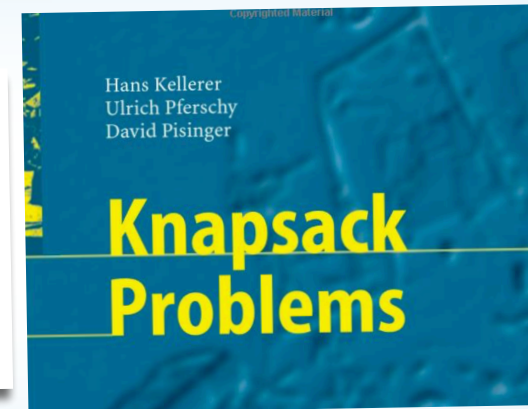




$$\begin{aligned} &\text{maximize} && \sum_{j=1}^n p_j x_j \\ &\text{subject to} && \sum_{j=1}^n w_j x_j \leq c, \\ &&& x_j = 0 \text{ or } 1, \quad j = 1, \dots, n, \end{aligned}$$



Algorithmen und Datenstrukturen 2

Sommer 2026

Prof. Dr. Sándor Fekete



Algorithmen und Datenstrukturen 2

Im Sommersemester 2026

[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:

Organisation

Vorlesung

Dienstag 9:45–11:15 Uhr, PK 11.2

Erste Vorlesung: 14.04.2026

Große Übung

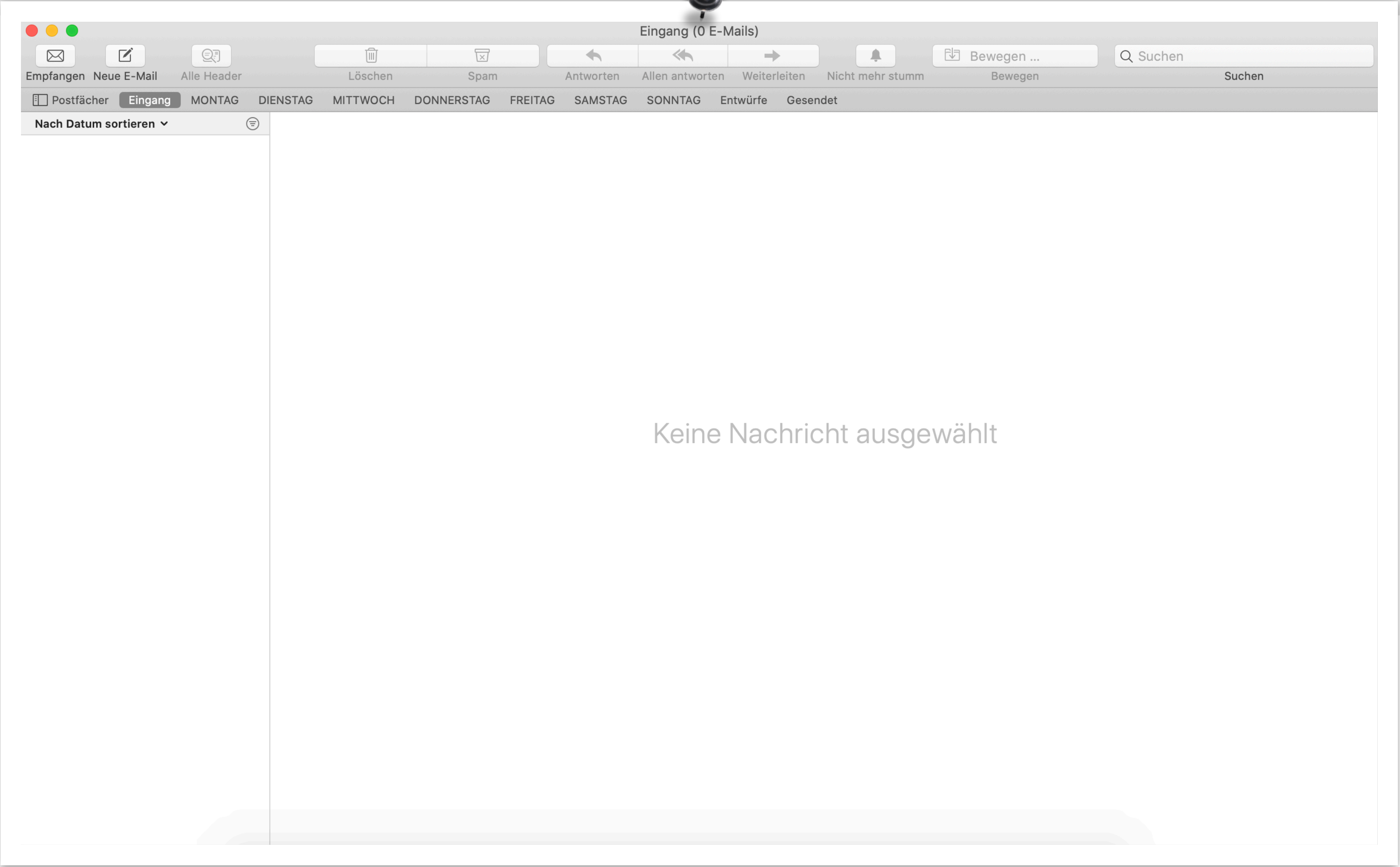
Mittwoch 15:00–16:30 Uhr, SN 19.1

Erste große Übung: 15.04.2026

s.fekete@tu-bs.de

Fragen von Ihnen?

kosfeld@ibr.tu-bs.de
loi@ibr.tu-bs.de



Eingang (0 E-Mails)

Empfangen

Neue E-Mail

Alle Header

Löschen

Spam

Antworten

Allen antworten

Weiterleiten

Nicht mehr

Bewegen ...

Suchen

Postfächer

Eingang

MONTAG

DIENTAG

MITTWOCH

DONNERSTAG

FREITAG

SAMSTAG

SONNTAG

Entwürfe

Gesendet

Nach Datum sortieren

mail.ibr.cs.tu-bs.de/mailman/listinfo/aud2

Aud2 --

About Aud2

To see the collection of prior postings to the list, visit the [Aud2 Archives](#). (The current archive is only available to the list members.)

Using Aud2

To post a message to all the list members, send email to aud2@ibr.cs.tu-bs.de.

You can subscribe to the list, or change your existing subscription, in the sections below.

Subscribing to Aud2

Subscribe to Aud2 by filling out the following form. You will be sent email requesting confirmation, to prevent others from gratuitously subscribing you. Once con approval by the list moderator. You will be notified of the moderator's decision by email. This is also a hidden list, which means that the list of members is availabl

Your email address:

Your name (optional):

You may enter a privacy password below. This provides only mild security, but should prevent others from messing with your subscription. **Do not use a valuable password** as it will occasionally be emailed back to you in cleartext.

If you choose not to enter a password, one will be automatically generated for you, and it will be sent to you once you've confirmed your subscription. You can always request a mail-back of your password when you edit your personal options.

Pick a password:

Reenter password to confirm:

Which language do you prefer to display your messages?

English (USA)

Subscribe



Technische
Universität
Braunschweig

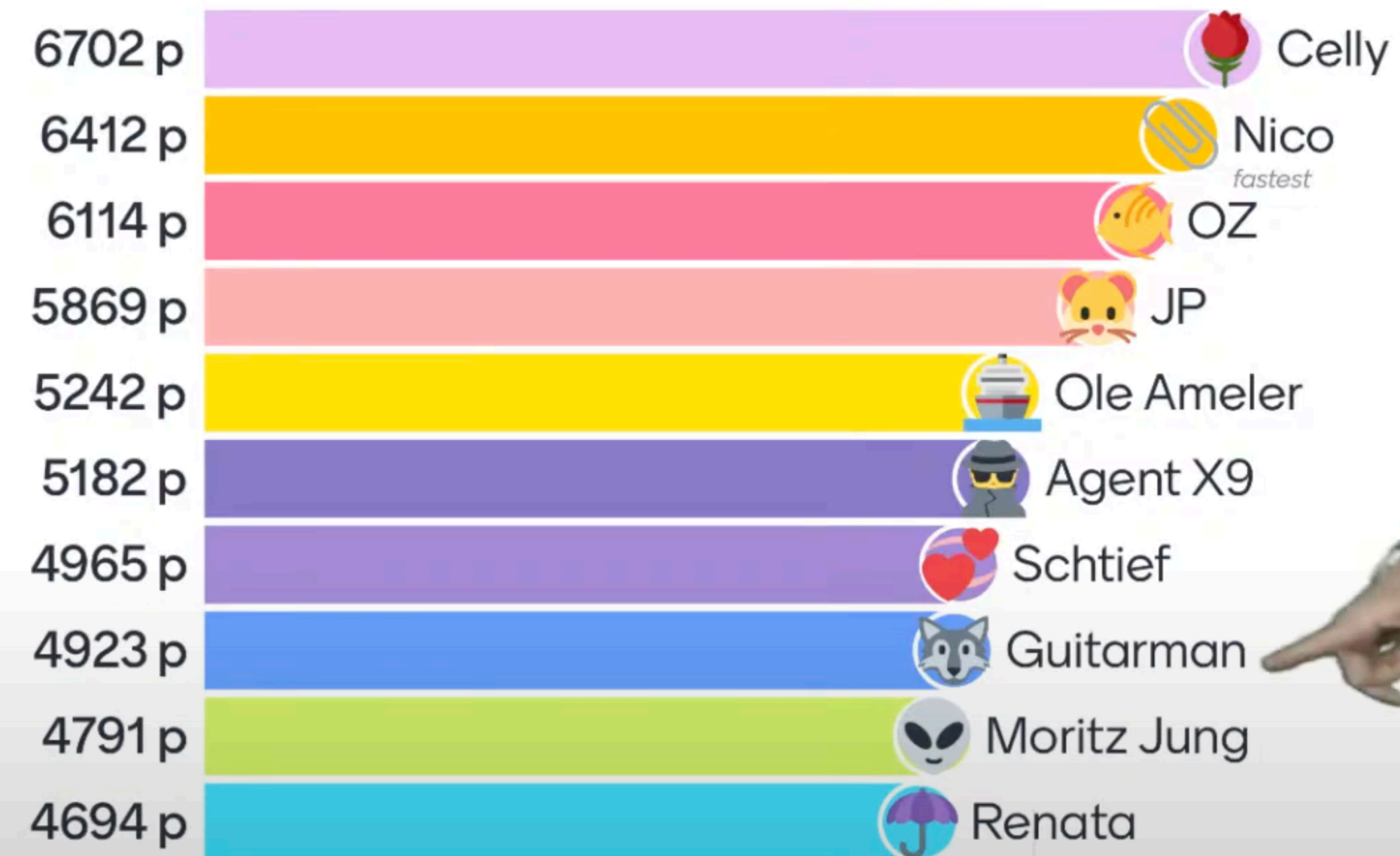
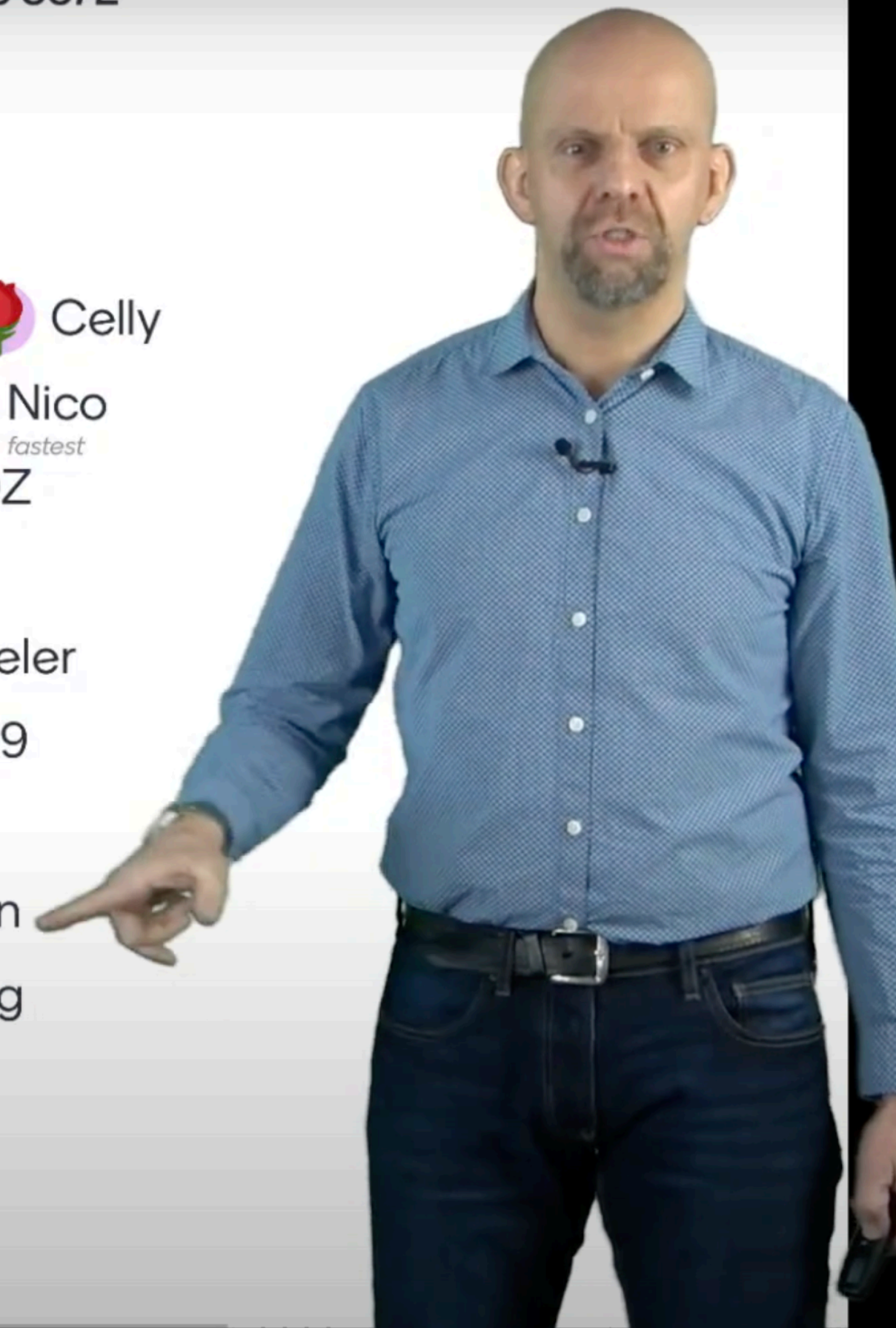
Sándor Fekete | Knapsack-Probleme | AuD2 2026

3

Fragen an Sie!

Go to www.menti.com and use the code 7858 8872

Leaderboard



Fragen an Sie!

Join at menti.com | use code **8943 4184**



Vorschau!

Geschäftsmodell

The Washington Post Moon helium deal is biggest purchase of natural resources from space

A salesperson must travel to 22 cities, visiting each city only once and returning to the first city in the most efficient route...a laptop computer would take about 1,000 years to figure out the best course. A quantum computer could one day compute the solution in minutes...

The Washington Post

Mostly sunny 37/27 • Tomorrow: Cloudy 47/33 [B6](#) *Democracy Dies in Darkness* MONDAY, FEBRUARY 5, 2018 • \$2

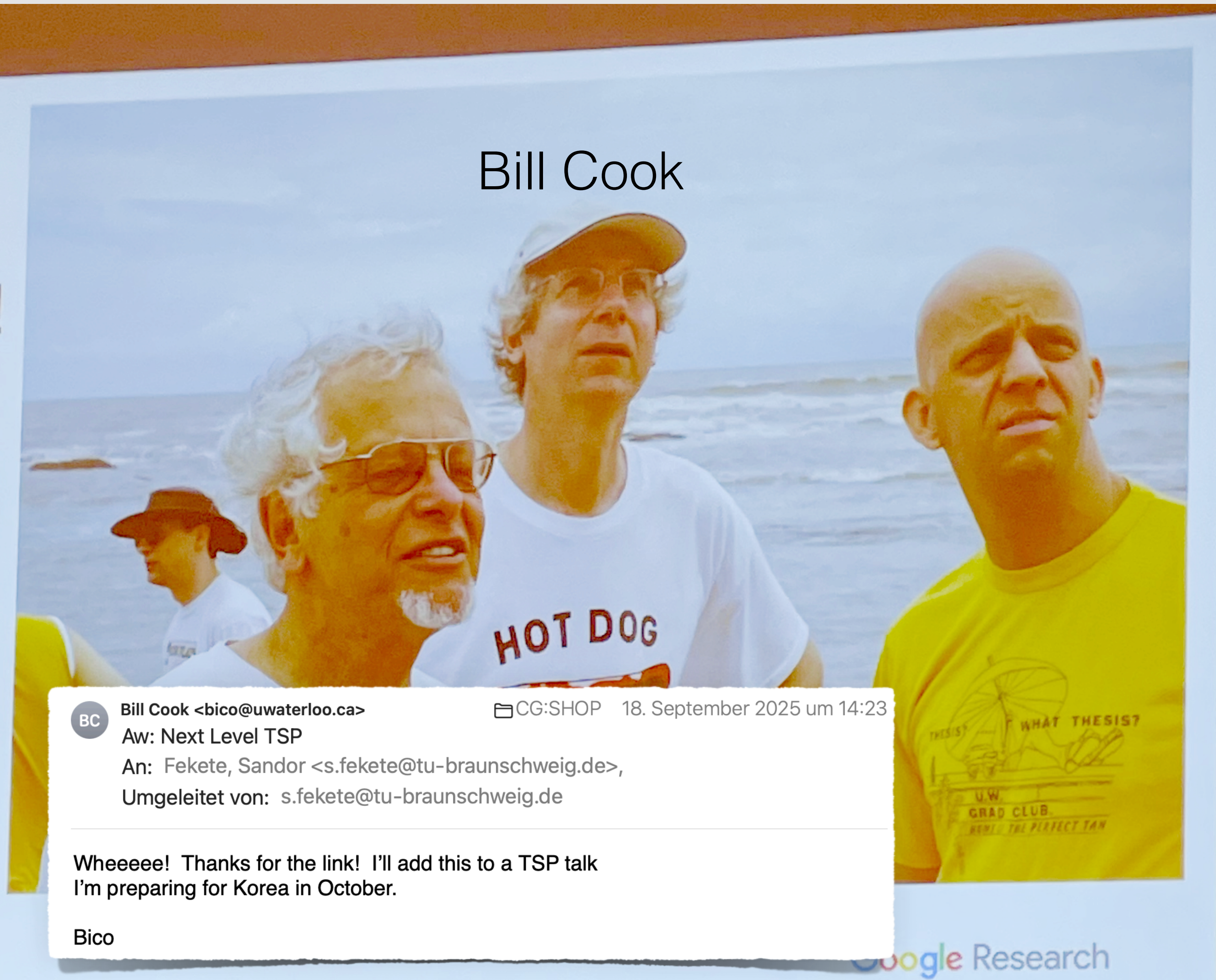
“It would take a laptop computer 1,000 years to compute the most efficient route between 22 cities, for example. A quantum computer could do this within minutes, possibly seconds.”

February 5, 2018

- (1) 300 Mio. einsammeln
- (2) Zum Mond fliegen
- (3) Helium-3 abbauen
- (4) Zur Erde bringen
- (5) Quantencomputer bauen
- (6) Beste Tour berechnen

Geschäftsmodell

Bill Cook



BC

Bill Cook <bico@uwaterloo.ca>

CG:SHOP 18. September 2025 um 14:23

Aw: Next Level TSP

An: Fekete, Sandor <s.fekete@tu-braunschweig.de>,

Umgeleitet von: s.fekete@tu-braunschweig.de

Wheeeee! Thanks for the link! I'll add this to a TSP talk
I'm preparing for Korea in October.

Bico

The Washington Post

Mostly sunny 37/27 • Tomorrow: Cloudy 47/33 [16](#) MONDAY, FEBRUARY 5, 2018 • \$2

“It would take a laptop computer 1,000 years to compute the most efficient route between 22 cities, for example. A quantum computer could do this within minutes, possibly seconds.”

February 5, 2018

- (1) 300 Mio. einsammeln
- (2) Zum Mond fliegen
- (3) Helium-3 abbauen
- (4) Zur Erde bringen
- (5) Quantencomputer bauen
- (6) Beste Tour berechnen

Geschäftsmodell

Bill Cook

The Washington Post

“It would take a laptop computer 1,000 years to compute the most efficient route between 22 cities, for example. A quantum computer could do this within minutes, possibly seconds.”

February 5, 2018



Bill Cook @wjcook · Feb 12

To clarify, computing an optimal TSP route on 22 points takes 0.005 seconds on my iMac, not the 1000 years reported in [@washingtonpost](#) An error factor of 6 trillion. Like reporting the US National Debt is \$4.
[washingtonpost.com/news/innovatio...](#)

BC

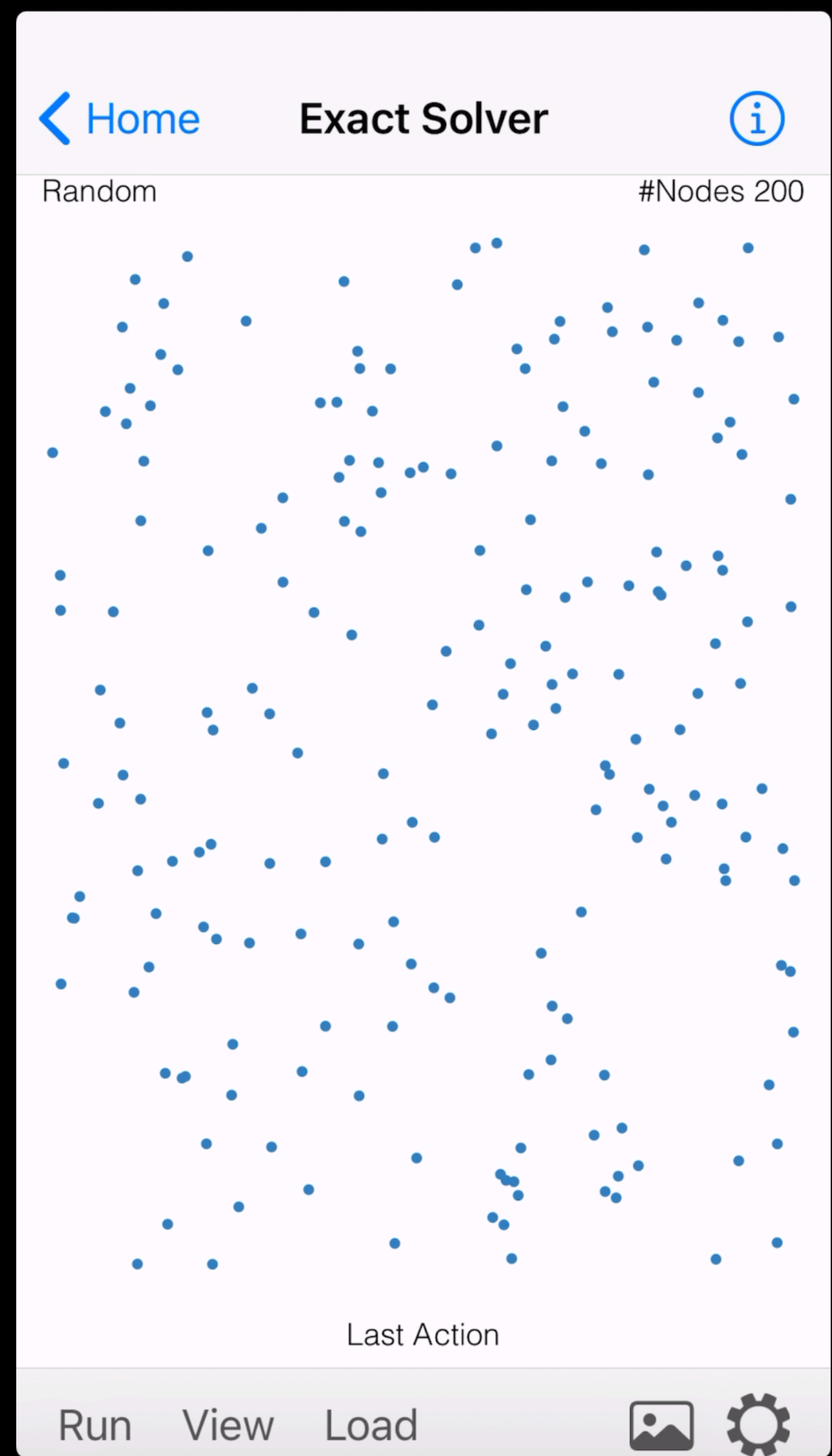
Bill Cook <bico@uwaterloo.ca>
Aw: Next Level TSP
An: Fekete, Sandor <s.fekete@tu-braunschweig.de>,
Umgeleitet von: s.fekete@tu-braunschweig.de

Wheeeee! Thanks for the link! I'll add this to a TSP talk
I'm preparing for Korea in October.

Bico

- (5) Quantencomputer bauen
- (6) Beste Tour berechnen

$$200! \approx 10^{370}$$





Carl-Friedrich-Gauß-Vorlesung 2026

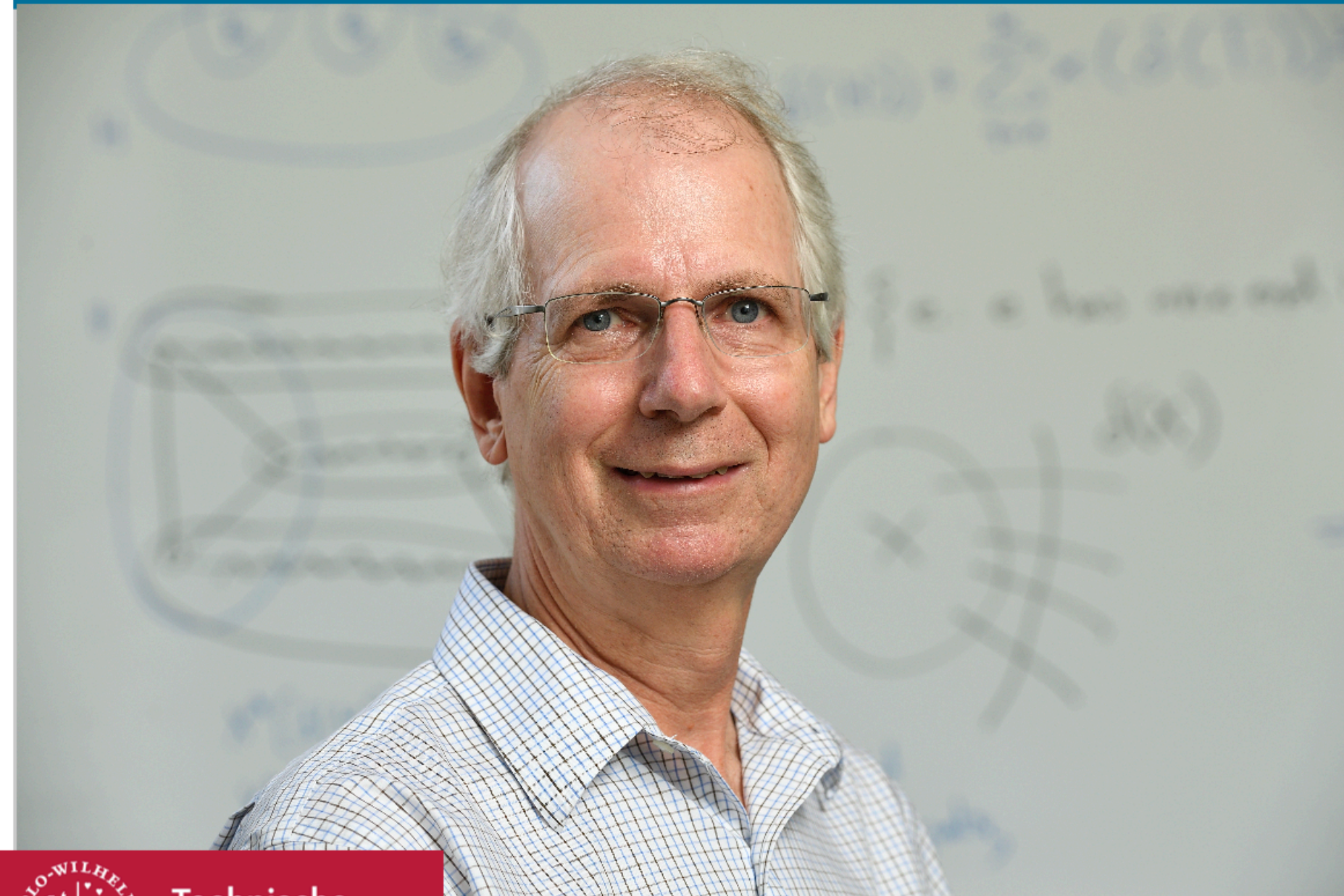
Prof. Dr. William J. Cook

University of Waterloo

**The Traveling Salesman Problem:
Package Deliveries, Pub Walks,
and Astro Tours**

Donnerstag, 07.05.2026, 17:00

Aula, Haus der Wissenschaft, Pockelsstr. 11

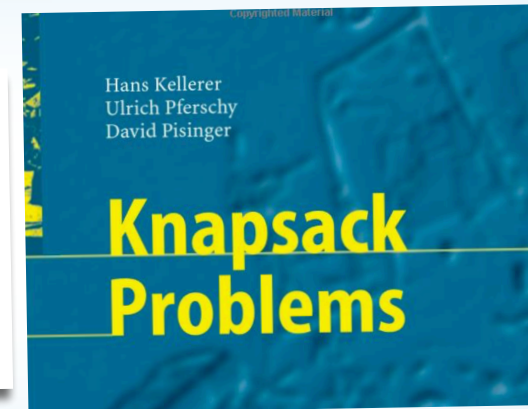


Technische
Universität
Braunschweig

Carl-Friedrich-Gauß-Fakultät



$$\begin{aligned} &\text{maximize} && \sum_{j=1}^n p_j x_j \\ &\text{subject to} && \sum_{j=1}^n w_j x_j \leq c, \\ &&& x_j = 0 \text{ or } 1, \quad j = 1, \dots, n, \end{aligned}$$



1 Einführung: Knapsack-Probleme

Algorithmen und Datenstrukturen 2
Sommer 2026

Prof. Dr. Sándor Fekete

Eine Klausursituation



Die Zeit läuft!



Knut Donald, erster Studierender der Informatik



- 150 Minuten
- 20 Aufgaben
- 100 Punkte
- 50 Punkte zum Bestehen
- Die Zeit läuft!

30 Minuten später:



- 6 sichere Punkte
- 10 hoffnungslose Punkte
- Noch 44 Punkte zum Bestehen
- Restliche Aufgaben...

Restliche Aufgaben

Diagram showing 13 stacks of blocks representing numbers, minutes, and points. The stacks are arranged in a grid-like pattern. The first three columns are labeled 'Nummer', 'Minuten', and 'Punkte' in red text. The stacks contain the following values:

Stack	Nummer	Minuten	Punkte
1	1	20	3
2	2	32	3
3	3	40	10
4	4	8	5
5	5	16	2
6	6	4	4
7	7	32	2
8	8	40	9
9	9	8	2
10	10	32	5
11	11	28	3
12	12	20	9
13	13	16	24
		4	

Kann Knut die Klausur bestehen?

Kann Knut die Klausur bestehen?

Problemstellung

Gegeben:

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

Zeitschranke: $Z = 120$

Punkteschranke: $P = 44$

Gesucht:

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

Sortieren?

nach Wert $\left(\frac{z_i}{p_i}\right)$ sortieren:

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

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

Sortieren?

nach Wert $\left(\frac{z_i}{p_i}\right)$ sortieren:

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

$$\sum_{i=1}^n z_i x_i = 4$$

$$\sum_{i=1}^n p_i x_i = 4$$

Sortieren?

nach Wert $\left(\frac{z_i}{p_i}\right)$ sortieren:

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

$$\sum_{i=1}^n z_i x_i = 12$$

$$\sum_{i=1}^n p_i x_i = 9$$

Sortieren?

nach Wert $\left(\frac{z_i}{p_i}\right)$ sortieren:

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

$$\sum_{i=1}^n z_i x_i = 28$$

$$\sum_{i=1}^n p_i x_i = 19$$

Sortieren?

nach Wert $\left(\frac{z_i}{p_i}\right)$ sortieren:

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

$$\sum_{i=1}^n z_i x_i = 48$$

$$\sum_{i=1}^n p_i x_i = 28$$

Sortieren?

nach Wert $\left(\frac{z_i}{p_i}\right)$ sortieren:

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

$$\sum_{i=1}^n z_i x_i = 56$$

$$\sum_{i=1}^n p_i x_i = 30$$

Sortieren?

nach Wert $\left(\frac{z_i}{p_i}\right)$ sortieren:

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

$$\sum_{i=1}^n z_i x_i = 96$$

$$\sum_{i=1}^n p_i x_i = 40$$

Sortieren?

nach Wert $\left(\frac{z_i}{p_i}\right)$ sortieren:

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

$$\sum_{i=1}^n z_i x_i = 120$$

$$\sum_{i=1}^n p_i x_i = 44$$

Knut kann die Klausur bestehen!

Problemdefinition

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

Gegeben:

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

Gesucht:

Eine Menge

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

mit

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

und

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

Optimierungsvariante

Problem 1.2' (MAXIMUM KNAPSACK).

Gegeben:

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

Gesucht:

Eine Menge

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

mit

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

und

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

Teilaufgaben?!

Problem 1.3 (FRACTIONAL KNAPSACK).

Gegeben:

- n Objekte $1, \dots, n$ mit jeweils Größe $z_i > 0$ 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

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

Algorithmus

Algorithmus 1.4. (*Greedy-Algorithmus*)

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**

Teilaufgaben?!

nach Wert $\left(\frac{z_i}{p_i}\right)$ sortieren:

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

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

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

Teilaufgaben?!

nach Wert $\left(\frac{z_i}{p_i}\right)$ sortieren:

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

$$\sum_{i=1}^n z_i x_i = 4$$

$$\sum_{i=1}^n p_i x_i = 4$$

Teilaufgaben?!

nach Wert $\left(\frac{z_i}{p_i}\right)$ sortieren:

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

$$\sum_{i=1}^n z_i x_i = 12$$

$$\sum_{i=1}^n p_i x_i = 9$$

Teilaufgaben?!

nach Wert $\left(\frac{z_i}{p_i}\right)$ sortieren:

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

$$\sum_{i=1}^n z_i x_i = 28$$

$$\sum_{i=1}^n p_i x_i = 19$$

Teilaufgaben?!

nach Wert $\left(\frac{z_i}{p_i}\right)$ sortieren:

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

$$\sum_{i=1}^n z_i x_i = 48$$

$$\sum_{i=1}^n p_i x_i = 28$$

Teilaufgaben?!

nach Wert $\left(\frac{z_i}{p_i}\right)$ sortieren:

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

$$\sum_{i=1}^n z_i x_i = 56$$

$$\sum_{i=1}^n p_i x_i = 30$$

Teilaufgaben?!

nach Wert $\left(\frac{z_i}{p_i}\right)$ sortieren:

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

$$\sum_{i=1}^n z_i x_i = 96$$

$$\sum_{i=1}^n p_i x_i = 40$$

Teilaufgaben?!

nach Wert $\left(\frac{z_i}{p_i}\right)$ sortieren:

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

$$\sum_{i=1}^n z_i x_i = 120$$

$$\sum_{i=1}^n p_i x_i = 46$$

$$x_{15} = 0,6$$

Algorithmus

Algorithmus 1.4. (*Greedy-Algorithmus*)

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**

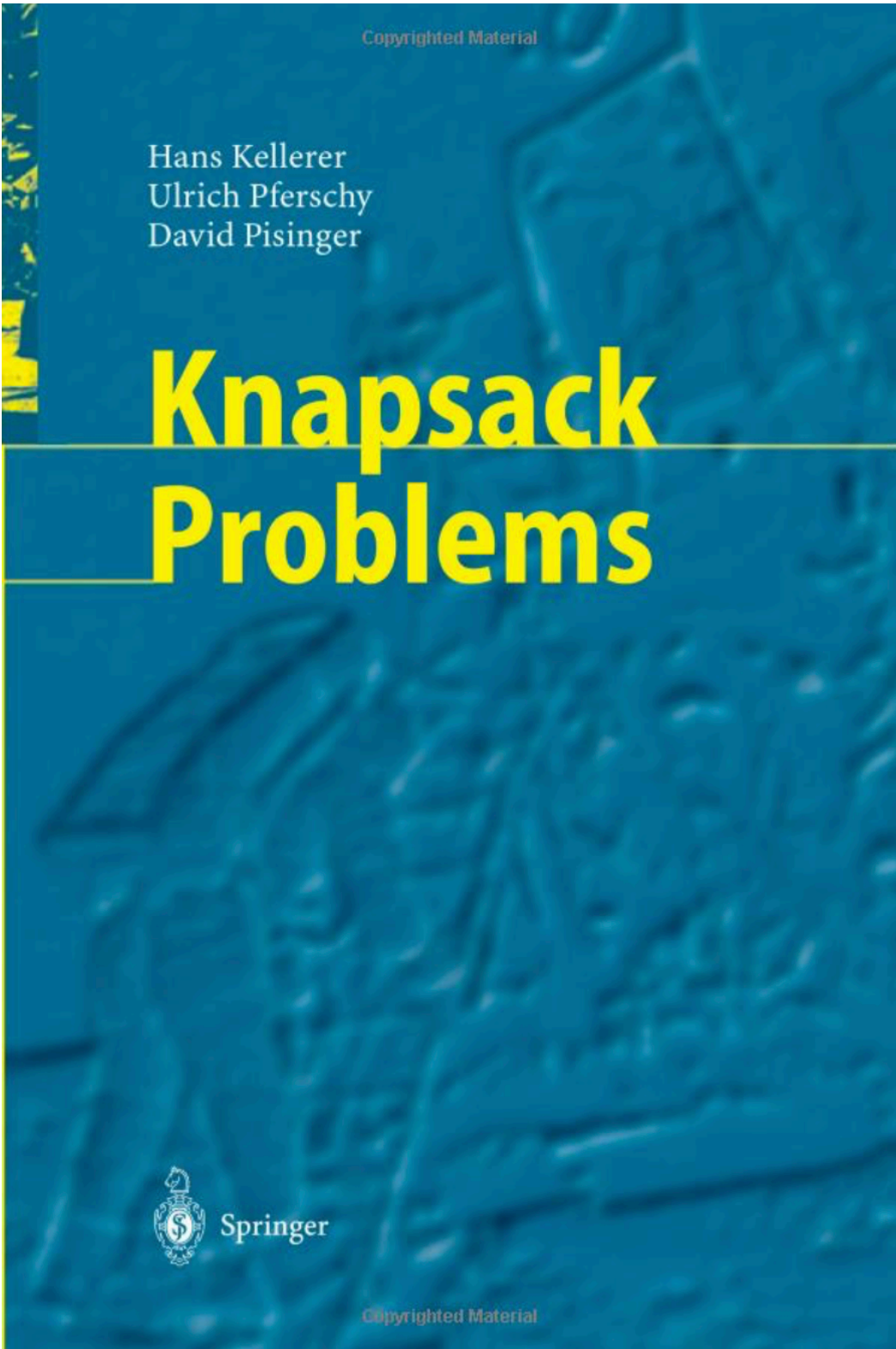
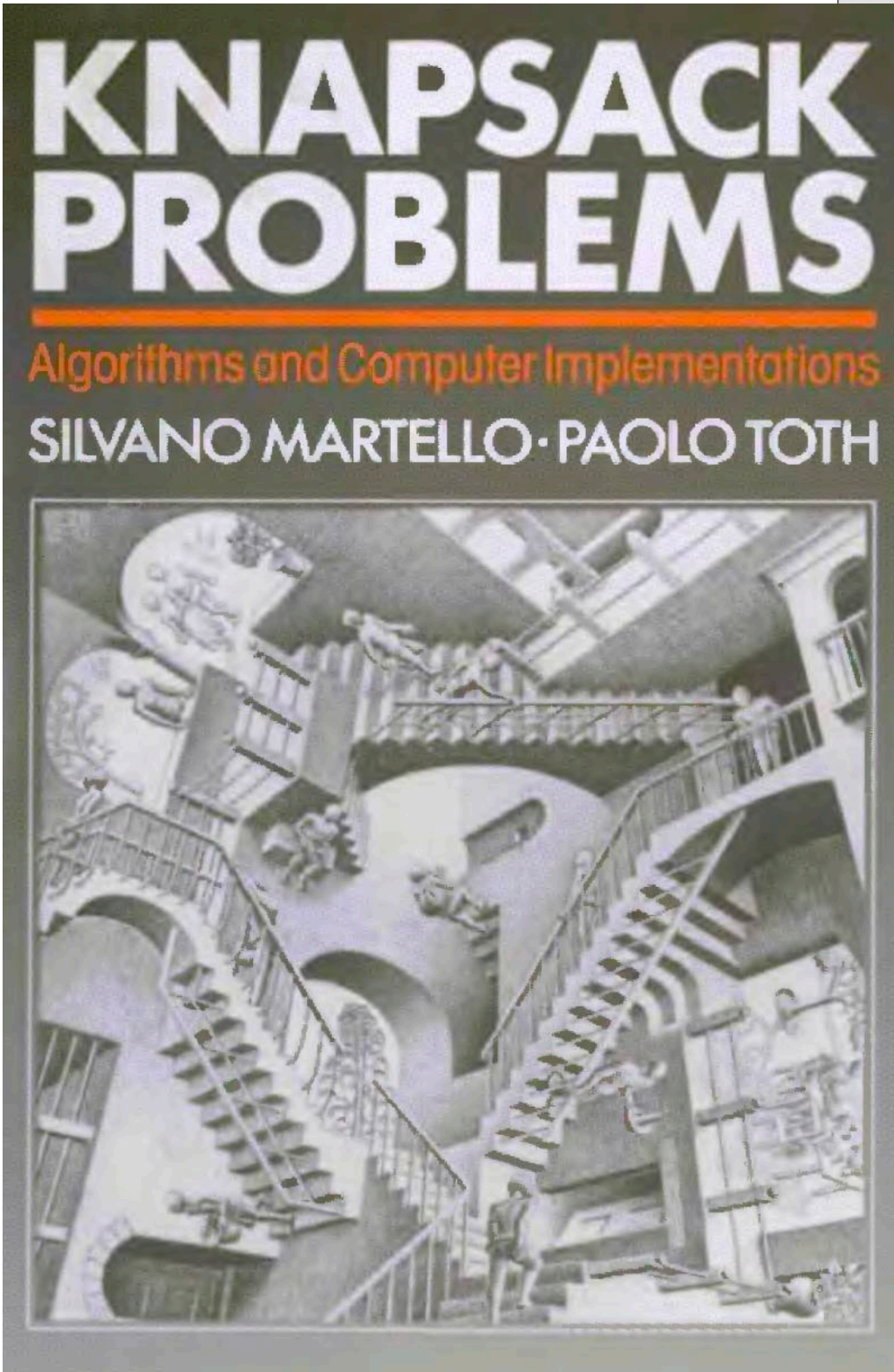
3: $x_{\pi(j)} := 1$

4: $j := j + 1$

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

6: **return**

Literatur



KnapsackProblems.pdf (Seite 5 von 306)

Hervorhebung Drehen Markierungen Suchen nach

Contents

Preface	xi
1 Introduction	1
1.1 What are knapsack problems?	1
1.2 Terminology	2
1.3 Computational complexity	6
1.4 Lower and upper bounds	9
2 0-1 Knapsack problem	13
2.1 Introduction	13
2.2 Relaxations and upper bounds	16
2.2.1 Linear programming relaxation and Dantzig's bound	16
2.2.2 Finding the critical item in $O(n)$ time	17
2.2.3 Lagrangian relaxation	19
2.3 Improved bounds	20
2.3.1 Bounds from additional constraints	20
2.3.2 Bounds from Lagrangian relaxations	23
2.3.3 Bounds from partial enumeration	24
2.4 The greedy algorithm	27

Literatur

Algorithmen und Datenstrukturen 2

Sándor P. Fekete

L^AT_EX Version: Arne Schmidt

21. Juni 2024

Vielen Dank!

s.fekete@tu-bs.de