The $\{0, 1\}$-knapsack problem with qualitative benefits

Bundesministerium für Bildung und Forschung

Grant No. 13N14561

L. Schäfer, T. Dietz, M. Barbati, S. Greco, J. Figueira, S. Ruzika

AG Optimization
TU Kaiserslautern

2020-10-09

Knapsack problem

Knapsack problem

value: 200€
weight: 2kg

value: 70€
weight: 2kg

4kg

value: 250€
weight: 3kg

value: 40€
weight: 1kg

Knapsack problem

4kg

value: 70€
weight: 2kg

value: 200€
weight: 2kg

value: 250€
weight: 3kg

value: 40€
weight: 1kg

Knapsack problem

Given:

- set of items $\mathcal{S} = \{s_1, \ldots, s_n\}$
- value function $v : \mathcal{S} \to \mathbb{Z}_+$
- weight function $w : \mathcal{S} \to \mathbb{Z}_+$
- knapsack capacity $W \in \mathbb{Z}_+$

Knapsack problem

Given:

- set of items $\mathcal{S} = \{s_1, \dots, s_n\}$
- value function $v : \mathcal{S} \to \mathbb{Z}_+$
- weight function $w : \mathcal{S} \to \mathbb{Z}_+$
- knapsack capacity $W \in \mathbb{Z}_+$

Task: Find a feasible subset $S^* \subseteq \mathcal{S}$ s.t. $w(S^*) \leq W$ and $v(S^*)$ maximal.

Knapsack problem

Given:

- set of items $\mathcal{S} = \{s_1, \ldots, s_n\}$
- value function $v : \mathcal{S} \to \mathbb{Z}_+$
- weight function $w : \mathcal{S} \to \mathbb{Z}_+$
- knapsack capacity $W \in \mathbb{Z}_+$

Task: Find a feasible subset $S^* \subseteq \mathcal{S}$ s.t. $w(S^*) \leq W$ and $v(S^*)$ maximal.

Facts:

- $\mathcal{NP}$-hard
- pseudo-polynomial time algorithm

Knapsack problem with qualitative benefits

Knapsack problem with qualitative benefits

4kg

Knapsack problem with qualitative benefits

importance: medium
weight: 2kg

importance: medium
weight: 2kg

4kg

importance: high
weight: 3kg

importance: low
weight: 1kg

Knapsack problem with qualitative benefits

importance: medium
weight: 2kg

4kg

importance: medium
weight: 2kg

importance: high
weight: 3kg

importance: low
weight: 1kg

Knapsack problem with qualitative benefits

importance: medium
weight: 2kg

importance: medium
weight: 2kg

4kg

importance: high
weight: 3kg

importance: low
weight: 1kg

Related literature

Knapsack problem

- Kellerer, H., Pferschy, U.,, Pisinger, D. (2004). Knapsack Problems. Springer, Berlin, Germany.
- Morabito, R., & Garcia, V. (1998). The cutting stock problem in a hardboard industry: A case study. Computers & Operations Research, 25(6), 469-485.
- Naldi, M., Nicosia, G., Pacifici, A., Pferschy, U., & Leder, B. (2016, November). A simulation study of fairness-profit trade-off in project selection based on HHI and knapsack models. In 2016 European Modelling Symposium (EMS) (pp. 85-90). IEEE.
- Choi, S., Park, S., & Kim, H. M. (2011). The Application of the 0-1 Knapsack problem to the load-shedding problem in microgrid operation. In Control and automation, and energy system engineering (pp. 227-234). Springer, Berlin, Heidelberg.
- ...

Related literature

Knapsack extensions

Related literature

Knapsack extensions

- Fuzzy approaches

- Kasperski, A., & Kulej, M. (2007). The 0-1 knapsack problem with fuzzy data. Fuzzy Optimization and Decision Making, 6(2), 163-172.
- Lin, F. T., & Yao, J. S. (2001). Using fuzzy numbers in knapsack problems. European Journal of Operational Research, 135(1), 158-176.
- . . .

Related literature

Knapsack extensions

- Fuzzy approaches

- Kasperski, A., & Kulej, M. (2007). The 0-1 knapsack problem with fuzzy data. Fuzzy Optimization and Decision Making, 6(2), 163-172.
- Lin, F. T., & Yao, J. S. (2001). Using fuzzy numbers in knapsack problems. European Journal of Operational Research, 135(1), 158-176.
- . . .

- Multiobjective approaches and applications

- Bazgan, C., Hugot, H., & Vanderpooten, D. (2009). Solving efficiently the 0–1 multi-objective knapsack problem. Computers & Operations Research, 36(1), 260-279.
- Saen, R. F. (2006). A decision model for technology selection in the existence of both cardinal and ordinal data. Applied Mathematics and computation, 181(2), 1600-1608.
- Keeney, R. L., & McDaniels, T. L. (1999). Identifying and structuring values to guide integrated resource planning at BC Gas. Operations Research, 47(5), 651-662.
- . . .

Setup

Given:

- set of items $\mathcal{S} = \{s_1, \ldots, s_n\}$

Setup

Given:

- set of items $\mathcal{S} = \{s_1, \ldots, s_n\}$




Setup

Given:

- set of items $\mathcal{S} = \{s_1, \dots, s_n\}$
- set of qualitative levels $\mathcal{L} = \{\ell_1, \dots, \ell_k\}$ with $\ell_i \prec \ell_{i+1}$

Setup

Given:

- set of items $\mathcal{S} = \{s_1, \ldots, s_n\}$
- set of qualitative levels $\mathcal{L} = \{\ell_1, \ldots, \ell_k\}$ with $\ell_i \prec \ell_{i+1}$
- k is a fixed parameter

Setup

Given:

- set of items $\mathcal{S} = \{s_1, \ldots, s_n\}$
- set of qualitative levels $\mathcal{L} = \{\ell_1, \ldots, \ell_k\}$ with $\ell_i \prec \ell_{i+1}$
- k is a fixed parameter
- rank function $r : \mathcal{S} \to \mathcal{L}$




Setup

Given:

- set of items $\mathcal{S} = \{s_1, \ldots, s_n\}$
- set of qualitative levels $\mathcal{L} = \{\ell_1, \ldots, \ell_k\}$ with $\ell_i \prec \ell_{i+1}$
- k is a fixed parameter
- rank function $r : \mathcal{S} \to \mathcal{L}$

medium

high

low

medium

Setup

Given:

- set of items $\mathcal{S} = \{s_1, \ldots, s_n\}$
- set of qualitative levels $\mathcal{L} = \{\ell_1, \ldots, \ell_k\}$ with $\ell_i \prec \ell_{i+1}$
- k is a fixed parameter
- rank function $r : \mathcal{S} \to \mathcal{L}$
- weight function $w : \mathcal{S} \to \mathbb{Z}_+$

medium

high

low

medium

Setup

Given:

- set of items $\mathcal{S} = \{s_1, \ldots, s_n\}$
- set of qualitative levels $\mathcal{L} = \{\ell_1, \ldots, \ell_k\}$ with $\ell_i \prec \ell_{i+1}$
- k is a fixed parameter
- rank function $r : \mathcal{S} \to \mathcal{L}$
- weight function $w : \mathcal{S} \to \mathbb{Z}_+$

medium
2kg

high
3kg

low
1kg

medium
2kg

Setup

Given:

- set of items $\mathcal{S} = \{s_1, \ldots, s_n\}$
- set of qualitative levels $\mathcal{L} = \{\ell_1, \ldots, \ell_k\}$ with $\ell_i \prec \ell_{i+1}$
- k is a fixed parameter
- rank function $r : \mathcal{S} \to \mathcal{L}$
- weight function $w : \mathcal{S} \to \mathbb{Z}_+$
- capacity $W \in \mathbb{Z}_+$

medium
2kg

high
3kg

low
1kg

medium
2kg

Setup

Given:

- set of items $\mathcal{S} = \{s_1, \dots, s_n\}$
- set of qualitative levels $\mathcal{L} = \{\ell_1, \dots, \ell_k\}$ with $\ell_i \prec \ell_{i+1}$
- k is a fixed parameter
- rank function $r : \mathcal{S} \to \mathcal{L}$
- weight function $w : \mathcal{S} \to \mathbb{Z}_+$
- capacity $W \in \mathbb{Z}_+$

Task: Find a feasible subset $S^* \subseteq \mathcal{S}$ s.t. $w(S^*) \leq W$ and $r(S^*)$ "maximal".

Setup

Given:

- set of items $\mathcal{S} = \{s_1, \ldots, s_n\}$
- set of qualitative levels $\mathcal{L} = \{\ell_1, \ldots, \ell_k\}$ with $\ell_i \prec \ell_{i+1}$
- k is a fixed parameter
- rank function $r : \mathcal{S} \to \mathcal{L}$
- weight function $w : \mathcal{S} \to \mathbb{Z}_+$
- capacity $W \in \mathbb{Z}_+$

Task: Find a feasible subset $S^* \subseteq \mathcal{S}$ s.t. $w(S^*) \leq W$ and $r(S^*)$ "maximal".

Definitions I

Definitions I

Numerical representation

- $v : \mathcal{L} \to \mathbb{Q}_+$ numerical representation w.r.t. $r : \mathcal{S} \to \mathcal{L}$ if

$$r(s_1) \succ r(s_2) \Leftrightarrow v(r(s_1)) > v(r(s_2)), \text{ for all } s_1, s_2 \in \mathcal{S} \text{ and}$$
$$r(s_1) \sim r(s_2) \Leftrightarrow v(r(s_1)) = v(r(s_2)), \text{ for all } s_1, s_2 \in \mathcal{S}$$

- $\mathcal{V}_r$: set of all numerical representations w.r.t. r

Definitions I

Numerical representation

- $v : \mathcal{L} \to \mathbb{Q}_+$ numerical representation w.r.t. $r : \mathcal{S} \to \mathcal{L}$ if

$$r(s_1) \succ r(s_2) \Leftrightarrow v(r(s_1)) > v(r(s_2)), \text{ for all } s_1, s_2 \in \mathcal{S} \text{ and}$$
$$r(s_1) \sim r(s_2) \Leftrightarrow v(r(s_1)) = v(r(s_2)), \text{ for all } s_1, s_2 \in \mathcal{S}$$

- $\mathcal{V}_r$: set of all numerical representations w.r.t. r

high | medium | low

 $\succ$  $\succ$ 

Definitions I

Numerical representation

- $v : \mathcal{L} \to \mathbb{Q}_+$ numerical representation w.r.t. $r : \mathcal{S} \to \mathcal{L}$ if

$$r(s_1) \succ r(s_2) \Leftrightarrow v(r(s_1)) > v(r(s_2)), \text{ for all } s_1, s_2 \in \mathcal{S} \text{ and}$$
$$r(s_1) \sim r(s_2) \Leftrightarrow v(r(s_1)) = v(r(s_2)), \text{ for all } s_1, s_2 \in \mathcal{S}$$

- $\mathcal{V}_r$: set of all numerical representations w.r.t. r

high		medium		low
	$\succ$		$\succ$	
$v(\text{high})$	$>$	$v(\text{medium})$	$>$	$v(\text{low})$

Definitions I

Rank cardinality function

- $g_i : 2^{\mathcal{S}} \to \mathbb{Z}_+$ with

$$g_i(S) = |\{s \in S \mid r(s) = \ell_i\}| \text{ for } i = 1, \dots, k$$

- rank cardinality vector: $g(S) = (g_1(S), \dots, g_k(S))^\top$
- value of $S \subseteq \mathcal{S}$: $v(S) = \ell_v \cdot g(S)$, $\ell_v = (v(\ell_1), \dots, v(\ell_k))$

Definitions I

Rank cardinality function

- $g_i : 2^{\mathcal{S}} \to \mathbb{Z}_+$ with

$$g_i(S) = |\{s \in S \mid r(s) = \ell_i\}| \text{ for } i = 1, \ldots, k$$

- rank cardinality vector: $g(S) = (g_1(S), \ldots, g_k(S))^\top$
- value of $S \subseteq \mathcal{S}$: $v(S) = \ell_v \cdot g(S)$, $\ell_v = (v(\ell_1), \ldots, v(\ell_k))$




Definitions I

Rank cardinality function

- $g_i : 2^{\mathcal{S}} \to \mathbb{Z}_+$ with

$$g_i(S) = |\{s \in S \mid r(s) = \ell_i\}| \text{ for } i = 1, \dots, k$$

- rank cardinality vector: $g(S) = (g_1(S), \dots, g_k(S))^\top$
- value of $S \subseteq \mathcal{S}$: $v(S) = \ell_v \cdot g(S)$, $\ell_v = (v(\ell_1), \dots, v(\ell_k))$

medium

high

low

medium

Definitions I

Rank cardinality function

- $g_i : 2^{\mathcal{S}} \to \mathbb{Z}_+$ with

$$g_i(S) = |\{s \in S \mid r(s) = \ell_i\}| \text{ for } i = 1, \ldots, k$$

- rank cardinality vector: $g(S) = (g_1(S), \ldots, g_k(S))^\top$
- value of $S \subseteq \mathcal{S}$: $v(S) = \ell_v \cdot g(S)$, $\ell_v = (v(\ell_1), \ldots, v(\ell_k))$

medium

high

low

medium

$$\mathbf{g(S) = (1, 2, 1)}$$

Definitions I

Rank cardinality function

- $g_i : 2^{\mathcal{S}} \to \mathbb{Z}_+$ with

$$g_i(S) = |\{s \in S \mid r(s) = \ell_i\}| \text{ for } i = 1, \dots, k$$

- rank cardinality vector: $g(S) = (g_1(S), \dots, g_k(S))^\top$
- value of $S \subseteq \mathcal{S}$: $v(S) = \ell_v \cdot g(S)$, $\ell_v = (v(\ell_1), \dots, v(\ell_k))$

medium
$v(\text{medium}) = 2$

high
$v(\text{high}) = 3$

low
$v(\text{low}) = 1$

medium
$v(\text{medium}) = 2$

Definitions I

Rank cardinality function

- $g_i : 2^{\mathcal{S}} \to \mathbb{Z}_+$ with

$$g_i(S) = |\{s \in S \mid r(s) = \ell_i\}| \text{ for } i = 1, \ldots, k$$

- rank cardinality vector: $g(S) = (g_1(S), \ldots, g_k(S))^\top$
- value of $S \subseteq \mathcal{S}$: $v(S) = \ell_v \cdot g(S)$, $\ell_v = (v(\ell_1), \ldots, v(\ell_k))$

medium
$v(\text{medium}) = 2$

high
$v(\text{high}) = 3$

low
$v(\text{low}) = 1$

medium
$v(\text{medium}) = 2$

$$\mathbf{v(S) = 1 \cdot 1 + 2 \cdot 2 + 3 \cdot 1 = 8}$$

Does it make sense?

importance: medium
weight: 2kg

importance: medium
weight: 2kg

4kg

importance: high
weight: 3kg

importance: low
weight: 1kg

Does it make sense?

$v_1(\text{medium}) = 2$
weight: 2kg

$v_1(\text{medium}) = 2$
weight: 2kg

4kg

$v_1(\text{high}) = 4$
weight: 3kg

$v_1(\text{low}) = 1$
weight: 1kg

Does it make sense?

$v_1(\text{medium}) = 2$
weight: 2kg

$v_1(\text{medium}) = 2$
weight: 2kg

4kg

$v_1(\text{high}) = 4$
weight: 3kg

$v_1(\text{low}) = 1$
weight: 1kg

Does it make sense?

4kg

$v_2(\text{medium}) = 3$
weight: 2kg

$v_2(\text{medium}) = 3$
weight: 2kg

$v_2(\text{high}) = 4$
weight: 3kg

$v_2(\text{low}) = 1$
weight: 1kg

Does it make sense?

4kg

$v_2(\text{medium}) = 3$
weight: 2kg

$v_2(\text{medium}) = 3$
weight: 2kg

$v_2(\text{high}) = 4$
weight: 3kg

$v_2(\text{low}) = 1$
weight: 1kg

Efficiency & Dominance

Efficiency & Dominance

Efficiency & Dominance

Efficiency & Dominance

- $S_1 \succeq S_2$ iff $v(S_1) \geq v(S_2)$ **for all** $v \in \mathcal{V}_r$

Efficiency & Dominance

Efficiency & Dominance

- $S_1 \succeq S_2$ iff $v(S_1) \geq v(S_2)$ **for all** $v \in \mathcal{V}_r$

medium

high

low

medium

Efficiency & Dominance

Efficiency & Dominance

- $S_1 \succeq S_2$ iff $v(S_1) \geq v(S_2)$ **for all** $v \in \mathcal{V}_r$

medium

$v(\text{medium}) = u$

high

$v(\text{high}) = v$

S_1

low

$v(\text{low}) = t$

medium

$v(\text{medium}) = u$

S_2

Efficiency & Dominance

Efficiency & Dominance

- $S_1 \succeq S_2$ iff $v(S_1) \geq v(S_2)$ **for all** $v \in \mathcal{V}_r$

We know: v > u > t

medium | high | low | medium

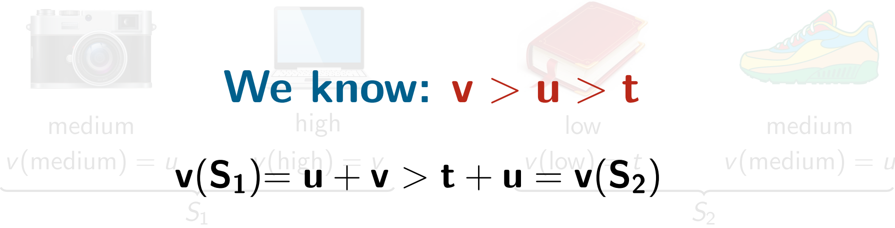
$v(\text{medium}) = u$ | $v(\text{high}) = v$ | $v(\text{low}) = t$ | $v(\text{medium}) = u$

S_1 | S_2

Efficiency & Dominance

Efficiency & Dominance

- $S_1 \succeq S_2$ iff $v(S_1) \geq v(S_2)$ **for all** $v \in \mathcal{V}_r$



We know: v > u > t

$$\mathbf{v(S_1) = u + v > t + u = v(S_2)}$$

Efficiency & Dominance

Efficiency & Dominance

- $S_1 \succeq S_2$ iff $v(S_1) \geq v(S_2)$ **for all** $v \in \mathcal{V}_r$
- $S_1 \succ S_2$ iff $S_1 \succeq S_2$ and $\exists v^* \in \mathcal{V}_r$ s.t. $v^*(S_1) > v^*(S_2)$

Efficiency & Dominance

Efficiency & Dominance

- $S_1 \succeq S_2$ iff $v(S_1) \geq v(S_2)$ **for all** $v \in \mathcal{V}_r$
- $S_1 \succ S_2$ iff $S_1 \succeq S_2$ and $\exists v^* \in \mathcal{V}_r$ s.t. $v^*(S_1) > v^*(S_2)$
- S^* efficient, if $\nexists S$ with $S \succ S^*$

Efficiency & Dominance

Efficiency & Dominance

- $S_1 \succeq S_2$ iff $v(S_1) \geq v(S_2)$ **for all** $v \in \mathcal{V}_r$
- $S_1 \succ S_2$ iff $S_1 \succeq S_2$ and $\exists v^* \in \mathcal{V}_r$ s.t. $v^*(S_1) > v^*(S_2)$
- S^* efficient, if $\nexists S$ with $S \succ S^*$
- $g(S^*)$ non-dominated rank cardinality vector

Efficiency & Dominance

Efficiency & Dominance

- $S_1 \succeq S_2$ iff $v(S_1) \geq v(S_2)$ **for all** $v \in \mathcal{V}_r$
- $S_1 \succ S_2$ iff $S_1 \succeq S_2$ and $\exists v^* \in \mathcal{V}_r$ s.t. $v^*(S_1) > v^*(S_2)$
- S^* efficient, if $\nexists S$ with $S \succ S^*$
- $g(S^*)$ non-dominated rank cardinality vector

Theorem

The dominance relation $\succeq$ defined on the set of feasible subsets of $\mathcal{S}$ is a preorder.

Main result

Main result

Theorem

$S_1 \succeq S_2$ if and only if $\sum_{i=j}^{k} g_i(S_1) \geq \sum_{i=j}^{k} g_i(S_2)$ for all $j = 1, \ldots, k$

Main result

Theorem

$S_1 \succeq S_2$ if and only if $\sum_{i=j}^{k} g_i(S_1) \geq \sum_{i=j}^{k} g_i(S_2)$ for all $j = 1, \ldots, k$

medium

high

S_1

low

medium

S_2

Main result

Theorem

$S_1 \succeq S_2$ if and only if $\sum_{i=j}^{k} g_i(S_1) \geq \sum_{i=j}^{k} g_i(S_2)$ for all $j = 1, \ldots, k$

medium

high

S_1

low

medium

S_2

$g(S_1) = (0, 1, 1)$

$g(S_2) = (1, 1, 0)$

Main result

Theorem

$S_1 \succeq S_2$ if and only if $\sum_{i=j}^{k} g_i(S_1) \geq \sum_{i=j}^{k} g_i(S_2)$ for all $j = 1, \ldots, k$

medium

high

low

medium

$$\underbrace{\text{medium} \quad \text{high}}_{S_1} \qquad \underbrace{\text{low} \quad \text{medium}}_{S_2}$$

$$g(S_1) = (0, 1, 1) \qquad g(S_2) = (1, 1, 0)$$

$$\mathbf{j = 1}: \quad \sum_{i=1}^{3} g_i(S_1) = 2 \geq 2 = \sum_{i=1}^{3} g_i(S_2) \quad \checkmark$$

Main result

Theorem

$S_1 \succeq S_2$ if and only if $\sum_{i=j}^{k} g_i(S_1) \geq \sum_{i=j}^{k} g_i(S_2)$ for all $j = 1, \ldots, k$


medium high

S_1


low medium

S_2

$g(S_1) = (0, 1, 1)$ $g(S_2) = (1, 1, 0)$

$$\mathbf{j = 1}: \quad \sum_{i=1}^{3} g_i(S_1) = 2 \geq 2 = \sum_{i=1}^{3} g_i(S_2) \quad \checkmark$$

$$\mathbf{j = 2}: \quad \sum_{i=2}^{3} g_i(S_1) = 2 \geq 1 = \sum_{i=2}^{3} g_i(S_2) \quad \checkmark$$

Main result

Theorem

$S_1 \succeq S_2$ if and only if $\sum_{i=j}^{k} g_i(S_1) \geq \sum_{i=j}^{k} g_i(S_2)$ for all $j = 1, \ldots, k$


medium


high

S_1


low


medium

S_2

$g(S_1) = (0, 1, 1)$

$g(S_2) = (1, 1, 0)$

$\mathbf{j = 1}: \quad \sum_{i=1}^{3} g_i(S_1) = 2 \geq 2 = \sum_{i=1}^{3} g_i(S_2) \quad \checkmark$

$\mathbf{j = 2}: \quad \sum_{i=2}^{3} g_i(S_1) = 2 \geq 1 = \sum_{i=2}^{3} g_i(S_2) \quad \checkmark$

$\mathbf{j = 3}: \quad \sum_{i=3}^{3} g_i(S_1) = 1 \geq 0 = \sum_{i=3}^{3} g_i(S_2) \quad \checkmark$

Main result

Theorem

$S_1 \succeq S_2$ if and only if $\sum_{i=j}^{k} g_i(S_1) \geq \sum_{i=j}^{k} g_i(S_2)$ for all $j = 1, \ldots, k$

Consequences:

Main result

Theorem

$S_1 \succeq S_2$ if and only if $\sum_{i=j}^{k} g_i(S_1) \geq \sum_{i=j}^{k} g_i(S_2)$ for all $j = 1, \ldots, k$

Consequences:

- dominance can be checked in constant time

Main result

Theorem

$S_1 \succeq S_2$ if and only if $\sum_{i=j}^{k} g_i(S_1) \geq \sum_{i=j}^{k} g_i(S_2)$ for all $j = 1, \ldots, k$

Consequences:

- dominance can be checked in constant time
- no need of numerical representations anymore

Definitions II

Definitions II

r-lexicographical order

- Sort items in non-increasing manner w.r.t. r
- in case of ties, take the item with lower weight first

Definitions II

r-lexicographical order

- Sort items in non-increasing manner w.r.t. r
- in case of ties, take the item with lower weight first

high
3kg

medium
2kg

medium
2kg

low
1kg

Definitions II

r-lexicographical order

- Sort items in non-increasing manner w.r.t. r
- in case of ties, take the item with lower weight first

w-lexicographical order

- Sort items in non-decreasing manner w.r.t. w
- in case of ties, take the item with higher rank first

Definitions II

r-lexicographical order

- Sort items in non-increasing manner w.r.t. r
- in case of ties, take the item with lower weight first

w-lexicographical order

- Sort items in non-decreasing manner w.r.t. w
- in case of ties, take the item with higher rank first

low
1kg

medium
2kg

medium
2kg

high
3kg

Greedy algorithm I

Greedy algorithm w.r.t. r

Algorithm

Greedy algorithm I

Greedy algorithm w.r.t. **r**

Algorithm

- Sort items r-lexicographically

Greedy algorithm I

Greedy algorithm w.r.t. r

Algorithm

- Sort items r-lexicographically
- pack items as long as they fit into the knapsack

Greedy algorithm I

Greedy algorithm w.r.t. r

Algorithm

- Sort items r-lexicographically
- pack items as long as they fit into the knapsack

high
3kg

medium
2kg

medium
2kg

low
1kg

Greedy algorithm I

Greedy algorithm w.r.t. r

Algorithm

- Sort items r-lexicographically
- pack items as long as they fit into the knapsack

high
3kg

medium
2kg

medium
2kg

low
1kg

Remaining capacity: 4kg

Greedy algorithm I

Greedy algorithm w.r.t. **r**

Algorithm

- Sort items r-lexicographically
- pack items as long as they fit into the knapsack

high
3kg

medium
2kg

medium
2kg

low
1kg

$S^* = \{$  $\}$

Remaining capacity: 1kg

Greedy algorithm I

Greedy algorithm w.r.t. r

Algorithm

- Sort items r-lexicographically
- pack items as long as they fit into the knapsack

high	medium	medium	low
3kg	2kg	2kg	1kg
			

$S^* = \{$  $\}$

Remaining capacity: 1kg

Greedy algorithm I

Greedy algorithm w.r.t. r

Algorithm

- Sort items r-lexicographically
- pack items as long as they fit into the knapsack

high	medium	medium	low
3kg	2kg	2kg	1kg

$S^* = \{$  $\}$

Remaining capacity: 1kg

Greedy algorithm I

Greedy algorithm w.r.t. r

Algorithm

- Sort items r-lexicographically
- pack items as long as they fit into the knapsack

high
3kg

medium
2kg

medium
2kg

low
1kg

$S^* = \{$  $\}$

Remaining capacity: 1kg

Greedy algorithm I

Greedy algorithm w.r.t. **r**

Algorithm

- Sort items r-lexicographically
- pack items as long as they fit into the knapsack

high 3kg

medium 2kg

medium 2kg

low 1kg

$S^* = \{$  ,  $\}$

Remaining capacity: 0kg

Greedy algorithm I

Greedy algorithm w.r.t. r

Algorithm

- Sort items r-lexicographically
- pack items as long as they fit into the knapsack

Theorem

The solution S^* returned by the algorithm is efficient.

Greedy algorithm I

Greedy algorithm w.r.t. r

Algorithm

- Sort items r-lexicographically
- pack items as long as they fit into the knapsack

Theorem

The solution S^* returned by the algorithm is efficient.

Theorem

The algorithm runs in $\mathcal{O}(n \log n)$.

Greedy algorithm II

Greedy algorithm w.r.t. **w**

Algorithm

Greedy algorithm II

Greedy algorithm w.r.t. **w**

Algorithm

- Sort items w-lexicographically

Greedy algorithm II

Greedy algorithm w.r.t. **w**

Algorithm

- Sort items *w*-lexicographically
- pack items as long as they fit into the knapsack

Greedy algorithm II

Greedy algorithm w.r.t. **w**

Algorithm

- Sort items *w*-lexicographically
- pack items as long as they fit into the knapsack

low
1kg

medium
2kg

medium
2kg

high
3kg

Greedy algorithm II

Greedy algorithm w.r.t. **w**

Algorithm

- Sort items *w*-lexicographically
- pack items as long as they fit into the knapsack

low	medium	medium	high
1kg	2kg	2kg	3kg

Remaining capacity: 4kg

Greedy algorithm II

Greedy algorithm w.r.t. **w**

Algorithm

- Sort items *w*-lexicographically
- pack items as long as they fit into the knapsack

low	medium	medium	high
1kg	2kg	2kg	3kg

$S^* = \{$  $\}$

Remaining capacity: 3kg

Greedy algorithm II

Greedy algorithm w.r.t. w

Algorithm

- Sort items w-lexicographically
- pack items as long as they fit into the knapsack

low	medium	medium	high
1kg	2kg	2kg	3kg
			

$S^* = \{$  $\}$

Remaining capacity: 3kg

Greedy algorithm II

Greedy algorithm w.r.t. **w**

Algorithm

- Sort items *w*-lexicographically
- pack items as long as they fit into the knapsack

low
1kg

medium
2kg


medium
2kg

high
3kg

$S^* = \{$  ,  $\}$

Remaining capacity: 1kg

Greedy algorithm II

Greedy algorithm w.r.t. **w**

Algorithm

- Sort items w-lexicographically
- pack items as long as they fit into the knapsack

low	medium	medium	high
1kg	2kg	2kg	3kg

$S^* = \{$  $\}$

Remaining capacity: 1kg

Greedy algorithm II

Greedy algorithm w.r.t. w

Algorithm

- Sort items w-lexicographically
- pack items as long as they fit into the knapsack

low	medium	medium	high
1kg	2kg	2kg	3kg
			

$S^* = \{$ ,  $\}$

Remaining capacity: 1kg

Greedy algorithm II

Greedy algorithm w.r.t. **w**

Algorithm

- Sort items *w*-lexicographically
- pack items as long as they fit into the knapsack

low
1kg

medium
2kg

medium
2kg

high
3kg

$S^* = \{$ ,  $\}$

Remaining capacity: 1kg

Greedy algorithm II

Greedy algorithm w.r.t. **w**

Algorithm

- Sort items *w*-lexicographically
- pack items as long as they fit into the knapsack

low	medium	medium	high
1kg	2kg	2kg	3kg


S^* is dominated

$S^* = \{$, $\}$

Remaining capacity: 1kg

Greedy algorithm II

Greedy algorithm w.r.t. **w**

Algorithm

- Sort items w-lexicographically
- pack items as long as they fit into the knapsack

Theorem

The solution S^* returned by the algorithm is efficient, if $w(S^*) = W$.

Greedy algorithm II

Greedy algorithm w.r.t. **w**

Algorithm

- Sort items w-lexicographically
- pack items as long as they fit into the knapsack

Theorem

The solution S^* returned by the algorithm is efficient, if $w(S^*) = W$.

Theorem

The algorithm runs in $\mathcal{O}(n \log n)$.

Dynamic programming

Algorithm

Dynamic programming

Algorithm

- $L_{i,x}$: rank cardinality vectors using first i items and total size smaller or equal to x

Dynamic programming

Algorithm

- $L_{i,x}$: rank cardinality vectors using first i items and total size smaller or equal to x
- $L_{0,x} = \emptyset$

Dynamic programming

Algorithm

- $L_{i,x}$: rank cardinality vectors using first i items and total size smaller or equal to x
- $L_{0,x} = \emptyset$
- $L_{i,x} \leftarrow \begin{cases} \max_{\succeq} \left\{ L_{i-1,x} \cup \left(g(\{s_i\}) \oplus L_{i-1,x-w(s_i)} \right) \right\}, & \text{if } w(s_i) \leq x \\ L_{i-1,x}, & \text{else.} \end{cases}$

Dynamic programming

x \ i	0	1	2	3	4
4					
3					
2					
1					
0					

medium
2kg

high
3kg

low
1kg

medium
2kg

Dynamic programming

4	$\emptyset$				
3	$\emptyset$				
2	$\emptyset$				
1	$\emptyset$				
0	$\emptyset$				
x / i	0	1	2	3	4

medium
2kg

high
3kg

low
1kg

medium
2kg

Dynamic programming

4	∅				
3	∅				
2	∅				
1	∅				
0	∅	∅			
x \ i	0	1	2	3	4

medium
2kg

high
3kg

low
1kg

medium
2kg

Dynamic programming

4	$\emptyset$				
3	$\emptyset$				
2	$\emptyset$				
1	$\emptyset$	$\emptyset$			
0	$\emptyset$	$\emptyset$			
x / i	0	1	2	3	4

medium
2kg

high
3kg

low
1kg

medium
2kg

Dynamic programming

4	∅				
3	∅				
2	∅	{}			
1	∅	∅			
0	∅	∅			
x / i	0	1	2	3	4

medium
2kg

high
3kg

low
1kg

medium
2kg

Dynamic programming

4	$\emptyset$				
3	$\emptyset$	{}			
2	$\emptyset$	{}			
1	$\emptyset$	$\emptyset$			
0	$\emptyset$	$\emptyset$			
x / i	0	1	2	3	4

medium
2kg

high
3kg

low
1kg

medium
2kg

Dynamic programming

4	$\emptyset$	{}			
3	$\emptyset$	{}			
2	$\emptyset$	{}			
1	$\emptyset$	$\emptyset$			
0	$\emptyset$	$\emptyset$			
x / i	0	1	2	3	4

medium
2kg

high
3kg

low
1kg

medium
2kg

Dynamic programming

4	$\emptyset$	{}			
3	$\emptyset$	{}			
2	$\emptyset$	{}			
1	$\emptyset$	$\emptyset$			
0	$\emptyset$	$\emptyset$	$\emptyset$		
x / i	0	1	2	3	4

medium
2kg

high
3kg

low
1kg

medium
2kg

Dynamic programming

4	$\emptyset$	{}			
3	$\emptyset$	{}			
2	$\emptyset$	{}			
1	$\emptyset$	$\emptyset$	$\emptyset$		
0	$\emptyset$	$\emptyset$	$\emptyset$		
x / i	0	1	2	3	4

medium
2kg

high
3kg

low
1kg

medium
2kg

Dynamic programming

4	$\emptyset$	{}			
3	$\emptyset$	{}			
2	$\emptyset$	{}	{}		
1	$\emptyset$	$\emptyset$	$\emptyset$		
0	$\emptyset$	$\emptyset$	$\emptyset$		
x / i	0	1	2	3	4

medium
2kg


high
3kg


low
1kg

medium
2kg

Dynamic programming

4	$\emptyset$	{  }			
3	$\emptyset$	{  }	{  }		
2	$\emptyset$	{  }	{  }		
1	$\emptyset$	$\emptyset$	$\emptyset$		
0	$\emptyset$	$\emptyset$	$\emptyset$		
x / i	0	1	2	3	4

medium
2kg


high
3kg


low
1kg

medium
2kg

Dynamic programming

x \ i	0	1	2	3	4
4	$\emptyset$	{}	{}		
3	$\emptyset$	{}	{}		
2	$\emptyset$	{}	{}		
1	$\emptyset$	$\emptyset$	$\emptyset$		
0	$\emptyset$	$\emptyset$	$\emptyset$		

medium
2kg

high
3kg

low
1kg

medium
2kg

Dynamic programming

4	$\emptyset$	{  }	{  }		
3	$\emptyset$	{  }	{  }		
2	$\emptyset$	{  }	{  }		
1	$\emptyset$	$\emptyset$	$\emptyset$		
0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	
x / i	0	1	2	3	4

medium
2kg


high
3kg


low
1kg


medium
2kg

Dynamic programming

4	$\emptyset$	{  }	{  }		
3	$\emptyset$	{  }	{  }		
2	$\emptyset$	{  }	{  }		
1	$\emptyset$	$\emptyset$	$\emptyset$	{  }	
0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	
x / i	0	1	2	3	4

medium
2kg


high
3kg


low
1kg


medium
2kg

Dynamic programming

4	$\emptyset$	$\{$$\}$	$\{$$\}$		
3	$\emptyset$	$\{$$\}$	$\{$$\}$		
2	$\emptyset$	$\{$$\}$	$\{$$\}$	$\{$$\}$	
1	$\emptyset$	$\emptyset$	$\emptyset$	$\{$$\}$	
0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	
x / i	0	1	2	3	4

medium
2kg

high
3kg

low
1kg

medium
2kg

Dynamic programming

4	$\emptyset$	{}	{}		
3	$\emptyset$	{}	{}	{}, {}	
2	$\emptyset$	{}	{}	{}	
1	$\emptyset$	$\emptyset$	$\emptyset$	{}	
0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	
x / i	0	1	2	3	4

medium
2kg


high
3kg


low
1kg


medium
2kg

Dynamic programming

4	∅	{}	{}	{}	
3	∅	{}	{}	{}, {}	
2	∅	{}	{}	{}	
1	∅	∅	∅	{}	
0	∅	∅	∅	∅	
x / i	0	1	2	3	4

medium
2kg

high
3kg

low
1kg

medium
2kg

Dynamic programming

4	$\emptyset$	{  }	{  }	{  }	
3	$\emptyset$	{  }	{  }	{  }, {  }	
2	$\emptyset$	{  }	{  }	{  }	
1	$\emptyset$	$\emptyset$	$\emptyset$	{  }	
0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
x / i	0	1	2	3	4

medium	high	low	medium
2kg	3kg	1kg	2kg
			

Dynamic programming

x \ i	0	1	2	3	4
4	$\emptyset$	{}	{}	{}	
3	$\emptyset$	{}	{}	{}, {}	
2	$\emptyset$	{}	{}	{}	
1	$\emptyset$	$\emptyset$	$\emptyset$	{}	{}
0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

medium
2kg

high
3kg

low
1kg

medium
2kg

Dynamic programming

4	$\emptyset$	{}	{}	{}	
3	$\emptyset$	{}	{}	{}, {}	
2	$\emptyset$	{}	{}	{}	{}, {}
1	$\emptyset$	$\emptyset$	$\emptyset$	{}	{}
0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
x \ i	0	1	2	3	4

medium	high	low	medium
2kg	3kg	1kg	2kg
			

Dynamic programming

4	$\emptyset$	{}	{}	{}	
3	$\emptyset$	{}	{}	{}, {}	{}, {}, {}
2	$\emptyset$	{}	{}	{}	{}, {}
1	$\emptyset$	$\emptyset$	$\emptyset$	{}	{}
0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
x / i	0	1	2	3	4

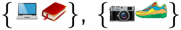
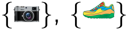
medium
2kg


high
3kg


low
1kg


medium
2kg


Dynamic programming

4	∅				
3	∅				
2	∅				
1	∅	∅	∅		
0	∅	∅	∅	∅	∅
x / i	0	1	2	3	4

medium	high	low	medium
2kg	3kg	1kg	2kg




Dynamic programming

Theorem

The algorithm correctly computes the set of non-dominated rank cardinality vectors.

Dynamic programming

Theorem

The algorithm correctly computes the set of non-dominated rank cardinality vectors.

Theorem

The number of labels in $L_{i,x}$ is polynomially bounded by $\mathcal{O}(i^k)$.

Dynamic programming

Theorem

The algorithm correctly computes the set of non-dominated rank cardinality vectors.

Theorem

The number of labels in $L_{i,x}$ is polynomially bounded by $\mathcal{O}(i^k)$.

Theorem

The algorithm runs in $\mathcal{O}(n^{2k+1}W)$.

Keep in mind

Keep in mind

- extension of the classical knapsack problem using qualitative benefits

Keep in mind

- extension of the classical knapsack problem using qualitative benefits
- check for dominance without consideration of numerical representations possible

Keep in mind

- extension of the classical knapsack problem using qualitative benefits
- check for dominance without consideration of numerical representations possible
- single non-dominated points can efficiently be found by greedy algorithms

Keep in mind

- extension of the classical knapsack problem using qualitative benefits
- check for dominance without consideration of numerical representations possible
- single non-dominated points can efficiently be found by greedy algorithms
- all non-dominated points can be found in pseudopolynomial time

Keep in mind

- extension of the classical knapsack problem using qualitative benefits
- check for dominance without consideration of numerical representations possible
- single non-dominated points can efficiently be found by greedy algorithms
- all non-dominated points can be found in pseudopolynomial time

Details: Schäfer, L. E., Dietz, T., Barbati, M., Figueira, J. R., Greco, S., & Ruzika, S. (2020). The binary knapsack problem with qualitative levels. European Journal of Operational Research.

Keep in mind

- extension of the classical knapsack problem using qualitative benefits
- check for dominance without consideration of numerical representations possible
- single non-dominated points can efficiently be found by greedy algorithms
- all non-dominated points can be found in pseudopolynomial time

Details: Schäfer, L. E., Dietz, T., Barbati, M., Figueira, J. R., Greco, S., & Ruzika, S. (2020). The binary knapsack problem with qualitative levels. European Journal of Operational Research.

luca.schaefer@mathematik.uni-kl.de