Universität Bremen

FB3: Mathematik/Informatik

Prof. Dr. Nicole Megow
Franziska Eberle

Sommer 2019

Operations Research

Übungsblatt 8 (Abgabe am 11.06.2019)

Übung 8.1 (Algorithmus von Floyd und Warshall) (7 Punkte)

Ein Unternehmen verkauft sieben Arten von Kisten. Das Volumen dieser Kisten beträgt zwischen 17 und 33 Liter. Die Nachfrage und Größe jeder Kiste entnehmen Sie bitte unten stehender Tabelle.

Box	1	2	3	4	5	6	7
Größe	33	30	26	24	19	18	17
Nachfrage	400	300	500	700	200	400	200

Die veränderlichen Produktionskosten einer Kiste in Euro entspricht ihrem Volumen. Außerdem entstehen Fixkosten in Höhe von 1 000 Euro, sollte sich das Unternehmen entscheiden, eine bestimmte Art von Kiste zu produzieren. Es ist möglich, die Nachfrage nach einer Kiste mit einer Kiste mit einem größeren Volumen zu erfüllen.

(a) Formulieren Sie ein Problem des kürzesten Weges um die Nachfrage mit minimalen Produktionskosten zu erfüllen. (2 Punkte)

(b) Zeichnen Sie den entsprechenden Graphen. (1 Punkt)

(c) Lösen Sie das Problem mit Hilfe des Algorithmus von Floyd-Warshall. Geben Sie die Distanzmatrix nach jeder Durchführung der äußeren Schleife an. (4 Punkte)

Übung 8.2 (Arbitragemöglichkeiten im Währungsmarkt) (6 Punkte)

Auf dem Währungsmarkt werden Fremdwährungen gehandelt. Der Währungsmarkt ist einer der Märkte mit dem größten Handelsvolumen. Zu jedem Zeitpunkt gibt es für zwei Währungen, bspw. US-Dollar und Euro, zwei Wechselraten: ein Dollar ist r_1 Euro wert und umgekehrt ist ein Euro r_2 Dollar wert. Es gibt eine direkte Arbitragemöglichkeit zwischen zwei Währungen im Währungsmarkt, wenn $r_1 r_2 > 1$ gilt. Als Arbitragemöglichkeit bezeichnet man ein Geschäft, bei dem man risikolos Gewinn machen kann.

Es stellt sich nun die Frage, wie man Arbitragemöglichkeiten mit mehreren Währungen im Währungsmarkt entdecken und gegebenenfalls ausnutzen kann. Betrachten Sie dafür die unten stehenden, hypothetischen Wechselkurse zwischen USD (US-Dollar), EUR (Euro), GBP (Britisches Pfund), AUD (australische Dollar) und JPY (japanische Yen).

	USD	EUR	GBP	AUD	JPY
USD	1	0.639	0.537	1.0835	98.89
EUR	1.564	1	0.843	1.6958	154.773
GBP	1.856	1.186	1	2.014	184.122
AUD	0.9223	0.589	0.496	1	91.263
JPY	0.01011	0.00645	0.00543	0.01095	1

(a) Prüfen Sie, ob es zwei Währungen gibt, die eine direkte Arbitragemöglichkeit darstellen. Ist es möglich, durch den Tausch von Währung 1 in Währung 2 und danach zurück in Währung 1 Gewinn zu machen? (1 Punkt)

(b) Stellen Sie ein LP auf, das herausfindet, ob Arbitragemöglichkeiten mit Wechseln zwischen beliebig vielen Währungen bestehen. Gehen Sie davon aus, dass keine Transaktionskosten anfallen. (2 Punkte)

Hinweis: Verwenden Sie die Variable y_k um abzubilden, wie viel Geld in Währung k nach allen Transaktionen existiert. Verwenden Sie außerdem die Variable x_{jk} um abzubilden, wie viel von Währung k in Währung j getauscht wird.

(c) Lösen Sie Ihr LP mit Hilfe von CPLEX. (2 Punkte)

(d) Welche Tauschgeschäfte müssen Sie durchführen um USD 1 zu erwirtschaften? (1 Punkt)

Hinweis: Falls Ihr LP aus Aufgabe (c) unbeschränkt ist, fügen Sie eine obere Schranke für bspw. y_1 hinzu.

Übung 8.3 (Knapsack mit Greedy) (2+2 Punkte)

Sei durch einen Profitvektor $v \in \mathbb{N}^n$ und einen Gewichtsvektor $w \in \mathbb{N}^n$ sowie durch eine Kapazität $B \in \mathbb{N}$ ein 0-1-Knapsackproblem definiert. Dieses Problem ist $\mathbb{NP}$-schwer; das heißt unter der Annahme $\mathbb{NP} \neq \mathbb{P}$ gibt es keinen Algorithmus mit polynomieller Laufzeit, der das Knapsackproblem exakt löst.

Wir sind dennoch an einer guten Lösung interessiert und betrachten dafür das Verhältnis

$$\frac{\textsc{Opt}(v, w, B)}{A(v, w, B)}.$$

Hier sei $A(v, w, B)$ der Profit einer Knapsacklösung, die von einem Algorithmus A gefunden wurde. Analog ist $\textsc{Opt}(v, w, B)$ der Wert eines *optimalen* Knapsacks mit Profitvektor v, Gewichtsvektor w und Kapazität B.

Falls

$$\frac{\textsc{Opt}(v, w, B)}{A(v, w, B)} \leq \rho$$

für alle möglichen Kombinationen $v, w \in \mathbb{N}^n$ und $B \in \mathbb{N}$ gilt, so hat der Algorithmus A eine Approximationsgüte von ρ.

Seien die Gegenstände nun so sortiert, dass

$$\frac{v_1}{w_1} \geq \frac{v_2}{w_2} \geq \ldots \geq \frac{v_n}{w_n}$$

gilt.

(a) Betrachten Sie folgenden Greedy-Algorithmus für das 0-1-Knapsackproblem.

Algorithmus 1
Finde das maximale k mit $\sum_{j=1}^{k} w_j \leq B < \sum_{j=1}^{k+1} w_j$. Gib $v_G = \sum_{j=1}^{k} v_j$ zurück.

Finden Sie ein Beispiel, sodass die Approximationsgüte von Algorithmus 1 beliebig schlecht ist. (2 Punkte)

(b) Betrachten Sie nun den folgenden, modifizierten Greedy-Algorithmus.

Algorithmus 2
Finde das maximale k mit $\sum_{j=1}^{k} w_j \leq B < \sum_{j=1}^{k+1} w_j$. Gib $v'_G = \max\{\sum_{j=1}^{k} v_j, v_{k+1}\}$ zurück.

Bestimmen Sie die Approximationsgüte von Algorithmus 2. (2 Punkte*)

Übung 8.4 (Knapsack mit DP) (5 Punkte)

In der Vorlesung haben Sie ein DP kennengelernt, mit dem das 0-1-Knapsackproblem gelöst werden kann. Dabei wird in einem Tabelleneintrag (j, b) der maximale Profit gespeichert, der von einer Teilmenge der Gegenständen $1, \dots, j$ mit Gesamtgewicht b erreicht werden kann. Sei durch $v = (22, 64, 48, 100)^T \in \mathbb{R}^4$ und $w = (2, 4, 3, 4) \in \mathbb{R}^4$ sowie $B = 6$ ein 0-1-Knapsackproblem gegeben, wobei v den Profit der Gegenstände und w ihr Gewicht modelliert. Führen Sie das Dynamische Programm aus, dass Sie in der Vorlesung kennengelernt haben.

Übung 8.5 (Knapsack-DP 2) (3 Punkte*)

Alternativ zu dem obigen DP kann man in einem Tabelleneintrag (j, k) auch das minimale Gewicht speichern, mit dem durch eine Teilmenge der Gegenstände $1, \dots, j$ ein Profit k erreicht werden kann. Formulieren Sie das entsprechende DP.

Beachten Sie, dass abgegebener CPLEX-Code für Modellieraufgaben nicht mehr bewertet wird.

---

* Diese Aufgaben können gelöst werden, um Punkte für den Notenbonus zu sammeln. Sie werden *nicht* für die Berechnung der erreichbaren maximalen Punktzahl verwendet.