



Technische
Universität
Braunschweig



Algorithmen und Datenstrukturen 2

Arne Schmidt

Vorstellung

Interessen

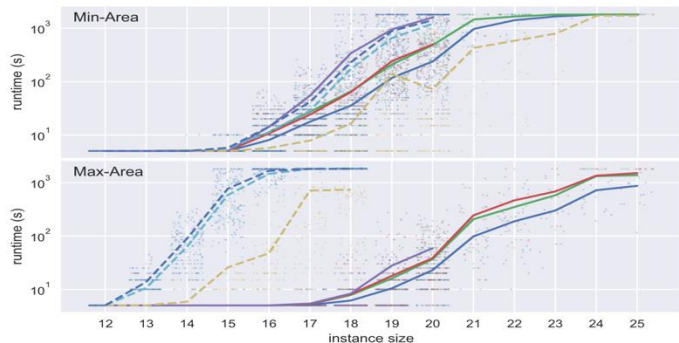
- Geometrische Optimierung,
- Programmierbare Materie,
- Komplexitätstheorie

#GernePerDu

Computing Area-Optimal Simple Polygonizations

SÁNDOR P. FEKETE, ANDREAS HAAS, PHILLIP KELDENICH, MICHAEL PERK, and ARNE SCHMIDT, Department of Computer Science, TU Braunschweig

We consider the problem of finding a simple polygonization of a set of points in the plane. This problem is NP-hard, and we provide a polynomial-time algorithm for finding a simple polygonization of a set of points in the plane.

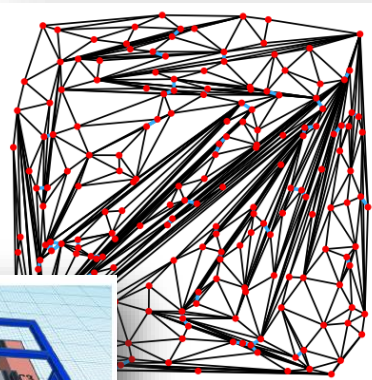


Computing MaxMin Edge Length Triangulations

Sándor P. Fekete* Winfried Hellmann* Michael Hemmer* Arne Schmidt*
Julian Trogel*

Abstract

whil
hull
,
algorithm
of a set of
ity of finding
as a natural
problem by
te. Moreover,
polynomial-
imate MELT
edge

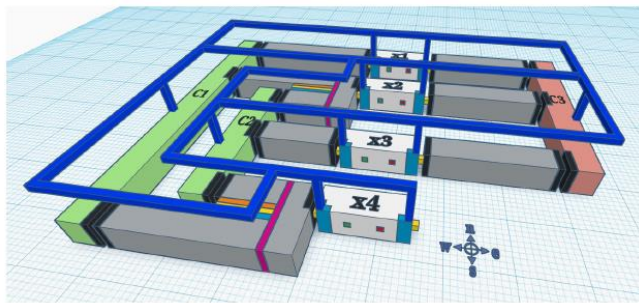


Particle-Based Assembly Using Precise Global Control*

Jo Keller¹[0000-0001-9988-953X], Christian Rieck¹[0000-0003-0846-5163],
and Scheffer²[0000-0002-3471-2706], and Arne Schmidt¹[0000-0001-8950-3963]

¹ Department of Computer Science, TU Braunschweig, Germany.
{jkeller, rieck, aschmidt}@ibr.cs.tu-bs.de
² Department of Computer Science, University of Münster, Germany.
christian.

a micro- and n
ternal force like
particles that c



Das Team

Vorlesung



Arne Schmidt

aschmidt@ibr.cs.tu-bs.de

Gr. Übung



Ramin Kosfeld

kosfeld@ibr.cs.tu-bs.de



Chek-Manh Loi

loi@ibr.cs.tu-bs.de

Kl. Übung

Ash

Finn

Henrik

Lisa

Fragen



Inhaltlich oder
allgemeiner Ablauf

Vorlesung / große Übung



Zu Übungsblättern
oder Korrektur

Kleine Übungen / Tutoren



Individuelle Fragen

Immer per Mail an loi@ibr.cs.tu-bs.de und
kosfeld@ibr.cs.tu-bs.de (Nicht über StudIP)



Sprechstunde

Montags, 09:45 Uhr im Raum IZ 333
(Am besten vorher per Mail ankündigen)

Die Webseite (aud2.ibr.cs.tu-bs.de)



Algorithmen und Datenstrukturen 2

Im Sommersemester 2025

[Startseite](#)

[Vorlesungen](#) ▾

[Kleine Übung & Hausaufgaben](#)

[Kontakt](#)

[Archiv](#)

Algorithmen und Datenstrukturen 2

Die Vorlesung **Algorithmen und Datenstrukturen 2** ist eine Wahlpflichtveranstaltung für Studierende der Informatik, Wirtschaftsinformatik, Informations- und Systemtechnik; außerdem ist sie offen für interessierte Studierende anderer Studiengänge.

Algorithmen sind das methodische Herz der theoretischen und praktischen Informatik; Datenstrukturen ermöglichen die effiziente Umsetzung von Algorithmen und den effizienten Zugriff auf Input- und Outputdaten. In dieser weiterführenden Vorlesung werden die folgenden grundlegenden Begriffe erarbeitet:

- Elementare Aspekte zu Heuristiken
- Exakte Verfahren: Dynamic Programming, Branch-and-Bound
- Approximationsalgorithmen
- Komplexitätsaspekte
- Hashing

Allgemeines

Organisation

Vorlesung

Dienstag 9:45-11:15 Uhr, SN 19.2

Erste Vorlesung: 15.04.2025

Große Übung

Mittwoch 15:00-16:30 Uhr, SN 19.1

Erste große Übung: 16.04.2025

Klausur

Voraussichtlich am Freitag, den

08.08.2025, 15:30-17:30 im Bunker

Semesterplan ([hier klicken](#))

Hier gibt es eine vollständige und stets aktuelle Übersicht über alle Termine der Veranstaltung.

Mailingliste (<https://lists.ibr.cs.tu-bs.de/postorius/lists/aud2.ibr.cs.tu-bs.de/>)



Postorius

✉ Listen

🗄 Archiv

➔ Anmelden

👤 Registrieren

Aud2 aud2@ibr.cs.tu-bs.de

Zusammenfassung

Benutzen Sie folgende Adresse, um die Listen-Besitzer zu kontaktieren: *aud2-owner@ibr.cs.tu-bs.de*

You have to sign in to visit the archives of this list.

Mitglied werden/Mitgliedschaft beenden

To subscribe or unsubscribe from this list, please sign in first. If you have not previously signed in, you may need to set up an account with the appropriate email address.

Anmelden

You can also subscribe without creating an account. If you wish to do so, please use the form below.

Ihre E-Mail-Adresse

Ihr Name (optional)

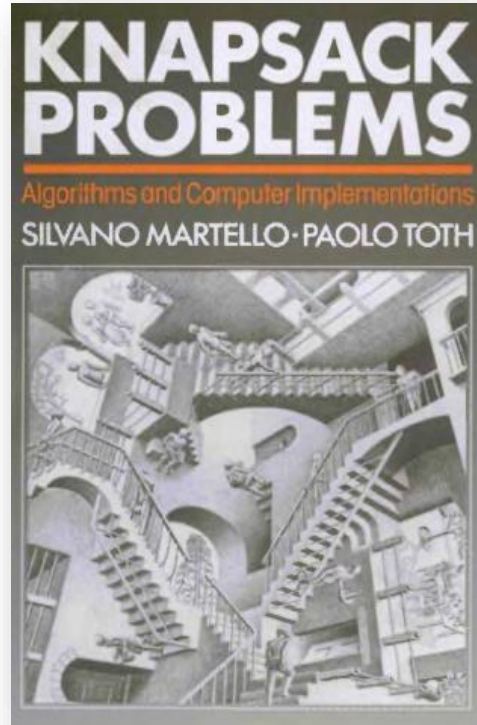
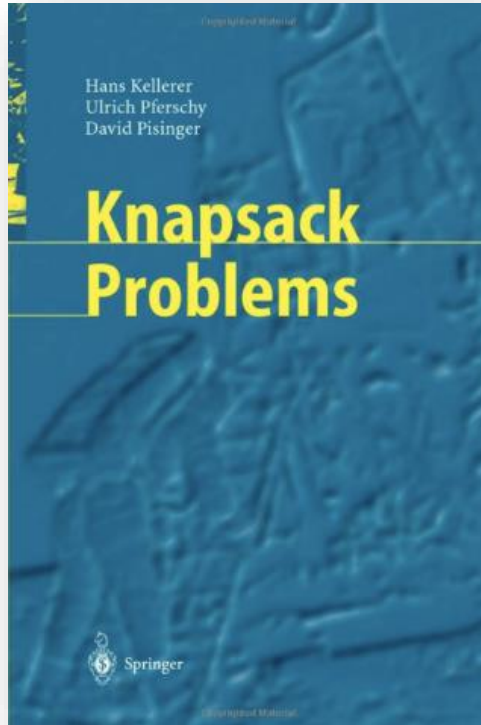
Abonnieren

Semesterplan

Semesterplan AuD II SS25

Woche (KW)	Woche (Datum)	Vorlesung (Di.)	Gr. Übung (Mi.)	Kl. Übung	HA Ausgabe (Fr.)	HA Abgabe (Fr. bis 13:30 Uhr)	HA und Präsenz Besprechung
15	07.04.	/					
16	14.04.	1	0				
17	21.04.	2			HA1		
18	28.04.	3	1	0			P1
19	05.05.	4	2		HA2	HA1	
20	12.05.	5		1			HA1+P2
21	19.05.	6	3		HA3	HA2	
22	26.05.	7		2			HA2+P3
23	02.06.	8			HA4	HA3	
24	09.06.				Exkursionswoche		
25	16.06.	9	4	3			HA3+P4
26	23.06.	10			HA5	HA4	
27	30.06.	11	5	4			HA4+P5
28	07.07.	12				HA5	
29	14.07.	13	6	5			HA5+P6

Literatur



Algorithmen und Datenstrukturen 2

Sándor P. Fekete
PDF-X Version: Arne Schmidt

21. Juni 2024

Institut für Betriebssysteme und Rechnerverbund
Abteilung Algorithmen

Kapitel 1 – Einführung Knapsack-Probleme

Klausursituation



Klausursituation – 30 Minuten später

6 Punkte sicher!
10 schaffe ich nicht.
Noch 44 Punkte, um
zu bestehen...

Ich weiß jetzt aber,
wie lange die letzten
Aufgaben benötigen
und wie viele Punkt
ich mind. erhalte!



Beispiel 1.1

Gegeben:

Aufgabe i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Zeit z_i	20	32	40	8	16	4	32	40	8	32	28	20	16	20	40	24
Punkte p_i	3	3	10	5	2	4	2	9	2	5	3	9	10	3	10	4

Zeitschranke: 120

Punktschranke: 44

Gesucht:

Eine Menge $S \subseteq \{1, \dots, 16\}$ mit $\sum_{i \in S} z_i \leq 120$ und $\sum_{i \in S} p_i \geq 44$,
falls diese überhaupt existiert.

Fragen



Beispiel 1.1 (Gemeinsame Lösung)

Zeitschranke: 120

Punktschranke: 44

Aufgabe i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Zeit z_i	20	32	40	8	16	4	32	40	8	32	28	20	16	20	40	24
Punkte p_i	3	3	10	5	2	4	2	9	2	5	3	9	10	3	10	4

Beispiel 1.1 (Sortiert nach z_i/p_i)

Zeitschranke: 120

Punktschranke: 44

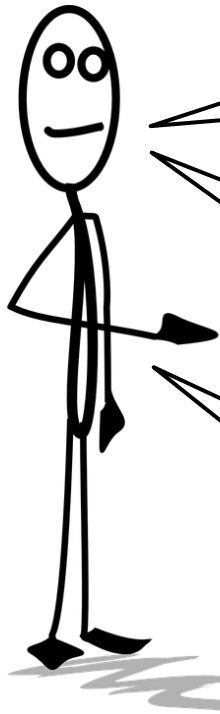
Aufgabe i	6	4	13	12	9	3	15	8	16	10	1	14	5	11	2	7
Zeit z_i	4	8	16	20	8	40	40	40	24	32	20	20	16	28	32	32
Punkte p_i	4	5	10	9	2	10	10	9	4	5	3	3	2	3	3	2
	↑	↑	↑	↑	↑	↑			↑							

Wir wählen Aufgaben 3, 4, 6, 9, 12, 13, 16.

Zusammen:

120 Punkte, 44 Minuten

0-1-Knapsack



Das können wir
formal festhalten:

Oft auch nur als
Frage: Gibt es
diese Menge S ?

Dies nennt man
**Entscheidungs-
problem**

Problem 1.2 (Rucksackproblem, 0-1-KNAPSACK).

Gegeben:

- n Objekte $1, \dots, n$ mit jeweils Größe $z_i > 0$ Gewinn $p_i > 0$
- Größenschranke Z
- Gewinnschranke P

Gesucht:

Eine Menge

$$S \subseteq \{1, \dots, n\}$$

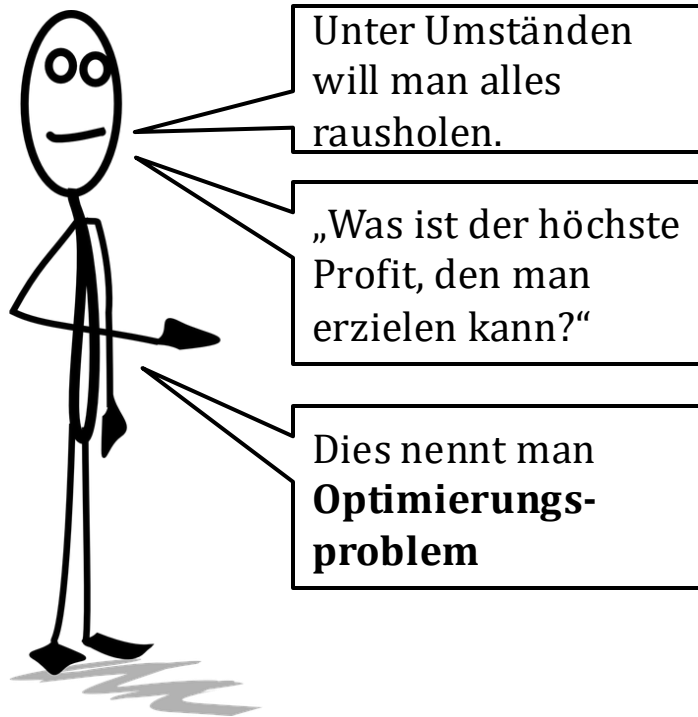
mit

$$\sum_{i \in S} z_i \leq Z$$

und

$$\sum_{i \in S} p_i \geq P$$

Maximum Knapsack



Problem 1.2' (MAXIMUM KNAPSACK).

Gegeben:

- n Objekte $1, \dots, n$ mit jeweils Größe $z_i > 0$ und Gewinn $p_i > 0$
- Größenschranke Z

Gesucht:

Eine Menge

$$S \subseteq \{1, \dots, n\}$$

mit

$$\sum_{i \in S} z_i \leq Z$$

und

$$\sum_{i \in S} p_i = \text{Maximal}$$

Greedy für Maximum Knapsack

Algorithmus (GREEDY₀)

Eingabe: $z_1, \dots, z_n, Z, p_1, \dots, p_n$

Ausgabe: $S \subseteq \{1, \dots, n\}$ inklusionsmaximal mit

$$\sum_{i \in S} z_i \leq Z$$

1. Sortiere $\{1, \dots, n\}$ nach z_i/p_i aufsteigend.
Das ergibt die Permutation $\pi(1), \dots, \pi(n)$.
2. Setze $j = 1$ und $S = \emptyset$.
3. while ($j \leq n$)
 - a. if ($\sum_{i \in S} z_i + z_{\pi(j)} \leq Z$) then
 - i. Füge $\pi(j)$ zu S hinzu.
 - b. Setze $j = j + 1$
4. return S

Verfahren:

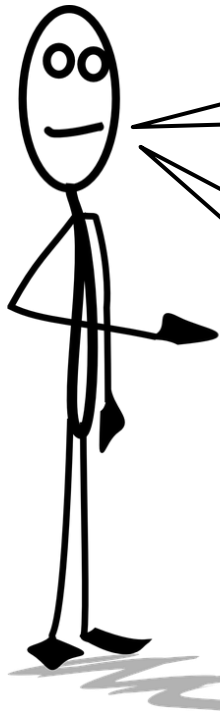
Sortiere alle Elemente

Gehe alle Elemente in
dieser Reihenfolge durch.

Wenn es noch passt,
nimm es auf.



Fractional Knapsack



Manchmal ist es erlaubt, Objekte zum Teil zu nutzen.

Diese nennt man oft **Relaxierung des ganzzahligen Problems**

Problem 1.3 (FRACTIONAL KNAPSACK).

Gegeben:

- n Objekte $1, \dots, n$ mit jeweils Größe $z_i > 0$ und Gewinn $p_i > 0$
- Größenschranke Z

Gesucht:

Für jedes Objekt ein Wert

$$x_i \in [0, 1]$$

sodass

$$\sum_{i=1}^n z_i x_i \leq Z$$

und

$$\sum_{i=1}^n p_i x_i = \textit{Maximal}$$

Beispiel 1.1 (Sortiert nach z_i/p_i)

Zeitschranke: 120

Punktschranke: 44

Aufgabe i	6	4	13	12	9	3	15	8	16	10	1	14	5	11	2	7
Zeit z_i	4	8	16	20	8	40	40	40	24	32	20	20	16	28	32	32
Punkte p_i	4	5	10	9	2	10	10	9	4	5	3	3	2	3	3	2
	↑	↑	↑	↑	↑	↑	↑									

Wir wählen Aufgaben 3, 4, 6, 9, 12, 13 und zu einem Teil Aufgabe 15.

Zu wie vielen Teilen können wir Aufgabe 15 wählen?

3, 4, 6, 9, 12, 13 kosten 96 Minuten; bleiben noch 24 → Aufgabe 15 kann zu 24 von 40 Minuten gelöst werden (Anteil 0.6) und bringt damit $\frac{24}{40} \cdot 10 = 0.6 \cdot 10 = 6$.

Zusammen:

120 Minuten, 46 Punkte

Greedy für Fractional Knapsack

Algorithmus 1.4 Greedy-Algorithmus für FRACTIONAL KNAPSACK

Eingabe: $z_1, \dots, z_n, Z, p_1, \dots, p_n$

Ausgabe: $x_1, \dots, x_n \in [0, 1]$

mit

$$\sum_{i=1}^n z_i x_i \leq Z$$

und

$$\sum_{i=1}^n p_i x_i = \text{Maximal}$$

1: Sortiere $\{1, \dots, n\}$ nach $\frac{z_i}{p_i}$ aufsteigend;
Dies ergibt die Permutation $\pi(1), \dots, \pi(n)$.

Setze $j = 1$.

2: **while** $(\sum_{i=1}^j z_{\pi(i)} \leq Z)$ **do**

3: $x_{\pi(j)} := 1$

4: $j := j + 1$

5: Setze $x_{\pi(j)} := \frac{Z - \sum_{i=1}^{j-1} z_{\pi(i)}}{z_{\pi(j)}}$

6: **return**

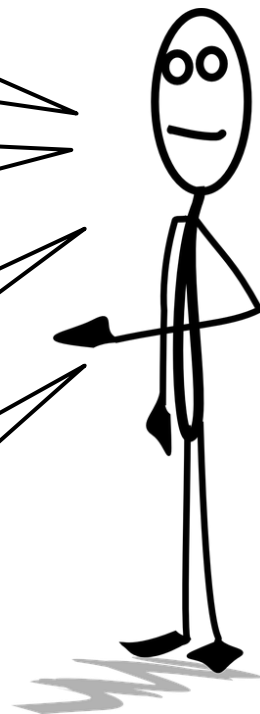
Verfahren:

Sortiere alle Elemente

Gehe alle Elemente in
dieser Reihenfolge durch.

Nimm alle der
Reihenfolge auf.

Das erste, das nicht mehr
passt, wird anteilig
gepackt.



Korrektheit, Laufzeit und Optimalität

Satz 1.5:

Algorithmus 1.4 liefert eine optimale Lösung für Problem 1.3 (Fractional Knapsack) in Zeit $O(n \log n)$.

Wir wollen drei Teile zeigen:

- (1) Laufzeit ist $O(n \log n)$.
- (2) Wir erhalten eine zulässige Lösung.
- (3) Es gibt keine bessere zulässige Lösung.

Dazu: ab an die Tafel!

Nächste Woche(n)

