Methoden und Anwendungen der Optimierung (MAO)

Kapitel 3: Lösungsqualität und Approximation

Univ.-Prof. Dr. Michael Schneider
Christian Schröder

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

schroeder@dpo.rwth-aachen.de

WS 2017/18

Gesamtgliederung

Agenda

Lösungsqualität und Approximation

Ziele des Kapitels:

- Wissen, wie man die Lösungsqualität einer Heuristik analysieren kann
- Worst-Case-Performance-Analyse an Beispielen erklären und durchführen können
- Unterschiede und Gemeinsamkeiten von Worst-Case-Analyse, Average-Case-Analyse und empirischer Analyse erklären können

Agenda

Performance Analyse

Ziel: Formalisierung von „möglichst gut lösen“

Performance Analyse

Ziel: Formalisierung von „möglichst gut lösen“

Gegeben: Eine Instanz $P = (X, c)$ eines Problems Π und eine Heuristik H. Eine optimale Lösung x^* von P erfülle $z_{opt}(P) = c(x^*) > 0$.

Performance Analyse

Ziel: Formalisierung von „möglichst gut lösen“

Gegeben: Eine Instanz $P = (X, c)$ eines Problems Π und eine Heuristik H. Eine optimale Lösung x^* von P erfülle $z_{opt}(P) = c(x^*) > 0$.

Performance-Verhältnis (Definition)

Das Performance-Verhältnis oder kurz die Performance der Heuristik H für die Instanz P ist $R_H(P) = \frac{z_H(P)}{z_{opt}(P)}$.

Performance Analyse

Performance-Verhältnis (Definition)

Das Performance-Verhältnis oder kurz die Performance der Heuristik H für die Instanz P ist $R_H(P) = \frac{z_H(P)}{z_{opt}(P)}$.

Bemerkungen:

- Ein Algorithmus A, welcher für alle Instanzen P des Problems Π ein Performance-Verhältnis von 1 hat, ist ein

Performance Analyse

Performance-Verhältnis (Definition)

Das Performance-Verhältnis oder kurz die Performance der Heuristik H für die Instanz P ist $R_H(P) = \frac{z_H(P)}{z_{opt}(P)}$.

Bemerkungen:

- Ein Algorithmus A, welcher für alle Instanzen P des Problems Π ein Performance-Verhältnis von 1 hat, ist ein

Performance Analyse

Performance-Verhältnis (Definition)

Das Performance-Verhältnis oder kurz die Performance der Heuristik H für die Instanz P ist $R_H(P) = \frac{z_H(P)}{z_{opt}(P)}$.

Bemerkungen:

- Ein Algorithmus A, welcher für alle Instanzen P des Problems Π ein Performance-Verhältnis von 1 hat, ist ein exakter Optimierungsalgorithmus

Performance Analyse

Performance-Verhältnis (Definition)

Das Performance-Verhältnis oder kurz die Performance der Heuristik H für die Instanz P ist $R_H(P) = \frac{z_H(P)}{z_{opt}(P)}$.

Bemerkungen:

- Ein Algorithmus A, welcher für alle Instanzen P des Problems Π ein Performance-Verhältnis von 1 hat, ist ein exakter Optimierungsalgorithmus
- Für Minimierungsprobleme ist immer

Performance Analyse

Performance-Verhältnis (Definition)

Das Performance-Verhältnis oder kurz die Performance der Heuristik H für die Instanz P ist $R_H(P) = \frac{z_H(P)}{z_{opt}(P)}$.

Bemerkungen:

- Ein Algorithmus A, welcher für alle Instanzen P des Problems Π ein Performance-Verhältnis von 1 hat, ist ein exakter Optimierungsalgorithmus
- Für Minimierungsprobleme ist immer

Performance Analyse

Performance-Verhältnis (Definition)

Das Performance-Verhältnis oder kurz die Performance der Heuristik H für die Instanz P ist $R_H(P) = \frac{z_H(P)}{z_{opt}(P)}$.

Bemerkungen:

- Ein Algorithmus A, welcher für alle Instanzen P des Problems Π ein Performance-Verhältnis von 1 hat, ist ein exakter Optimierungsalgorithmus
- Für Minimierungsprobleme ist immer $R_H(P) \geq 1$
- Für Maximierungsprobleme ist immer $R_H(P) \leq 1$

Performance Analyse

Beispiel Next-Fit (aus Kapitel 2):
Instanz $P = (n, C, w)$ 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:

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

Performance Analyse

Beispiel (Forts.): In Kapitel 2 hatten wir bereits mittels First-Fit-Decreasing (FFD) eine andere Lösung mit nur 4 Behältern gefunden.

Performance Analyse

Beispiel (Forts.): In Kapitel 2 hatten wir bereits mittels First-Fit-Decreasing (FFD) eine andere Lösung mit nur 4 Behältern gefunden.

Da die Summe der Gewichte $\sum_{i=1}^{8} w_i = 35$ und $C = 10$, sind mindestens $4 = \lceil 3.5 \rceil$ Behälter notwendig, d.h. $z_{opt}(P) = 4$.

Performance Analyse

Beispiel (Forts.): In Kapitel 2 hatten wir bereits mittels First-Fit-Decreasing (FFD) eine andere Lösung mit nur 4 Behältern gefunden.

Da die Summe der Gewichte $\sum_{i=1}^{8} w_i = 35$ und $C = 10$, sind mindestens $4 = \lceil 3.5 \rceil$ Behälter notwendig, d.h. $z_{opt}(P) = 4$.

Also ist das Performance-Verhältnis

$$R_{Next-Fit}(P) = \frac{5}{4} = 1{,}25.$$

Performance Analyse

Beispiel Greedy-Rucksack (aus Kapitel 2): Rucksackproblem mit $n = 5$ Gegenständen mit Profit $p = (p_1, \ldots, p_5) = (3, 7, 2, 8, 4)$ und Gewicht $w = (w_1, \ldots, w_5) = (2, 4, 1, 3, 6)$ und ein Rucksack mit Kapazität $C = 9$. Die zugehörige Instanz ist $P = (n, p, w, C) = (5, (3, 7, 2, 8, 4), (2, 4, 1, 3, 6), 9)$.

Performance Analyse

Beispiel Greedy-Rucksack (aus Kapitel 2): Rucksackproblem mit $n = 5$ Gegenständen mit Profit $p = (p_1, \ldots, p_5) = (3, 7, 2, 8, 4)$ und Gewicht $w = (w_1, \ldots, w_5) = (2, 4, 1, 3, 6)$ und ein Rucksack mit Kapazität $C = 9$. Die zugehörige Instanz ist $P = (n, p, w, C) = (5, (3, 7, 2, 8, 4), (2, 4, 1, 3, 6), 9)$.

- Ergebnis des Greedy-Algorithmus: Lösung $x^\top = (x_1, \ldots, x_5) = (0, 1, 1, 1, 0)$ mit Profit $7 + 2 + 8 = 17$

Performance Analyse

Beispiel Greedy-Rucksack (aus Kapitel 2): Rucksackproblem mit $n = 5$ Gegenständen mit Profit $p = (p_1, \ldots, p_5) = (3, 7, 2, 8, 4)$ und Gewicht $w = (w_1, \ldots, w_5) = (2, 4, 1, 3, 6)$ und ein Rucksack mit Kapazität $C = 9$. Die zugehörige Instanz ist $P = (n, p, w, C) = (5, (3, 7, 2, 8, 4), (2, 4, 1, 3, 6), 9)$.

- Ergebnis des Greedy-Algorithmus: Lösung $x^\top = (x_1, \ldots, x_5) = (0, 1, 1, 1, 0)$ mit Profit $7 + 2 + 8 = 17$
- Optimale Lösung: $x^\top = (x_1, \ldots, x_5) = (1, 1, 0, 1, 0)$ mit Profit 18.

Performance Analyse

Beispiel Greedy-Rucksack (aus Kapitel 2): Rucksackproblem mit $n = 5$ Gegenständen 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 $C = 9$. Die zugehörige Instanz ist $P = (n, p, w, C) = (5, (3, 7, 2, 8, 4), (2, 4, 1, 3, 6), 9)$.

- Ergebnis des Greedy-Algorithmus: Lösung $x^\top = (x_1, \dots, x_5) = (0, 1, 1, 1, 0)$ mit Profit $7 + 2 + 8 = 17$
- Optimale Lösung: $x^\top = (x_1, \dots, x_5) = (1, 1, 0, 1, 0)$ mit Profit 18.
- Performance-Verhältnis der Greedy-Heuristik Gr ist

$$R_{Gr}(P) = 17/18 = 0{,}9\bar{4}.$$

Performance Analyse

Die Schreibweise $R_H(P)$ macht deutlich, dass hier die Performance der Heuristik H bezogen auf eine konkrete Instanz $P \in \Pi$ beschrieben ist.

Performance Analyse

Die Schreibweise $R_H(P)$ macht deutlich, dass hier die Performance der Heuristik H bezogen auf eine konkrete Instanz $P \in \Pi$ beschrieben ist.

Man ist aber daran interessiert, allgemeine Aussagen über die Heuristik H zu erhalten. Hierzu bieten sich folgende Alternativen an:

1. Worst-Case-Analyse
2. Average-Case-Analyse
3. empirische Analyse, d.h. eine Analyse bezogen auf eine gewählte Menge von Instanzen P

Agenda

3 Lösungsqualität und Approximation

- Einführung in die Performance Analyse
- Worst-Case und Average-Case Analyse
- Empirische Analyse

Worst-Case-Analyse

Bei der Worst-Case-Analyse interessiert man sich für das schlechteste Performance-Verhältnis, welches sich für eine Heuristik H ergeben kann.

Worst-Case-Analyse

Bei der Worst-Case-Analyse interessiert man sich für das schlechteste Performance-Verhältnis, welches sich für eine Heuristik H ergeben kann.

Für ein Minimierungsproblem (Maximierungsproblem)heißt

$$R_H := \sup_{P \in \Pi} R_H(P) \quad \left(R_H := \inf_{P \in \Pi} R_H(P) \right)$$

die *Worst-Case-Performance* von H.

Worst-Case-Analyse

Bei der Worst-Case-Analyse interessiert man sich für das schlechteste Performance-Verhältnis, welches sich für eine Heuristik H ergeben kann.

Für ein Minimierungsproblem (Maximierungsproblem)heißt

$$R_H := \sup_{P \in \Pi} R_H(P) \quad \left(R_H := \inf_{P \in \Pi} R_H(P) \right)$$

die *Worst-Case-Performance* von H.

Bemerkung: In der Regel gibt es für ein Problem Π unendlich viele Instanzen $P = (X, c) \in \Pi$. Daher gibt es i.Allg. nicht unbedingt *eine* Instanz mit schlechtester Worst-Case-Performance und somit nicht unbedingt ein Minimum/Maximum. Man verwendet also „sup“ (Supremum) statt „max“ und „inf“ (Infimum) statt „min“.

Approximationsalgorithmus

Für eine beliebige Zahl $\varepsilon \geq 0$ ist eine Heuristik H ein ε-**Approximationsalgorithmus**, falls

$$R_H \leq 1 + \varepsilon$$

im Fall der Minimierung und

$$R_H \geq 1 - \varepsilon$$

im Fall der Maximierung gilt.

Approximationsalgorithmus

Für eine beliebige Zahl $\varepsilon \geq 0$ ist eine Heuristik H ein ε-**Approximationsalgorithmus**, falls

$$R_H \leq 1 + \varepsilon$$

im Fall der Minimierung und

$$R_H \geq 1 - \varepsilon$$

im Fall der Maximierung gilt.

Beispiel: Für ein Minimierungsproblem und eine Heuristik H gilt:

$$R_H = 1{,}2 \qquad \rightarrow \qquad H \text{ ist ein } \tfrac{1}{5}\text{-Approximationsalgorithmus}$$

Beispiel: Greedy-Algorithmus Rucksackproblem

Beispiel: Wir hatten gerade gesehen, dass der Greedy-Algorithmus *Gr* für das Rucksackproblem $R_{Gr} \leq 17/18$ erfüllen muss.
Tatsächlich gilt:

Beispiel: Greedy-Algorithmus Rucksackproblem

Beispiel: Wir hatten gerade gesehen, dass der Greedy-Algorithmus *Gr* für das Rucksackproblem $R_{Gr} \leq 17/18$ erfüllen muss.
Tatsächlich gilt:

> Die Greedy-Heuristik *Gr* für das Rucksackproblem kann beliebig schlechte Lösungen liefern, d.h. $R_{Gr} = 0$.

Wie?

Beispiel: Greedy-Algorithmus Rucksackproblem II

Zu zeigen: Zu beliebigem $k > 0$ gibt es eine Instanz P_k des Rucksackproblems, so dass $z_{Gr}(P_k) \leq k \cdot z_{opt}(P_k)$ gilt.

- Wähle $n = 2, C = M$ mit M ganzzahlig und $M \geq 2/k$, $p = (p_1, p_2) = (2, M)$, $w = (w_1, w_2) = (1, M)$.
- Sortierung ist $(1, 2)$, denn $2/1 = 2 > M/M = 1$.
- Folglich liefert Greedy immer die Lösung $(1, 0)$ mit Profit 2.
- **Aber:** Die optimale Lösung ist für $M > 2$ immer $x^* = (0, 1)$ mit Profit M.
  Also: $R_{Greedy} \leq R_{Greedy}(P_k) = \frac{2}{M} \leq k$

Beispiel: Greedy-Algorithmus Rucksackproblem III

Beispiel (Forts.): Der Greedy-Algorithmus für das Rucksackproblem lässt sich zu einem Algorithmus Ext-Gr mit Worst-Case-Performance $R_{Ext-Gr} = \frac{1}{2}$ (hier gleichbedeutend mit $\frac{1}{2}$-Approximationsalgorithmus) verbessern. Beweis in (Kellerer u. a., 2004, S. 34).

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

// Input: Instanz $P = (n, p, w, C)$

LÖSE P mit dem Greedy-Algorithmus (Ergebnis sei $(\bar{x}_1, \ldots, \bar{x}_n) \in \{0,1\}^n$)

// Ist einzelner Gegenstand k^* besser als Greedy-Lösung?

SETZE $k^* := \arg\max_{j=1,\ldots,n} p_j$

if $p^\top \bar{x} < p_{k^*}$ **then**

  SETZE $\bar{x} = (0, \ldots, 0, 1, 0, \ldots, 0)^\top = e_{k^*}$

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

Worst-Case-Analyse Bin Packing Problem

Übersicht zu Next-/First-/Best-Fit-Heuristiken:

Algorithmus H	Laufzeit	R_H^∞
NF	$\mathcal{O}(n)$	2
FF	$\mathcal{O}(n \log n)$	1,7
BF	$\mathcal{O}(n \log n)$	1,7
NFD	$\mathcal{O}(n \log n)$	1,691...
FFD	$\mathcal{O}(n \log n)$	1,222...
BFD	$\mathcal{O}(n \log n)$	1,222...

aus: (Martello und Toth, 1990, S. 224)

Average-Case-Analyse

Manchmal ist man jedoch nicht am Worst-Case, sondern am Average-Case interessiert.

In diesem Fall wird eine Wahrscheinlichkeitsverteilung auf der Menge der Instanzen definiert und Aussagen über den Erwartungswert und die Varianz der Performance werden abgeleitet.

Die wesentlichen Schwierigkeiten bei dieser Art von Analyse sind:

- Das Definieren einer sinnvollen Verteilung über einer in der Regel unendlich großen Menge von Instanzen P des Problems Π
- Das Ausrechnen entsprechender Erwartungswerte und Varianzen

Agenda

3 Lösungsqualität und Approximation
- Einführung in die Performance Analyse
- Worst-Case und Average-Case Analyse
- Empirische Analyse

Empirische Analyse

In der Praxis ist die empirische Analyse weit verbreitet. Hier definiert man eine Menge $\mathcal{I} \subset \Pi$ von relevanten Instanzen des Problems Π.

Empirische Analyse

In der Praxis ist die empirische Analyse weit verbreitet. Hier definiert man eine Menge $\mathcal{I} \subset \Pi$ von relevanten Instanzen des Problems Π.

Das empirische Performance-Verhältnis ergibt sich als Mittelwert der Performance über diese Instanzen:

$$\bar{R}_H(\mathcal{I}) := \frac{1}{|\mathcal{I}|} \sum_{P \in \mathcal{I}} R_H(P)$$

Empirische Analyse

In der Praxis ist die empirische Analyse weit verbreitet. Hier definiert man eine Menge $\mathcal{I} \subset \Pi$ von relevanten Instanzen des Problems Π.

Das empirische Performance-Verhältnis ergibt sich als Mittelwert der Performance über diese Instanzen:

$$\bar{R}_H(\mathcal{I}) := \frac{1}{|\mathcal{I}|} \sum_{P \in \mathcal{I}} R_H(P)$$

Ebenfalls relevant sind die maximale Abweichung,

$$\max_{P \in \mathcal{I}} R_H(P)$$

und die empirische Varianz der Abweichung,

$$\frac{1}{|\mathcal{I}| - 1} \sum_{P \in \mathcal{I}} (R_H(P) - \bar{R}_H(\mathcal{I}))^2.$$

Empirische Analyse

Beispiel: Man generiert zufällig oder speichert aus realen Anwendungen heraus eine (große) Anzahl von Benchmark-Instanzen P zu Definition von $\mathcal{I}$.

Empirische Analyse

Beispiel: Man generiert zufällig oder speichert aus realen Anwendungen heraus eine (große) Anzahl von Benchmark-Instanzen P zu Definition von $\mathcal{I}$.

Die maximale Abweichung und empirische Varianz sind maßgebend für die Robustheit einer Heuristik.

Eine Heuristik kann als robust bezeichnet werden, wenn die Varianz der Performance gering ist und die maximale Abweichung nicht wesentlich vom Mittelwert der Performance verschieden ist.

Zur Vertiefung...

- (Grünert und Irnich, 2005), Abs. 3.8 (bis S. 190)
- (Kellerer u. a., 2004), Abs.2.5 und 2.6
- (Martello und Toth, 1990), Abs. 8-8.2

[Grünert und Irnich 2005] Grünert, T. ; Irnich, S.: *Optimierung im Transport Band I: Grundlagen*. Aachen : Shaker Verlag, 2005

[Kellerer u. a. 2004] Kellerer, H. ; Pferschy, U. ; Pisinger, D.: *Knapsack problems*. Berlin : Springer, 2004

[Martello und Toth 1990] Martello, S. ; Toth, P.: *Knapsack Problems: Algorithms and Computer Implementations*. New York : Wiley, 1990 (Wiley Intersience Series in Discrete Mathematics and Optimization)