

# Methoden und Anwendungen der Optimierung (MAO)

## Kapitel 2: Greedy Algorithmen

Univ.-Prof. Dr. Michael Schneider  
Christian Schröder

Deutsche Post Chair – Optimization of Distribution Networks (DPO)  
RWTH Aachen University

[schroeder@dpo.rwth-aachen.de](mailto:schroeder@dpo.rwth-aachen.de)

WS 2017/18



Deutsche Post  
Chair - Optimization of  
Distribution Networks

**RWTH**AACHEN  
UNIVERSITY

# Gesamtgliederung

- 1 Grundlagen
- 2 **Greedy Algorithmen**
- 3 Lösungsqualität und Approximation
- 4 Lokale Suche
- 5 Metaheuristiken

# Agenda

## 2 Greedy Algorithmen

- Greedy-Heuristik für das Rucksackproblem
- Greedy-Heuristiken für das Bin Packing Problem
- Allgemeines Prinzip der Greedy-Algorithmen

# Heuristiken und das Greedy-Prinzip

## Ziele des Kapitels:

- Das Greedy-Prinzip darstellen können
- Das Konzept von unabhängigen (freien und fixierten) und abhängigen Variablen in Hinblick auf das Greedy-Prinzip kennen
- Eigene Greedy-Algorithmen für neue Problemsstellungen entwickeln können

# Greedy-Algorithmen

*Greedy-Algorithmen* oder Greedy-Heuristiken sind Eröffnungsverfahren zur Konstruktion einer zulässigen Lösung.

Ihr Name kommt vom englischen Wort „greedy“, was mit „gierig“ übersetzt werden kann.

## Merke: Greedy Algorithmen



Die Idee der Greedy-Algorithmen: In jedem Schritt eine Variable auf den Wert zu fixieren, der in diesem Schritt zur *größten Verbesserung* einer geeignet gewählten (Ersatz-)Zielfunktion führt. Nach Fixierung einer Variablen können u.U. andere Variablen fixiert werden, wenn ihre Werte von der getroffenen Festlegung abhängen.

# Übersicht zu Greedy-Algorithmen

Wir beschreiben Greedy-Algorithmen für

- das Rucksack-Problem
- das Bin Packing Problem
- das MST (Wiederholung)

# Agenda

## 2 Greedy Algorithmen

- Greedy-Heuristik für das Rucksackproblem
- Greedy-Heuristiken für das Bin Packing Problem
- Allgemeines Prinzip der Greedy-Algorithmen

# Idee Greedy-Heuristik Rucksackproblem

**Rucksackproblem Notation:**  $n$  Gegenstände mit Profiten  $p_1, \dots, p_n$ , Gewichten  $w_1, \dots, w_n$  und Rucksack mit Kapazität  $C$

**Idee:** die „profitabelsten“ Elemente sind solche, für die der *relative Profit*, also  $p_i/w_i$  am größten ist; diese möchte man „als erstes“ in eine Lösung aufnehmen.



# Pseudocode Greedy-Heuristik Rucksackproblem

---

## Algorithmus 1 : Greedy-Algorithmus für das Rucksack-Problem

---

// Input: absteigend sortierte Elemente  $i \in \{1, \dots, n\}$  nach  
relativem Profit  $p_i/w_i$

SETZE  $C^{Rest} := C$

for  $j = 1$  bis  $n$  (gemäß Sortierung) do

    if  $w_j \leq C^{Rest}$  then

        SETZE  $\bar{x}_j := 1$

        SETZE  $C^{Rest} := C^{Rest} - w_j$

    else

        SETZE  $\bar{x}_j := 0$

// Output:  $(\bar{x}_1, \dots, \bar{x}_n)$

---

# Beispiel Greedy-Heuristik Rucksackproblem I

**Beispiel:**  $n = 5$  Gegenstände mit Profit

$p = (p_1, \dots, p_5) = (3, 7, 2, 8, 4)$  und Gewicht

$w = (w_1, \dots, w_5) = (2, 4, 1, 3, 6)$  und ein Rucksack mit Kapazität  
 $C = 9$

Relative Profite:  $p_i/w_i = (1.5, 1.75, 2, 2.666, 0.6666)$  also Sortierung  
 $j = 4, 3, 2, 1, 5$ .

Setze  $C^{Rest} := C = 9$  und laufe über alle Gegenstände gemäß  
Sortierung

- 1  $w_4 = 3 \leq C^{Rest} = 9$ ? ja, also  $\bar{x}_4 = 1$
- 2  $w_3 = 1 \leq C^{Rest} = 6$ ? ja, also  $\bar{x}_3 = 1$
- 3  $w_2 = 4 \leq C^{Rest} = 5$ ? ja, also  $\bar{x}_2 = 1$
- 4  $w_1 = 2 \leq C^{Rest} = 1$ ? Nein, also  $\bar{x}_1 = 0$
- 5  $w_5 = 6 \leq C^{Rest} = 1$ ? Nein, also  $\bar{x}_5 = 0$

Ergebnis: Lösung mit Profit  $8 + 2 + 7 = 17$ . Optimal?

## Beispiel Greedy-Heuristik Rucksackproblem II

Optimale Lösung:  $x^T = (x_1, \dots, x_5) = (1, 1, 0, 1, 0)$  mit Profit 18.  
D.h., die Greedy-Heuristik ist tatsächlich nur eine Heuristik und liefert hier **keine** optimale Lösung.

Wie könnte eine Greedy Heuristik für das Bin Packing Problem aussehen?

# Agenda

## 2 Greedy Algorithmen

- Greedy-Heuristik für das Rucksackproblem
- Greedy-Heuristiken für das Bin Packing Problem
- Allgemeines Prinzip der Greedy-Algorithmen

# Greedy-Heuristiken Bin Packing Problem I

Idee: Eine gierige Vorgehensweise besteht darin, nacheinander die Gegenstände jeweils einem am besten scheinenden bereits geöffneten Behälter mit ausreichender Restkapazität zuzuordnen, oder falls dies unmöglich ist, den Gegenstand in einen neuen Behälter zuzuordnen.

Zwei fundamentale Festlegungen:

- *Reihenfolge*, in der die *Gegenstände* nacheinander betrachtet werden

Decreasing: Die Gegenstände sind nach fallendem (nicht-steigendem) Gewicht sortiert

(Beliebig:) Die Gegenstände werden z.B. in der gegebenen Reihenfolge zugeordnet.

## Greedy-Heuristiken Bin Packing Problem II

- Die Wahl des *Behälters, in den der Gegenstand gepackt wird*.
  - Next-Fit: Zuordnung zum zuletzt geöffneten Behälter
  - First-Fit: Zuordnung zum geöffneten Behälter mit kleinstem Index
  - Best-Fit: Zuordnung zum Behälter mit kleinster noch ausreichender Restkapazität

# Beispiel Greedy-Heuristik Bin Packing Problem I

## Beispiel für First-Fit:

Instanz mit Behälterkapazität  $C = 10$  und  $n = 8$  Gegenständen mit folgenden Gewichten:

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

**Ablauf** (Eintrag jeweils {Gegenstände}, *Restkapazität*):

Iteration	Behälter 1	Behälter 2	Behälter 3	Behälter 4	Behälter 5
1	{1}, 8				
2	{1, 2}, 5				
3	{1, 2, 3}, 1				
4		{4}, 7			
5		{4, 5}, 3			
6			{6}, 5		
7				{7}, 4	
8					{8}, 2
Lösung	{1, 2, 3}, 1	{4, 5}, 3	{6}, 5	{7}, 4	{8}, 2

# Beispiel Greedy-Heuristik Bin Packing Problem II

## Beispiel für First-Fit-Decreasing:

Instanz mit Behälterkapazität  $C = 10$  und  $n = 8$  Gegenständen mit folgenden sortierten Gewichten:

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

## Ablauf:

Iteration	Behälter 1	Behälter 2	Behälter 3	Behälter 4	Behälter 5
1	{1}, 2				
2		{2}, 4			
3			{3}, 5		
4		{2, 4}, 0	{3}, 5		
5			{3, 5}, 1		
6				{6}, 7	
7				{6, 7}, 4	
8	{1, 8}, 0				
Lösung	{1, 8}, 0	{2, 4}, 0	{3, 5}, 1	{6, 7}, 4	



# Agenda

## 2 Greedy Algorithmen

- Greedy-Heuristik für das Rucksackproblem
- Greedy-Heuristiken für das Bin Packing Problem
- Allgemeines Prinzip der Greedy-Algorithmen

# Allgemeines Prinzip der Greedy-Algorithmen I

Ausgangspunkt des Greedy-Algorithmus ist die statische Unterteilung der Variablen in *abhängige* und *unabhängige Variablen*:

- Diese Unterteilung unterstellt, dass sich die abhängigen Variablen aufgrund der Werte der unabhängigen Variablen mit Hilfe eines anderen (effizienten) Algorithmus bestimmen lassen.
- Der Wert der unabhängigen Variablen wird mit Hilfe des Greedy-Algorithmus schrittweise ermittelt. Hierzu seien die unabhängigen Variablen die Variablen  $x_j$  mit  $j \in J$  mit *Indexmenge*  $J = \{1, \dots, n\}$ .

# Beispiel: Greedy-Heuristik für das Bin Packing Problem I

$$z_{BP} = \min \sum_{j=1}^n y_j$$

so dass  $\sum_{i=1}^n w_i x_{ij} \leq C y_j \quad \text{für alle } j \in \{1, \dots, n\}$

$$\sum_{j=1}^n x_{ij} = 1 \quad \text{für alle } i \in \{1, \dots, n\}$$

$$y_j \in \{0, 1\} \quad \text{für alle } j \in \{1, \dots, n\}$$

$$x_{ij} \in \{0, 1\} \quad \text{für alle } i, j \in \{1, \dots, n\}$$

Angenommen, man hat bereits die Zuordnung der Gegenstände zu den Bins entschieden, d.h.  $(x_{ij}) = (\bar{x}_{ij})$  ist gegeben. Dann kann man die besten zugehörigen Werte  $(\bar{y}_j)$  für  $(y_j)$  bestimmen gemäß  $\bar{y}_j := \max_{i=1, \dots, n} \bar{x}_{ij}$ . Die  $y$ -Variablen sind abhängig von den unabhängigen  $x$ -Variablen.

## Allgemeines Prinzip der Greedy-Algorithmen II

Die Indexmenge  $J$  der *unabhängigen Variablen* wird in zwei Mengen partitioniert:

- Die erste Menge  $J^{fix} \subseteq J$  enthält die Indizes der *fixierten* Variablen,
- die zweite Menge  $J^{frei} \subseteq J$  die Indizes der *freien* Variablen.

In jedem Schritt eines Greedy-Algorithmus gilt:

$$J^{fix} \cup J^{frei} = J \text{ und } J^{fix} \cap J^{frei} = \emptyset.$$

Zunächst sind alle Variablen freie Variablen, d.h.  $J^{frei} = J$  und  $J^{fix} = \emptyset$ .

- In jedem Iterationsschritt des Algorithmus wird eine freie Variable fixiert, d.h. ein  $j \in J^{frei}$  entfernt und zur Menge  $J^{fix}$  hinzugefügt.
- Aufgrund der bereits erfolgten Fixierung einiger Variablen wird der Bereich der zulässigen Werte der noch freien Variablen eingeschränkt.

## Allgemeines Prinzip der Greedy-Algorithmen III

Angenommen, die fixierten Variablen  $x_k$ ,  $k \in J^{fix}$  seien auf die Werte  $\bar{x}_k$  gesetzt worden:

$$x_k := \bar{x}_k \quad \text{für alle } k \in J^{fix}$$

Es bezeichne  $X_j(\bar{x}_k, k \in J^{fix})$  die dann möglichen Werte der freien

Variable  $x_j$ ,  $j \in J^{frei}$ .

Die Auswahl der *nächsten zu fixierenden Variablen* und die Bestimmung des *besten Wertes* geschieht durch Lösen eines Hilfsproblems  $P_j$  für jede freie Variable  $x_j$ ,  $j \in J^{frei}$ .

**Hilfsproblem  $P_j$  (Maximierungsproblem):**

$$\begin{aligned} z_j &= \max \bar{c}(x_j) \\ \text{so dass} \quad & x_j \in X_j(\bar{x}_k, k \in J^{fix}) \end{aligned}$$

Hierbei ist  $\bar{c}(x_j)$  eine (Ersatz-)Zielfunktion.

## Beispiel: Greedy-Heuristik für das Bin Packing Problem II

In einer Teillösung ist der Behälter  $j$  bereits mit den Gewichten

$$\sum_{k:(k,j) \in J^{fix}} w_k \bar{x}_{kj}$$

gefüllt und die Restkapazität des  $j$ -ten Behälters ist damit

$$C - \sum_{k:(k,j) \in J^{fix}} w_k \bar{x}_{kj}.$$

Ein Gegenstand  $i$  passt also **nicht mehr** in den  $j$ -ten Behälter, wenn

$$w_i + \sum_{k:(k,j) \in J^{fix}} w_k \bar{x}_{kj} > C$$

gilt. In diesem Fall kann man setzen:

$$X_{ij}(\bar{x}_{kj}, (k,j) \in J^{fix}) := \{0\}$$

# Allgemeines Prinzip der Greedy-Algorithmen IV

---

---

SETZE  $J^{fix} := \emptyset$  und  $J^{frei} := J$

**repeat**

    Löse das Hilfsproblem  $P_j$  für alle  $j \in J^{frei}$ .

    Es sei  $j^* \in J^{frei}$  der Index der nächsten zu fixierenden Variablen

    SETZE  $x_{j^*}^*$  auf den im Hilfsproblem  $P_{j^*}$  ermittelten Wert  $x_{j^*} := \bar{x}_{j^*}$

    SETZE  $J^{fix} := J^{fix} \cup \{j^*\}$  und  $J^{frei} := J^{frei} \setminus \{j^*\}$

**if** *zusätzliche Variablen aus  $J^{frei}$  können fixiert werden* **then**

        └ Fixiere diese und aktualisiere  $J^{frei}$  und  $J^{fix}$

**until**  $J^{fix} = J$

---

# Allgemeines Prinzip angewandt auf BP I

## Einteilung in unabhängige und abhängige Variablen:

unabhängig sind  $x_{ij}$  für  $i, j \in \{1, \dots, n\}$

abhängig sind  $y_j$  für  $j \in \{1, \dots, n\}$

Also

$$J = \{(i, j) : i, j \in \{1, \dots, n\}\}$$



## Allgemeines Prinzip angewandt auf BP II

**Best-Fit-Decreasing:** Die Gegenstände  $1, 2, \dots, n$  seien bereits nach fallendem (nicht-steigendem) Gewicht sortiert.

Es sei  $i^{next} := \min \{i : x_{ij} \in J^{frei}\}$  nächster Index des nächsten einzufügenden Gegenstands.

**Hilfsproblem:**

$$(P_{ij}) \quad \min z_{ij} = \begin{cases} C - \sum_{k:(k,j) \in J^{fix}} w_k \bar{x}_{kj}, & \text{falls } i = i^{next} \text{ und} \\ w_i + \sum_{k:(k,j) \in J^{fix}} w_k \bar{x}_{kj} \leq C \\ \infty, & \text{sonst} \end{cases}$$

**Fixierung:**

Fixiere die gewählte Variable  $x_{i^{next},j}$  auf 1

Fixiere  $x_{i^{next},j'}$  für alle  $j' \neq j$  auf 0

# Allgemeines Prinzip angewandt auf KP

**Rucksackproblem (KP):** Alle Variablen sind unabhängig.

$$z_{KP} = \max \sum_{j=1}^n p_j x_j$$

$$\text{so dass} \quad \sum_{i=1}^n w_j x_j \leq C$$
$$x_1, \dots, x_n \in \{0, 1\}$$

**Hilfsproblem:**

$$(P_j) \quad \max z_j = \begin{cases} \frac{p_j}{w_j}, & \text{falls } w_j \leq C - \sum_{k \in J^{\text{fix}}} w_k \bar{x}_k \\ 0, & \text{sonst} \end{cases}$$

**Fixierung:**

Fixiere die gewählte Variable  $x_{j^*}$  mit maximalem Wert  $z_{j^*}$  auf  $\bar{x}_{j^*} = 1$

Fixiere alle übrigen Variablen  $x_k$ , welche in  $P_k$  Bewertung  $z_k = 0$  haben, auf  $\bar{x}_k = 0$  (diese zusätzliche Fixierung ist optional)

## Allgemeines Prinzip angewandt auf MST I

---

### Algorithmus 2 : Algorithmus von Kruskal

---

// Input: sortierte Kanten  $E$  nach steigenden Kosten

SETZE  $T = \emptyset$

**repeat**

    Wähle nächste Kante  $\{i, j\} \in E$  gemäß der Sortierung

**if** *Hinzufügen von  $\{i, j\}$  zu  $T$  erzeugt keinen Kreis* **then**

        └ Füge  $\{i, j\}$  zu  $T$  hinzu.

**until** *alle Kanten wurden durchlaufen*

// Output: MST  $(V, T)$

---

## Allgemeines Prinzip angewandt auf MST II

**Hilfsproblem:** Jede Kante  $\{i, j\} \in E$  hat eine unabhängige Binärvariable  $x_{ij} \in \{0, 1\}$  (Kante in Baum ja/nein)

$$(P_{ij}) \quad \min z_{ij} = \begin{cases} c_{ij}, & \text{falls durch Hinzufügen von } \{i, j\} \text{ kein Kreis entsteht} \\ \infty, & \text{sonst} \end{cases}$$

### Fixierung:

Fixiere gewählte Variable  $x_{ij}$  (minimaler Wert  $z_{ij}$ ) auf 1

Fixiere alle Variablen  $x_{kl}$  auf 0, welche in  $P_{kl}$  Bewertung  $z_{kl} = \infty$  haben (diese zusätzliche Fixierung ist optional)

# Greedy-Algorithmen: Optimalität und Zulässigkeit

Greedy-Algorithmen sind *optimal* bei *Matroid-Struktur* des Problems

Greedy-Algorithmen können oftmals für Probleme mit komplexen Nebenbedingungen **nicht** garantieren, dass sie eine *zulässige Lösung* liefern.

## Beispiele:

- Einhalten der Zeitfenster-Bedingungen beim TSPTW oder VRPTW
- Benutzung einer vorgegebenen Anzahl von Fahrzeugen bei Varianten des VRP

# Greedy Algorithmen und Zulässigkeit

## Auswege:

- Man gibt sich mit unzulässigen Lösungen zufrieden. Diese kann man ggf. mit Strafkosten bewerten und darauf hoffen, dass ein Verbesserungsverfahren die Zulässigkeit herstellt.
- Man wiederholt den Greedy-Algorithmus, jedoch
  - mit modifizierten Kosten,
  - mit modifizierten Koeffizienten oder
  - mit modifiziertem (Ersatz-)Problem zur Auswahl einer nächsten zu fixierenden Variablen,in der Hoffnung, so eine andere dann zulässige Lösung zu erhalten.

## Zur Vertiefung...



- (Domschke und Drexl, 2007), Abs. 6.2 und 6.3
- (Grünert und Irnich, 2005), Abs. 3.8.1

- [Domschke und Drexl 2007] Domschke, W. ; Drexl, A.:  
*Einführung in Operations Research*. 7. Auflage. Berlin : Springer,  
2007
- [Grünert und Irnich 2005] Grünert, T. ; Irnich, S.: *Optimierung  
im Transport Band I: Grundlagen*. Aachen : Shaker Verlag, 2005