Lehrstuhl für Informatik I
Algorithmen, Komplexität und wissensbasierte Systeme
Universität Würzburg

Würzburg, den 14. Mai 2019

Prof. Dr. Alexander Wolff
Johannes Zink, M. Sc.

3. Übungsblatt zur Vorlesung Algorithmische Graphentheorie (Sommer 2019)

CPLEX

Aufgabe 1 – Lineare Programme lösen

Gegeben ist das folgende lineare Programm mit den Variablen x, y und z:

Zielfunktion:	$\text{Maximiere } 2,5x + 1,7y + 1,9z$			
Nebenbedingungen:	$500x$	$+350y$	$+350z$	≤ 57000
	$125x$		$+300z$	≤ 30000
	$125x$	$+275y$	$+100z$	≤ 26000
	$250x$	$+375y$	$+250z$	≤ 55000
Nichtnegativität:	$x \geq 0,$	$y \geq 0,$	$z \geq 0$	

Lösen Sie dieses Programm und geben Sie die optimale Lösung an. Sie können dazu beispielsweise die Software CPLEX verwenden, die wir Ihnen in den Übungen vorstellen. Laden Sie bitte zusätzlich Ihren Quelltext in Wuecampus hoch (Name und s-Nummer auf Ihrer Übungsblattabgabe angeben!), um ihren Lösungsweg aufzuzeigen.

3 Zusatzpunkte

Aufgabe 2 – Lineare Programmierung und Flüsse

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

Hinweis: Sie können hier CPLEX nutzen, um Ihre linearen Programme zu testen. Hier fordern wir explizit eine mathematische Beschreibung Ihres Modells, keinen Code. Beschreiben Sie zunächst Variablen und Constraints in Worten.

a) Für jede Straße $e = \{u, v\} \in E$ dürfen sich die beiden Städte u und v die Kosten $r(e)$ in beliebigem Verhältnis teilen, wobei natürlich keine negativen Zahlungen möglich sind. Wir wollen entscheiden, ob es eine zulässige Verteilung der Zahlungen gibt, so dass alle Renovierungen bezahlt werden, aber keine Stadt ihr Budget überschreitet.

   Geben Sie ein lineares Programm zum Lösen dieses Problems an. Argumentieren Sie, dass Ihre Lösung korrekt ist.

   3 Punkte

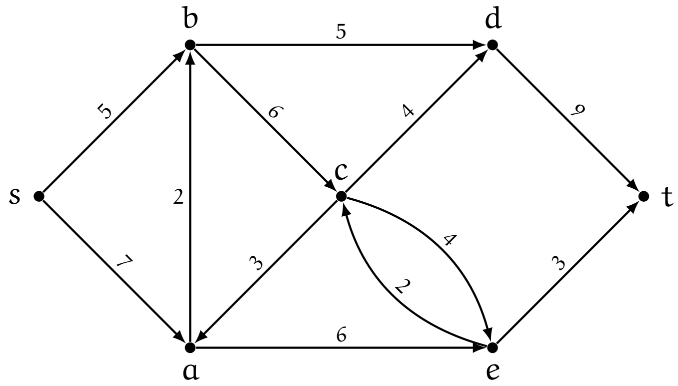
b) Wir wollen erneut die Renovierungskosten verteilen, lassen jetzt aber keine Aufteilung von Einzelkosten auf die Städte mehr zu; das heißt, für jede Straße $e = \{u, v\} \in E$ muss jetzt entweder die Stadt u oder die Stadt v den Gesamtbetrag $r(e)$ zahlen. Wir wollen wieder entscheiden, ob es eine zulässige Verteilung unter dieser Zusatzbedingung gibt.

   Lösen Sie dieses Problem mit Hilfe von *ganzzahliger* linearer Programmierung. Argumentieren Sie, dass Ihre Lösung korrekt ist. **2 Punkte**

c) Lösen Sie das Problem aus Teilaufgabe a) erneut, verwenden Sie jetzt aber keine lineare Programmierung, sondern eine Modellierung als Flussnetzwerk. Argumentieren Sie wieder, dass Ihre Lösung korrekt ist. **4 Punkte**

   Hinweis: Argumentieren Sie, wie aus der Lösung für Teilaufgabe a) ein Fluss in Ihrem Netzwerk abgeleitet werden kann und umgekehrt.

Aufgabe 3 – Flüsse & Schnitte



a) Ermitteln Sie für den angegebenen Graphen einen maximalen Fluss mit einem der in der Vorlesung vorgestellten Algorithmen. Geben Sie in nachvollziehbarer Art und Weise die Zwischenschritte an.

   Zeigen Sie außerdem die Optimalität des von Ihnen gefundenen Flusses, indem Sie einen minimalen Schnitt angeben. **4 Punkte**

b) Gegeben sei ein gerichteter Graph $G = (V, E)$ mit $s, t \in V$ und Kantenkapazitäten $c\colon E \to \mathbb{R}_{>0}$.

   Geben Sie einen effizienten Algorithmus an, der einen minimalen s-t-Schnitt ermittelt. Zeigen Sie die Korrektheit des Algorithmus. Welche Laufzeit hat er? **4 Punkte**

Aufgabe 4 – Längste Wege

Das Problem LÄNGSTER s-t-WEG besteht darin in einem gegebenen ungerichteten Graphen zu einem gegebenen Paar $\{s, t\}$ von Knoten mit $s \neq t$ einen längsten einfachen Weg zu finden. Zur Erinnerung: ein *Weg der Länge* $k \geq 0$ ist eine Folge $\langle v_0, \ldots, v_k \rangle$ mit der Eigenschaft, dass $\{v_0, v_1\}, \{v_1, v_2\}, \ldots, \{v_{k-1}, v_k\}$ Kanten des Graphen sind. Ein Weg ist *einfach*, wenn er jeden Knoten höchstens einmal durchläuft.

Wir wollen zeigen, dass LÄNGSTER s-t-WEG $\mathcal{NP}$-schwer ist, indem wir das Problem HAMILTONWEG auf LÄNGSTER s-t-WEG reduzieren.

a) Sei $G = (V, E)$ ein ungerichteter Graph. Wir erweitern G zu einem Graphen $G' = (V', E')$, dessen Knotenmenge $V' = V \cup \{s, t\}$ um zwei zusätzliche Knoten s und t ergänzt wurde. Außerdem ist $E' = E \cup \{\{s, v\} \mid v \in V\} \cup \{\{v, t\} \mid v \in V\}$, d.h. s und t sind jeweils zu jedem Knoten von G adjazent.

   Erklären Sie, wie man aus einem einfachen Weg der Länge k in G einen einfachen s-t-Weg der Länge $k + 2$ in G' erhält. Erklären Sie ebenfalls, wie man aus einem einfachen s-t-Weg der Länge k in G' einen einfachen Weg der Länge $k - 2$ in G konstruieren kann (zwischen beliebigen Knoten aus V). **2 Punkte**

b) Zeigen Sie: Es gibt einen Hamiltonweg (d.h. einen einfachen Weg der Länge $n - 1$) in G genau dann, wenn es einen einfachen s-t-Weg der Länge $n + 1$ in G' gibt. **1 Punkt**

c) Zeigen Sie damit: LÄNGSTER s-t-WEG ist $\mathcal{NP}$-schwer. **2 Zusatzpunkte**

   Hinweis: Das wichtigste haben wir in der vorherigen Teilaufgabe schon bewiesen. Es fehlen nur noch ein paar Details für eine zulässige Polynomialzeitreduktion.

Bitte werfen Sie Ihre Lösungen bis **Dienstag, 21. Mai 2019, 8:30 Uhr** in den Vorlesungs-Briefkasten im Informatik-Gebäude. Geben Sie stets die Namen und Übungsgruppen aller BearbeiterInnen sowie die Übungsgruppe, in der das Blatt zurückgegeben werden soll, an.

Begründen Sie stets Ihre Behauptungen und kommentieren Sie Ihren Pseudocode!

Aufgaben, die mit $\boxed{\text{CPLEX}}$ gekennzeichnet sind, fordern das Erstellen und Lösen von linearen Programmen. Laden Sie Ihren kommentierten Quellcode auf WueCampus hoch und vermerken Sie auf Ihrer Abgabe, welche BearbeiterIn hochgeladen hat.