

3. Übungsblatt zur Vorlesung Graphen und diskrete Optimierung

Abgabe am 10.05.2023 um 10:00 Uhr

Hinweis: Da in der Vorlesung nicht alle Folien zu Strömungen und Flüssen besprochen wurden, können diese Aufgaben alternativ beim Übungsblatt 4 ohne Punkteabzug abgegeben werden. Die betroffenen Aufgaben sind mit einem (*) gekennzeichnet.

Aufgabe 1. (Transportproblem)

7 Punkte

Sie betreiben eine Bäckerei in Würzburg, welche an m Backstätten $B_1, \dots, B_m$ jede Nacht frische Brötchen herstellt. Diese werden morgens an n Filialen $S_1, \dots, S_n$ im Umland ausgeliefert. Die Produktionskapazität der verschiedenen B_i ist gegeben durch a_i (Menge an Brötchen / Stunde) und die Nachfrage in einer Filiale S_j ist gegeben durch b_j (Menge an Brötchen / Stunde).

Das Transportnetz zum Ausliefern der Backwaren besteht aus den Straßen von Würzburg, wobei jede Straße eine maximale Kapazität c an Autos pro Stunde durchlassen kann, bevor es zum Stau kommt.

- (1.1) Modellieren Sie das Problem, jede Filiale (mindestens) mit der benötigten Menge an Brötchen zu beliefern, als Graphen. Wofür stehen die Kanten, Ecken, und Kantengewichte? Wie hängen Bedarf und Produktion zusammen? **3 Punkte**
- (1.2) Wie können Sie einen optimalen Auslieferungsplan finden, der die Nachfrage an Brötchen bedient ohne die Backstätten oder Straßen zu überlasten? **1 Punkt**
- (1.3) Fügen Sie der Modellierung noch Transportkosten k auf jeder Straße und Kosten der Herstellung von p_i pro Brötchen (in Backstätte i) hinzu. Wie verändert sich dadurch Ihr Modell aus (1.1)? (*) **2 Punkte**
- (1.4) Wie können Sie nun einen möglichst günstigsten Auslieferungsplan finden, der die Bedarfe der Filialen abdeckt? (*) **1 Punkt**

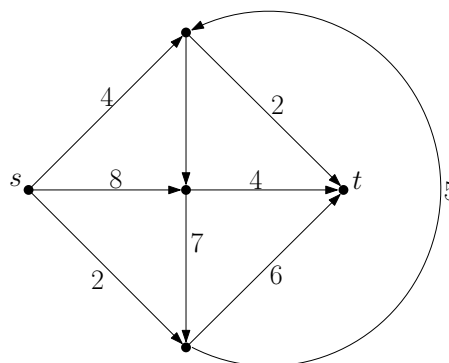
Aufgabe 2. (Programmieraufgabe)

3 Punkte

Implementieren Sie den Edmonds-Karp-Algorithmus in Java. Ihr Algorithmus sollte in jeder Iteration der While-Schleife den berechneten kürzesten augmentierenden Pfad ausgeben.

Sie dürfen für die Berechnung des kürzesten $s - t$ -Weges eine bestehende Implementierung (z.B. von BFS) in Java benutzen, solange Sie diese klar kennzeichnen und Ihre Quelle(n) angeben.

Testen Sie Ihre Implementierung an dem folgenden Graphen, wo die Kantenlabels die obere Kapazität c angeben:



Laden Sie für diese Aufgabe bitte zusätzlich zu der pdf-Abgabe der Lösungen Ihren kommentierten Quellcode (als .java) auf WueCampus hoch.

Aufgabe 3. (*Max-Flow und Min-Cut*)**6 Punkte**

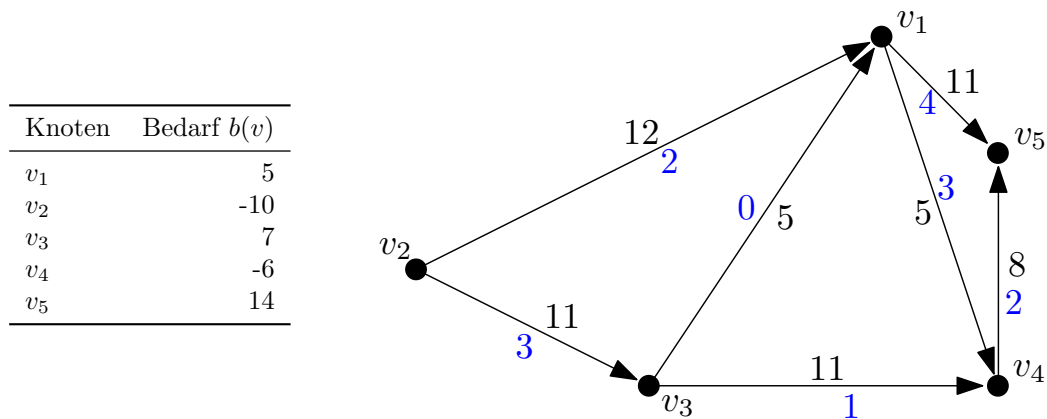
Konstruieren Sie jeweils ein Beispiel für einen gerichteten Graphen mit ganzzahligen Kantengewichten, so dass

- (3.1) es viele verschiedene minimale Schnitte (z.B., exponentiell in der Anzahl der Knoten) und einen eindeutigen maximalen Fluss auf dem Graphen gibt. **2 Punkte**
- (3.2) es viele maximale Flüsse und einen eindeutigen minimalen Schnitt auf dem Graphen gibt. **2 Punkte**
- (3.3) es viele maximale Flüsse und viele minimale Schnitte auf dem Graphen gibt. **2 Punkte**

Zeigen Sie die Korrektheit ihrer Beispiele.

Aufgabe 4. (*b-Fluss (*)*)**4 Punkte**

Gegeben sei folgender Graph $G = (V, E)$ mit unteren Kantenkapazitäten l in Blau und oberen Kantenkapazitäten c in Schwarz:



- (4.1) Konstruieren Sie den Obergraph $\hat{G}$ zu G , indem Sie einen Knoten x und ggf. benötigte Kanten hinzufügen. Wie verändern sich die (unteren und oberen) Kantenkapazitäten? **1 Punkt**
- (4.2) Konstruieren Sie nun einen Graph $\hat{G}'$ mit Quelle s' und Senke t' aus Ihrem Obergraph $\hat{G}$. **1 Punkt**
- (4.3) Fügen Sie folgende Kantenkosten hinzu: **1 Punkt**

Kante	Kosten	Kante	Kosten
(v_2, v_1)	4	(v_3, v_4)	3
(v_1, v_5)	1	(v_1, v_4)	6
(v_2, v_3)	1	(v_4, v_5)	2
(v_3, v_1)	2	sonst	1

- (4.4) Stellen Sie den Residualgraphen zu $\hat{G}'$ auf. **1 Punkt**
- (4.5) **Bonusaufgabe:** Benutzen Sie den in Aufgabe 2 implementierten Algorithmus, um einen maximalen Fluss auf dem Graphen G zu finden. **2 Punkte**

Bitte laden Sie Ihre Lösungen als pdf Datei auf WueCampus hoch. Die Übungsblätter können in Gruppen von maximal zwei Personen bearbeitet werden, geben Sie bei der Abgabe stets beide Namen an.