

2. Übungsblatt

Jasper Gude

Pia Röttgers

4. Mai 2026

/ 21

Aufgabe 1 – Kleinste Knotenüberdeckung

1. Zielfunktion:

$$\min \sum_{v \in V} x_v$$

2. Entscheidungsvariablen:

$$x_v \in \{0, 1\} \quad \forall v \in V$$

Für jeden Knoten wird eine binäre Variable eingeführt welche 1 ist, falls der Knoten x_v Teil der Überdeckungsmenge C ist und andernfalls den Wert 0 annimmt.

3. Nebenbedingung:

$$x_u + x_v \geq 1 \quad \forall \{u, v\} \in E$$

Für jede Kante $\{u, v\} \in E$ muss mindestens einer der beiden Eckpunkte in C enthalten sein.

4. **Korrektheit:** Die Nebenbedingung definiert klar, dass für jede Kante mindestens einer der Knoten den Wert 1 haben und somit in der Ergebnismenge C enthalten sein muss. Dies entspricht genau der Problemstellung.

Da die Summe der Variablen minimiert wird, folgt daraus zwangsläufig die kleinste Knotenüberdeckung.

PuLP Code

/ 3

/ 4

Aufgabe 2 – Straßenreparatur mittels Linearer Programmierung und Flüssen

Gegeben sei ein ungerichteter Graph $G = (V, E)$, der ein Verkehrsnetzwerk darstellt, wobei jeder Knoten $v \in V$ eine Stadt repräsentiert und jede Kante $e = \{u, v\}$ eine Straße zwischen den Städten u und v .

Die Straßen müssen erneuert werden. Dabei sind für jede Straße $e \in E$ Reparaturkosten nötig, die durch eine Funktion $r : E \rightarrow \mathbb{R}_{\geq 0}$ gegeben sind. Zusätzlich hat jede Stadt $v \in V$ nur ein begrenztes Budget, das durch eine Funktion $B : V \rightarrow \mathbb{R}_{\geq 0}$ gegeben ist.

- a) Seien $u_e, v_e \in \mathbb{R}_{\geq 0}$ die Reparaturbeträge, die Stadt u und v für eine inzidente Straße e zahlen. Um u_e, v_e zu erhalten gehen wir durch die Adjazenzliste und vermerken pro Kante die inzidenten Knoten. Dann müssen wir folgendes lineares Programm lösen.

Zielfunktion:

$$\arg \max \sum_{e \in E} u_e + v_e$$

Für alle $e \in E$ gilt

$$r(e) - (u_e + v_e) \leq 0$$

Damit stellen wir sicher, dass die vollen Reparaturkosten bezahlt werden.

Für alle $u \in V$ und zu u inzidente Kanten I gilt

$$\sum_{e \in I} u_e \leq B(u)$$

Damit stellen wir sicher, dass das Budget einer Stadt nicht überschritten wird.

Wenn die Zielfunktion den Wert der Summe aller Reparaturkosten hat, haben wir eine Lösung gefunden.

/ 3

- b) Sei $x_e \in \{0, 1\}$ der Indikator, ob Stadt x die Reparaturkosten der inzidenten Straße e trägt. Dann müssen wir folgendes ganzzahlig lineares Programm lösen.

Zielfunktion:

$$\arg \max \sum_{e \in E} x_e$$

Für alle $e \in E$ und deren inzidente Knoten x, y gilt

$$x_e + y_e \leq 1$$

Damit Stellen wir sicher, dass nur eine Stadt die Reparaturkosten einer Straße trägt.

Für alle $x \in V$ und zu x inzidente Kanten I gilt

$$\sum_{e \in I} x_e \cdot r(x) \leq B(x)$$

Damit stellen wir sicher, dass das Budget einer Stadt nicht überschritten wird.

Wenn die Zielfunktion den Wert der Anzahl an Knoten hat, haben wir eine Lösung gefunden.

/ 2

c) Das Problem kann wie in Abbildung 1 als Fluss modelliert werden.

- Dabei ist s der Startknoten und t der Zielknoten.
- Die Kantenkapazitäten von s zu den Städten $v_i \in V$ ist das Budget $B(v_i)$.
- Die Kantenkapazitäten von v_i zu den Straßen $e_j \in E$ müssen jeweils größer als $r(e_j)$ sein. Jede Straße e_j hat genau zwei eingehende Kanten von den beiden inzidenten Städten.
- Die Kantenkapazitäten von e_j zu t sind die Reparaturkosten $r(e_j)$

Wenn ein zulässiger Fluss alle in t eingehenden Kanten füllt, dann gibt es eine Lösung für das Problem.

Das Modell ist korrekt, da

- Die Beiträge, die eine Stadt v_i leisten kann von den eingehenden Kantenkapazitäten limitiert werden.
- Die Straßen e_j genau zwei eingehende Kanten haben.
- Die Reparaturkosten einer Straße e_j von den ausgehenden Kantenkapazitäten limitiert werden.

/ 4

Aufgabe 3 – Flüsse finden

a) Siehe Abbildung 2.

/ 4

b) Es gibt keinen größeren s - t -Fluss, da die beiden eingehenden Kanten von t vollständig benutzt werden.

/ 1

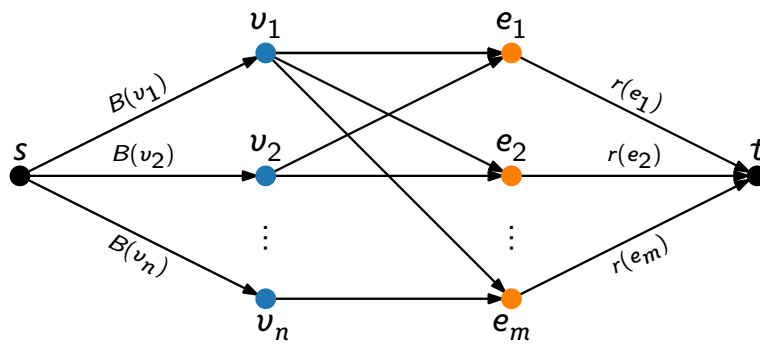


Abbildung 1: Das Problem als Flussproblem. Die blauen Knoten sind Städte und die orangenen Straßen.

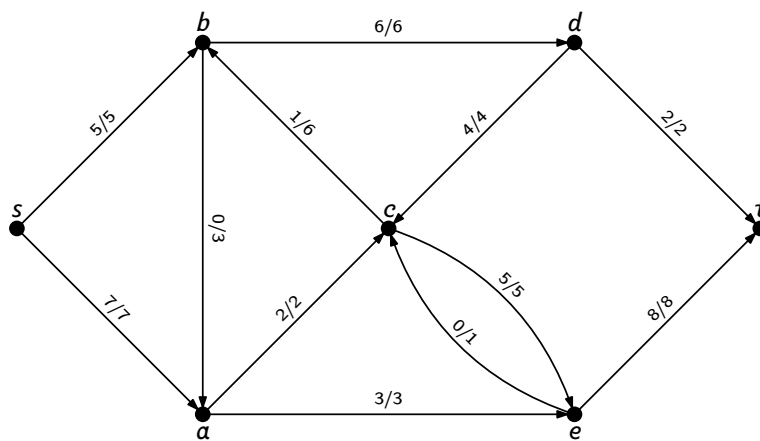


Abbildung 2: Maximaler s-t-Fluss mit Wert 10