Institut für Mathematische Optimierung
TU Braunschweig

Prof. Dr. Sándor Fekete
Nils Schweer

WS 06/07

Fortgeschrittene Diskrete Optimierung
Übung 3 vom 23.11.2006

(Abgabe bis zum 30.11.2006, 13:00 durch Einwurf in den Übungskasten im dritten Stock des Forumsgebäudes **vor** dem Raum F 310)

Aufgabe 1 (Der Preflow-Push Algorithmus):
Betrachte folgendes Netzwerk. Löse das zugehörige Maximaler-Fluss-Problem unter An-

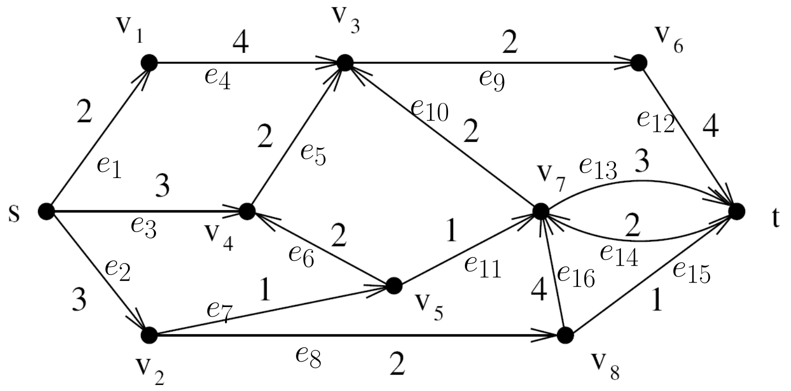

Abbildung 1: Ein Netzwerk mit Kapazitäten

wendung des Preflow-Push-Algorithmus in der Version von Goldberg und Tarjan. Die Liste Q der aktiven Knoten sollte dabei über den gesamten Verlauf des Algorithmuses wie folgt aussehen: $Q = \{v_1, v_4, v_2, v_3, v_5, v_8, v_6, v_7, v_1, v_3, v_1, v_3, \ldots, v_1\}$. Es wird immer der Knoten aus Q gewählt, der möglichst weit vorne steht und noch nicht abgearbeitet ist, d.h. ein Knoten wird solange betrachtet, bis er nicht mehr aktiv ist (ein Knoten kann natürlich später erneut aktiv werden vgl. v_1). Die Auswahl der aktiven Knoten wird nicht vollständig durch den Algorithmus von Goldberg und Tarjan vorgeschrieben, sondern dient nur der Eindeutigkeit für die Korrektur. Die Reihenfolge der betrachteten Kanten muss jedoch immer dieselbe sein. Die Liste $list(v)$ der Kanten die in $\delta^+(v)$ sind, soll immer zuerst die "Vorwärtskanten" (mit aufsteigender Index-Folge) und dann die "Rückwärtskanten" (ebenfalls mit aufsteigender Index-Folge) enthalten.

(10 Punkte)

Aufgabe 2 (Scheduling auf parallelen Maschinen):
In praktischen Anwendungen wie z.B. Fahrplanoptimierung oder Ablaufoptimierung ergeben sich häufig sog. Scheduling-Probleme. Viele Varianten der Scheduling-Probleme sind jedoch NP-vollständig und deshalb existieren (vermutlich) keine effizienten Algorithmen zur Lösung. Eine einfache Variante, die in polynomieller Zeit gelöst werden kann, soll im Folgenden betrachtet werden: Gegeben sei eine Menge J von Aufgaben ("Jobs") die auf M parallelen Maschinen ausgeführt werden sollen. Aufgabe $j \in J$ benötigt p_j Tage für ihre Ausführung. Sie darf jedoch erst ab Tag r_j ("release date", d.h. Freigabedatum) ausgeführt werden und muß bis zu Beginn des Tages d_j ("due date", d.h. Fälligkeitsdatum)

mit $d_j \geq r_j + p_j$ beendet werden. Jede Maschine kann zu einem Zeitpunkt nur eine Aufgabe bearbeiten, und jede Aufgabe kann zu einem Zeitpunkt nur von einer Maschine bearbeitet werden. Allerdings darf eine Aufgabe unterbrochen werden und an verschiedenen Tagen auf verschiedenen Maschinen durchgeführt werden. Gesucht ist ein zulässiger Ablaufplan ("Schedule"), falls einer existiert.

(a) Formuliere dieses Scheduling-Problem als ein äquivalentes Maximaler-Fluss-Problem für allgemeine J, M, p_j, r_j, d_j d.h. zeichne schematisch ein Netzwerk, in dem das Maximaler-Fluss-Problem gelöst werden soll. Für dieses Netzwerk soll gelten, dass wenn es einen maximalen s-t-Fluss der Größe $\sum_{j \in J} p_j$ gibt, dann gibt es einen zulässigen Schedule für das Scheduling-Problem.

Tipp: Sortiere die Zeiten r_j und d_j für alle $j \in J$ in aufsteigender Reihenfolge und bestimme die Menge T der $P \leq 2|J| - 1$ disjunkten Intervalle von aufeinander folgenden Zeitpunkten in dieser sortierten Reihenfolge. Sei $T_{k,l}$ das Zeitintervall, das am Tag k beginnt und am Beginn des Tages $l+1$ endet, d.h. $T_{k,l}$ umfaßt alle Tage von k bis einschlielich l. Beispiel: Steht in der sortierten Reihenfolge r_m direkt vor d_n, dann soll dafür ein $T_{r_m(d_n-1)}$ eingeführt werden. Beachte, dass im Intervall $T_{k,l}$ alle Jobs j mit $r_j \leq k$ und $d_j \geq l+1$ durchgeführt werden können.
Konstruiere ein Netzwerk $G' = (V', E')$ mit $V' = J \cup T \cup \{s, t\}$, Kanten (s, j) mit Kapazität p_j, Kanten $(T_{k,l}, t)$ mit Kapazität $(l - k + 1)M$ und Kanten $(j, T_{k,l})$ mit Kapazität $(l - k + 1)$, wobei $r_j \leq k$ und $d_j \geq l + 1$ gelten müssen.

(b) Zeichne das Netzwerk zu der folgenden Instanz:

$M := 3,\ J := \{1, 2, 3, 4\}$

$p_1 := 1,5,\ p_2 := 1,25,\ p_3 := 2,1,\ p_4 := 3,6$

$r_1 := 3,\ r_2 := 1,\ r_3 := 3,\ r_4 := 5$

$d_1 := 5,\ d_2 := 4,\ d_3 := 7,\ d_4 := 9$

Bestimme einen maximalen s-t-Fluss (durch die Anwendung eines Algorithmuses oder durch "Raten"). Bestimme nun einen zulässigen Schedule (dazu müssen noch die Aufgaben auf die Maschinen verteilt werden).

(c) Begründe mit eigenen Worten, warum im allgemeinem Fall (Teil (a)) immer ein zulässiger Schedule aus einem maximalen s-t-Fluss der Größe $\sum_{j \in J} p_j$ konstruiert werden kann.

(20 Punkte)

Die Fachgruppe Mathematik lädt alle Studierenden herzlich zur Nikolausfeier am 05.12.2006 um 18:00 Uhr in die Hängemathe ein.