Netzwerkalgorithmen

Sommersemester 2022

Übung 4

Aufgabe 1:
Ein *Feasible-Flow-Problem* wird definiert durch ein Netzwerk in dem außer den oberen Kapazitätsschranken u_{ij} für jede Kante $(i, j) \in E$, zusätzlich ein *Supply-Wert* $b(i)$ für jeden Knoten $i \in V$ gegeben ist. Gesucht ist eine Flussfunktion x, die die Kapazitätsbedingung erfüllt, so dass für alle Knoten $i \in V$ gilt:

$$b(i) = \sum_{j \in V \text{ mit } (i,j) \in E} x_{ij} \quad - \sum_{j \in V \text{ mit } (j,i) \in E} x_{ji}$$

Überlegen Sie sich wie man dieses Problem mit einem Maxflow-Algorithmus lösen kann.
Hinweis: Erweitern Sie das Netzwerk um 2 Knoten und einige Kanten.

Aufgabe 2:
Es sei ein maximaler Fluss (d.h. eine Flussfunktion x den den Gesamtfluss $Fmax$ maximiert) gegeben. Zeigen Sie, wie man in diesem Fall einen minimalen s-t–Schnitt in Zeit $O(m)$ berechnen kann.

Aufgabe 3:
Sei x eine Flussfunktion mit Flusswert F und sei F_{max} der Flusswert eines maximalen Flusses. Zeigen Sie, dass dass $G(x)$ einen erhöhenden Pfad mit einer Restkapazität von mindestens $(F_{max} - F)/m$ enthält.

Aufgabe 4:
Der *Maximum-Capacity-Augmentation* Algorithmus sucht in jeder Iteration nach einem erhöhenden Pfad maximaler Restkapazität. Folgern Sie aus Aufgabe 1, dass dieser Algorithmus höchstens $m \log U$ Iterationen zur Berechnung eines maximalen Flusses benötigt.