Deutsche Post Lehrstuhl
für Optimierung von
Distributionsnetzwerken

RWTH AACHEN UNIVERSITY

Methoden und Anwendungen der Optimierung **WS 2017/18**

Prof. Dr. Michael Schneider schroeder@dpo.rwth-aachen.de

Übung 3 - Lösungen

Aufgabe 1 (Greedy Algorithmen - Rucksackproblem):

Gegeben sei eine Menge von Gegenständen mit Gewichten w_i und Profiten p_i. Das Ziel ist es, eine Auswahl $I^* \subseteq I = \{1, 2, \ldots, n\}$ zu treffen, die die Summe der Profite maximiert und gleichzeitig die Kapazität C nicht übersteigt. Das zugehörige binäre Optimierungsproblem lautet:

$$\max z \quad = \quad \sum_{i=1}^{n} x_i p_i \tag{1}$$

sodass

$$\sum_{i=1}^{n} x_i w_i \leq C, \tag{2}$$

$$x_i \in \{\, 0, 1 \,\} \quad \forall i \in I \tag{3}$$

Die Greedy Heuristik für das Rucksackproblem berechnet im ersten Schritt die relativen Profite p_i/w_i für alle Gegestände und sortiert sie dementsprechend. Dann wird solange der bisher noch nicht eingefügte Gegenstand mit dem größten relativen Profit eingepackt, bis die Kapazität es nicht mehr zulässt. Ist dies der Fall, so wird der nächste Gegenstand in der sortierten Liste ausgewählt, der noch in den Rucksack passt. Dies wird wiederholt, bis man am Ende der Liste angekommen ist.

a) Gegeben sei die Instanz des Rucksackproblems aus Tabelle 1. Die Kapazität C sei 15. Führen Sie den Greedy Algorithmus durch und berechnen Sie die Performance der Heuristik.

   Zuerst werden die relativen Profite berechnet.

i	1	2	3	4	5	6	7
p_i	3	4	7	6	2	1	8
w_i	4	2	4	4	3	1	5

Tabelle 1: Instanz eines Rucksackproblems.

i	1	2	3	4	5	6	7
p_i	3	4	7	6	2	1	8
w_i	4	2	4	4	3	1	5
p_i/w_i	0.75	2	1.75	1.5	0.67	1	1.6

Dementsprechend lautet die sortierte Liste der Gegenstände $(2, 3, 7, 4, 6, 1, 5)$**.** P **und** W **seien der kumulierte Profit bzw. das kumulierte Gewicht.**

eingefügt	P	W
2	4	2
3	11	6
7	19	11
4	25	15

Nach dem Einfügen von Gegenstand 4 ist der Rucksack voll.

b) Ist die Lösung, die Sie in Aufgabenteil a) gefunden haben optimal? Wenn ja, wie lässt sich das begründen?

Die Lösung aus Aufgabe a) ist optimal, da sich nur die Gegenstände mit dem höchsten relativen Profit im Rucksack befinden und zusätzlich die gesamte Kapazität des Rucksacks ausgeschöpft ist. Daher ergibt sich für diese Instanz eine Performance der Heuristik $R_H(P1a) = 25/25 = 1$**.**

c) Führen Sie die Greedy Heuristik für die Instanz aus Tabelle 2 durch. Die Kapazität C sei 20. Wie ist die Performance jetzt zu bewerten?

Zuerst werden wieder die relativen Profite berechnet.

Die sortierte Liste lautet demnach $(1, 3, 5, 4, 2)$**.**

Der Reihe nach werden die Gegenstände $1, 3, 5$ **und** 4 **eingefügt, Gegenstand 2 passt nicht mehr hinein. Die Greedy Heuristik stoppt mit der Lösung** $x_{gr} = (1, 0, 1, 1, 1)$ **und dem Zielfunktionswert** $z_{gr} = 30$**. Es ist jedoch leicht einzusehen, dass die Optimallösung** $x^* = (0, 1, 0, 0, 0)$ **ist. Diese hat einen Zielfunktionswert von** $z^* = 60$**. Die Performance der Heuristik ist für diese Instanz daher** $R_H(P1c) = 30/60 = 0.5$**.**

d) Wie ändert sich die Performance der Heuristik, wenn man für die Instanz aus Aufgabe c) eine Kapazität C von 26 annimmt? Welche Lösung ist jetzt optimal,

i	1	2	3	4	5
p_i	6	60	9	7	8
w_i	1	20	2	2	2

Tabelle 2: Instanz eines Rucksackproblems.

i	1	2	3	4	5
p_i	6	60	9	7	8
w_i	1	20	2	2	2
p_i/w_i	6	3	4.5	3.5	4

eingefügt	P	W
1	6	1
3	15	3
5	23	5
4	30	7

und was fällt bezüglich der Optimallösung auf?

An der Sortierung und dem Ablauf des Greedy Algorithmus ändert sich nichts. Er liefert erneut $x_{gr} = (1, 0, 1, 1, 1)$ mit dem Zielfunktionswert $z_{gr} = 30$. Allerdings ist jetzt die Lösung $x^* = (0, 1, 1, 1, 1)$ mit Zielfunktionswert $z^* = 84$ optimal. Die Performance berechnet sich daher zu $R_H(P1d) = 30/84 \approx 0.357$. Auffällig ist bei dieser Instanz, dass in der Optimallösung als einziges der Gegenstand fehlt, der den besten relativen Profit aufweist.

e) Welche Erkenntnis ziehen Sie aus den bisherigen Ergebnissen im Bezug auf die Performance der Greedy Heuristik?

Die Performance der Heuristik hängt stark von der gewählten Instanz ab. Eine Performance für eine Heuristik zu berechnen gibt einem daher keine zuverlässige Aussage über die Güte der Heuristik. Lösung: (i) worst case (ii) average case (iii) empirische Analyse zur Bewertung der Performance heranziehen.

f) Wie steigert sich die Performance der Heuristik für die Instanz aus Aufgabe d) wenn man den Extended-Greedy-Algorithmus aus der Vorlesung anwendet?

Hierbei wird im ersten Schritt der Greedy-Algorithmus angewendet. Dann wird überprüft, ob ein einzelner Gegenstand eine bessere Lösung darstellt als die Greedy-Lösung. In diesem Fall ist der einzelne Gegenstand $i = 2$ besser als die Greedy-Lösung und man kommt auf $x_{ext\_gr} = (0, 1, 0, 0, 0)$ mit Zielfunktionswert $z_{ext\_gr} = 60$. Die Performance lautet $R_{ext\_gr}(P1d) = 60/84 \approx 0.714$.

Aufgabe 2 (Greedy Algorithmen - Bin Packing Problem):

Das Bin Packing Problem beschreibt die Suche nach einer Möglichkeit eine gegebene Menge von n Gegenständen in möglichst wenige Behälter (Bins) zu packen.

$$\begin{aligned}
z_{BP} &= \min \sum_{j=1}^{n} y_j && (1)\\
\text{so dass} \quad & \sum_{i=1}^{n} w_i x_{ij} \leq C y_j \quad \text{für alle } j = 1, \ldots, n && (2)\\
& \sum_{j=1}^{n} x_{ij} = 1 \quad \text{für alle } i = 1, \ldots, n && (3)\\
& y_j \in \{0,1\} \quad \text{für alle } j = 1, \ldots, n && (4)\\
& x_{ij} \in \{0,1\} \quad \text{für alle } i, j = 1, \ldots, n && (5)
\end{aligned}$$

a) Führen Sie zu der Instanz aus Tabelle 3 die *Best-Fit* Heuristik durch. Bei dieser werden die Gegenstände nacheinander so in einen der Bins gepackt, dass die Restkapazität dieses Bins minimal wird. Passt ein Gegenstand in keinen offenen Behälter, wird ein neuer geöffnet. Die Kapazität C jedes Bins sei 8.

   Wie untere Tabelle zeigt, werden 5 Bins benötigt.

b) Berechnen Sie die Anzahl der benötigten Bins, wenn Sie die *Best-Fit-Decreasing* Heuristik benutzen. Dabei werden die Gegenstände vor der Zuordnung zu den Bins nach fallendem Gewicht sortiert.

   Die Sortierung lautet $(1, 4, 6, 7, 2, 3, 8, 5)$**. Wie untere Tabelle zeigt, werden nur 4 Bins benötigt.**

i	1	2	3	4	5	6	7	8
w_i	7	4	3	6	1	5	4	2

Tabelle 3: Instanz eines Binpackingproblems.

Iteration	Bin1	Bin2	Bin3	Bin4	Bin5
1	{1} 1				
2	{1} 1	{2} 4			
3	{1} 1	{2,3} 1			
4	{1} 1	{2,3} 1	{4} 2		
5	{1,5} 0	{2,3} 1	{4} 2		
6	{1,5} 1	{2,3} 1	{4} 2	{6} 3	
7	{1,5} 1	{2,3} 1	{4} 2	{6} 3	{7} 4
8	{1,5} 1	{2,3} 1	{4,8} 0	{6} 3	{7} 4

Tabelle 4: Lösung zu Bin Packing 2a)

Iteration	Bin1	Bin2	Bin3	Bin4	Bin5
1	{1} 1				
2	{1} 1	{4} 2			
3	{1} 1	{4} 2	{6} 3		
4	{1} 1	{4} 2	{6} 3	{7} 4	
5	{1} 1	{4} 2	{6} 3	{7,2} 0	
6	{1} 1	{4} 2	{6,3} 0	{7,2} 0	
7	{1} 1	{4,8} 0	{6,3} 0	{7,2} 0	
8	{1,5} 0	{4,8} 0	{6,3} 0	{7,2} 0	

Tabelle 5: Lösung zu Bin Packing 2b)