Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Lehrstuhl für
INFORMATIK I
Algorithmen & Komplexität

Institut für Informatik

Algorithmische Graphentheorie

Sommersemester 2022

3. Vorlesung, Teil B

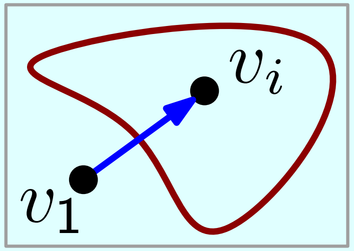
Dynamische Programmierung – Das Rucksackproblem

Prof. Dr. Alexander Wolff *Lehrstuhl für Informatik I*

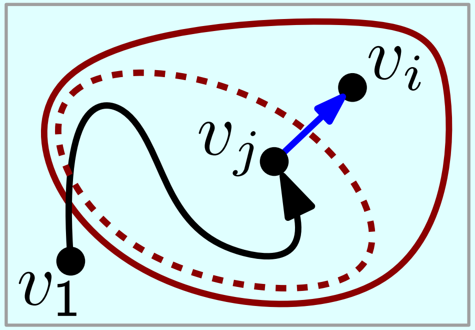
Das DP von Bellman & Held-Karp

BellmanHeldKarp(Knotenmenge V, Abstände $c\colon V \times V \to \mathbb{R}_{\geq 0}$)

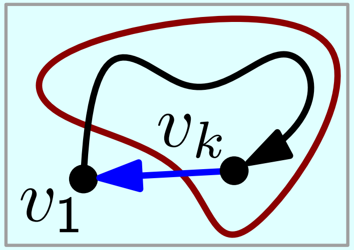
for $i = 2$ **to** n **do**
  $\mathrm{OPT}[\{v_i\}, v_i] = c(v_1, v_i)$



for $j = 2$ **to** $n - 1$ **do**
  foreach $W \subseteq \{v_2, \dots, v_n\}$ mit $|W| = j$ **do**
    foreach $v_i \in W$ **do**
      $\mathrm{OPT}[W, v_i] = \min_{v_j \in W \setminus \{v_i\}} \mathrm{OPT}[W \setminus \{v_i\}, v_j] + c(v_j, v_i)$



return $\min_{k \neq 1} \mathrm{OPT}[V \setminus \{v_1\}, v_k] + c(v_k, v_1)$



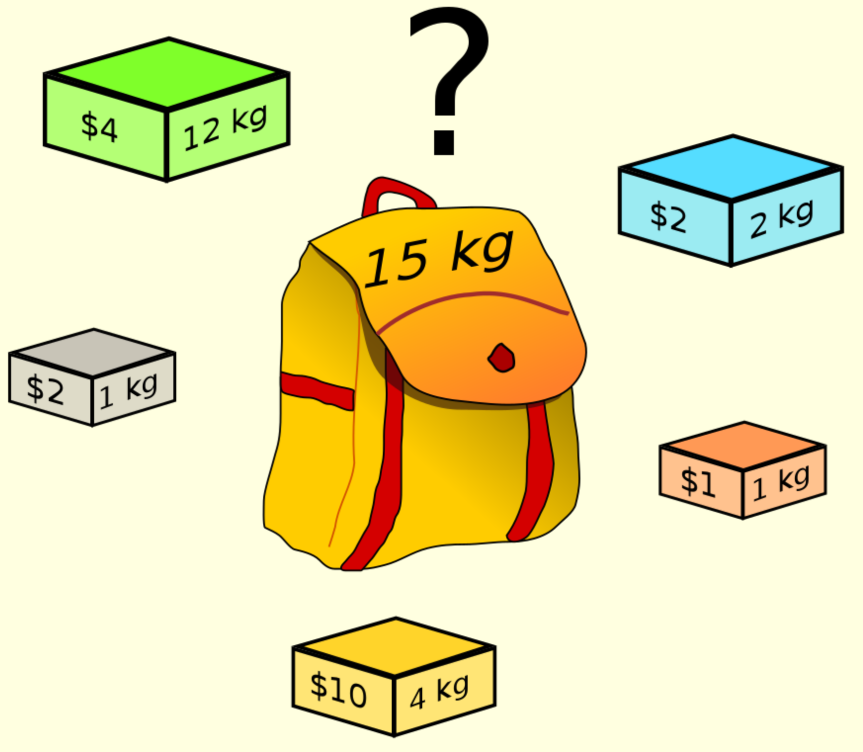
Wie viele Paare $\boxed{(W, v_i)}$ mit $v_i \in W$ gibt's? $\leq n \cdot 2^{n-1}$

Speicher: (= Größe der DP-Tabelle) $O\left(n \cdot 2^n\right)$

Das Rucksack-Problem

Gegeben: Menge U von Objekten und für jedes $i \in U$:
– ein Gewicht $w_i \in \mathbb{Q}^+$,
– ein Wert $v_i \in \mathbb{Q}^+$,
außerdem eine Gewichtsschranke $W > 0$.

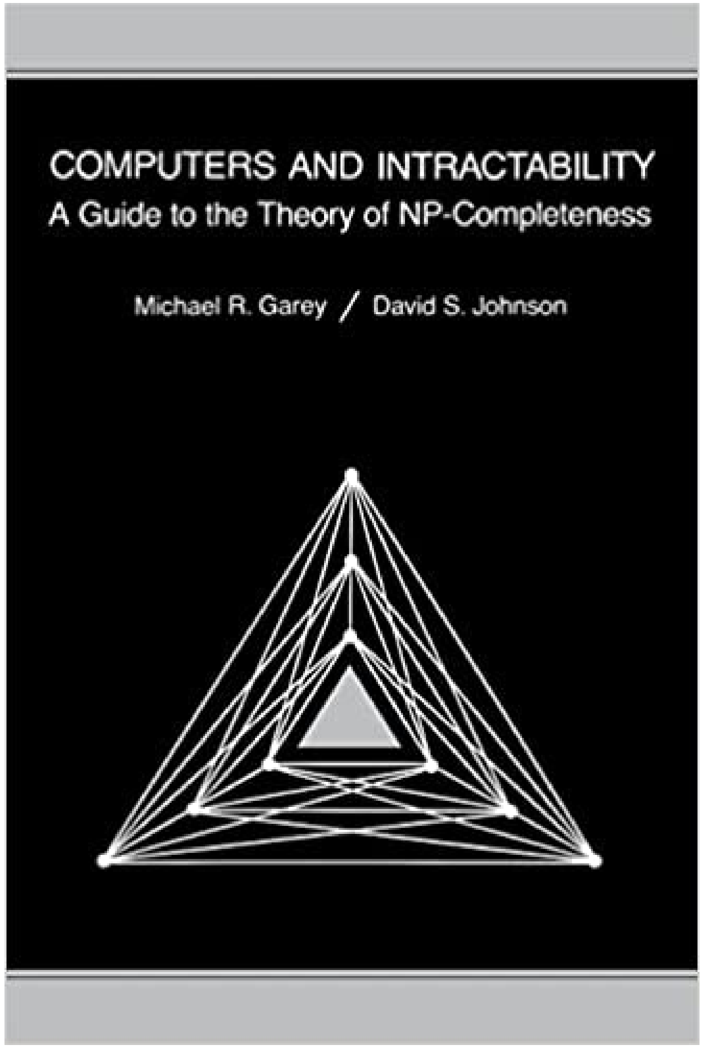
Gesucht: Teilmenge R von U, so dass
– das Gewicht der Objekte in R höchstens W ist:
$\sum_{i \in R} w_i \leq W$
– der Gesamtwert der Objekte in R maximal ist:
$\sum_{i \in R} v_i$ max!



Bad News

Satz. Das Rucksack-Problem ist NP-schwer.

Beweis. Durch Reduktion vom Problem ...



[siehe Buch von Garey & Johnson]

Was tun? – Mach das Problem leichter!

Gegeben: Menge U von Objekten und für jedes $i \in U$:

– ein Gewicht $w_i \in \cancel{\mathbb{Q}^+}$, $\mathbb{N}$

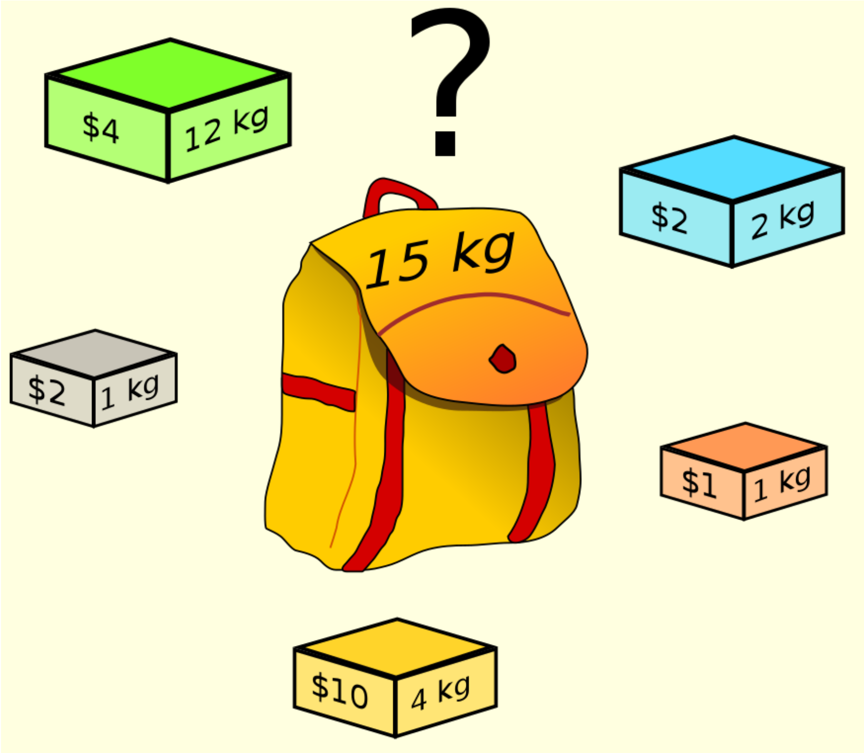
– ein Wert $v_i \in \mathbb{Q}^+$,

außerdem eine Gewichtsschranke $W > 0$.

Gesucht: Teilmenge R von U, so dass

– das Gewicht der Objekte in R höchstens W ist: $\sum_{i \in R} w_i \leq W$

– der Gesamtwert der Objekte in R maximal ist: $\sum_{i \in R} v_i$ max!



Rekursive Definition

Sei n die Anzahl der Objekte. O.B.d.A. sei $U = \{1, 2, \ldots, n\}$.

Sei $w \in \{1, \ldots, W\}$ und $1 \leq i \leq n$.

Idee: Def. $f(i, w) :=$ maximaler Wert von $\sum_{i \in R} v_i$, wobei $R \subseteq \{1, 2, \ldots, i\}$ und $\sum_{i \in R} w_i \leq w$.

Dann suchen wir $f(n, W)$.

Berechnung.

$$f(1, w) = \begin{cases} v_1 & \text{falls } w_1 \leq w, \\ 0 & \text{sonst.} \end{cases}$$

und für $2 \leq i \leq n$:

$$f(i, w) = \begin{cases} \max\{v_i + f(i-1, w - w_i), f(i-1, w)\} & \text{if } w_i \leq w, \\ f(i-1, w) & \text{else.} \end{cases}$$

Dynamisches Programm

Probieren Sie selbst, die Einträge der DP-Tabelle $f[i, w]$ in der richtigen Reihenfolge zu befüllen!

Ausblick

Master-Vorlesung „Advanced Algorithms“

Spannendes Video (30’) von Thomas van Dijk zu praktischen Aspekten der Implementierung eines Algorithmus für das Rucksack-Problem:

https://go.uniwue.de/algoprac4