definiert. Seien $\overrightarrow{S}$ bzw. $\overleftarrow{S}$ die Menge abgearbeiteter Knoten (settled nodes) von der Vorwärts-, bzw. Rückwärtssuche. Sobald eine der beiden Suchen einen Knoten $v \in V$ abarbeitet der bereits von der anderen Suche abgearbeitet wurde, also $|\overrightarrow{S} \cap \overleftarrow{S}| = 1$ erreicht wird, terminiert der Algorithmus und der induzierte Weg

$$P := [s, \dots, v, \dots, t]$$

wird ausgegeben.

(a) Zeigen Sie, dass P nicht notwendigerweise der kürzeste s-t-Weg ist.

(b) Sei P' der kürzester Weg von s nach t. Zeigen Sie, dass P' nur Knoten aus $\overrightarrow{S} \cup \overleftarrow{S}$ nach obigem Abbruchkriterium enthält (Der kürzeste Weg ist also im Suchraum enthalten).

(c) Geben Sie einen Algorithmus in Pseudocode an, der auf obigem Abbruchkriterium beruht und dennoch den kürzesten Weg findet.

(d) Studieren Sie das Abbruchkriterium aus der Vorlesung. Zeigen Sie, dass das hier eingeführte Kriterium sogar *schwächer* als das aus der Vorlesung ist, dass also von jeder der beiden Suchen mindestens so viele Knoten abgearbeitet werden wie bei dem Kriterium aus der Vorlesung.

Problem 2: A* Algorithmus **

Sei $G = (V, E, \text{len})$ ein gerichteter, gewichteter Graph. Weiterhin sei $\pi : V \to \mathbb{R}$ eine zunächst beliebige Potentialfunktion. Wie in der Vorlesung eingeführt, seien die *reduzierten Kosten* len_π definiert durch $\text{len}_\pi(u, v) := \text{len}(u, v) - \pi(u) + \pi(v)$. Analog bezeichne $G_\pi := (V, E, \text{len}_\pi)$ den Graph mit reduzierten Kosten.

(a) Zeigen Sie: DIJKSTRA's Algorithmus auf G_π entspricht genau A* mit Potentialen π.

(b) Welche Anforderung(en) müssen an π zusätzlich gestellt werden, damit beide Algorithmen den kürzesten Weg finden? Begründen Sie Ihre Behauptung(en).

Problem 3: Der ALT-Algorithmus ***

Gegeben sei ein gerichteter, gewichteter Graph $G = (V, E, \text{len})$ mit $|V| = n$ und eindeutigen kürzesten Wegen. Weiterhin sei $\ell \in V$ *eine* Landmarke.

(a) Zeigen Sie, dass für s-t-Anfragen, wobei t auf dem kürzesten Weg von s nach ℓ liegt, der ALT-Algorithmus ausschließlich Knoten entlang des kürzesten s-t-Weges abarbeitet.

(b) Geben Sie ein Gegenbeispiel zu (a) an, falls die Eigenschaft dass alle kürzesten Wege eindeutig sind nicht gilt.

(c) **Knobelaufgabe.**
Zeigen Sie, dass der Suchraum (Anzahl abgearbeiteter Knoten) von ALT um einen Faktor von $\mathcal{O}(n)$ größer sein kann als bei Anwendung des normalen Dijkstra Algorithmus.

Hinweis: Finden Sie eine Familie von Graphen mit Knoten s, t, ℓ für die die Suchraumgröße unter Dijkstra's Algorithmus konstant ist, die reduzierten Kosten unter dem ALT-Algorithmus jedoch bewirken dass zunächst (fast) der ganze Graph vor t abgearbeitet wird.