The binary knapsack problem with qualitative levels

Luca E. Schäfer\textsuperscript{a,∗}, Tobias Dietz\textsuperscript{a}, Maria Barbati\textsuperscript{b}, José Rui Figueira\textsuperscript{c}, Salvatore Greco\textsuperscript{b,d}, Stefan Ruzika\textsuperscript{a}

\textsuperscript{a}Department of Mathematics, Technische Universität Kaiserslautern, 67663 Kaiserslautern, Germany
\textsuperscript{b}Department of Economics and Business, University of Catania, 95129 Catania, Italy
\textsuperscript{c}CEG-IST, Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisboa, Portugal
\textsuperscript{d}Portsmouth Business School, Centre of Operations Research and Logistics (CORL), University of Portsmouth, PO1 3DE Portsmouth, United Kingdom

Abstract

A variant of the classical knapsack problem is considered in which each item is associated with an integer weight and a qualitative level. We define a dominance relation over the feasible subsets of the given item set and show that this relation defines a preorder. We propose a dynamic programming algorithm to compute the entire set of non-dominated rank cardinality vectors and we state two greedy algorithms, which efficiently compute a single efficient solution.

Keywords: Computing science, Knapsack problem, Non-dominance, Qualitative levels, Dynamic programming

\textsuperscript{∗}Corresponding author

Email addresses: luca.schaefer@mathematik.uni-kl.de (Luca E. Schäfer), dietz@mathematik.uni-kl.de (Tobias Dietz), maria.barbati@port.ac.uk (Maria Barbati), figureira@tecnico.ulisboa.pt (José Rui Figueira), salgreco@unict.it (Salvatore Greco), ruzika@mathematik.uni-kl.de (Stefan Ruzika)

July 21, 2020
1. Introduction

In the knapsack problem, each item has an associated profit and a weight value. The goal is to maximize the overall profit of the selected items under the constraint that the sum of the weights associated with the selected items does not exceed the knapsack capacity [12, 19].

The profit and the weight values of the items are usually assumed to be positive values and contribute to the definition of the objective function according to a quantitative evaluation. However, the profit (but also the weight values) of a knapsack problem can have a qualitative content [10]. For example, this is the case for a knapsack problem related to urban and territorial planning projects for which the profit expresses the overall environmental sustainability. In this context, it can be reasonable to use ordinal qualitative evaluations (such as “bad, medium, good”), because of the difficulty to assess numerical evaluations, which can be only apparently more precise and, instead, are always arbitrary to some extent.

**Example 1.1.** To motivate a knapsack setting with qualitative levels, we briefly propose an example in terms of a simple real life application. An external company has been asked to produce an environmental sustainability evaluation for a set of projects \( S = \{s_1, \ldots, s_5\} \). The evaluation of projects from \( S \) is given according to the following scale of qualitative levels \( \mathcal{L} = \{\ell_1 = Low, \ell_2 = Medium, \ell_3 = High, \ell_4 = Very High\} \). In detail, for each project the environmental sustainability has been evaluated as follows:

<table>
<thead>
<tr>
<th>Project</th>
<th>( s_1 )</th>
<th>( s_2 )</th>
<th>( s_3 )</th>
<th>( s_4 )</th>
<th>( s_5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>High</td>
<td>Very High</td>
</tr>
</tbody>
</table>
Consider two different managers, e.g., a quality manager and a production manager, who are in charge of deciding which projects to include in a portfolio. Due to some given budget and technical constraints, both managers focus their attention on three particular sets: \( S_1 = \{s_1, s_2, s_3\} \), \( S_2 = \{s_1, s_4\} \) and \( S_3 = \{s_2, s_5\} \). Considering sets \( S_3 \) and \( S_2 \), both managers can easily agree that the set of projects \( S_3 \) is preferred over the set of projects \( S_2 \), since project \( s_2 \) has a better evaluation than project \( s_1 \) and project \( s_5 \) has a better evaluation than project \( s_4 \). However, when considering sets \( S_1 \) and \( S_3 \), the comparison becomes more difficult. Thus, both managers associate a numerical evaluation with the qualitative levels \( L \) satisfying that a higher qualitative level corresponds to a higher numerical evaluation. Additionally, both managers agree on the sum of the numerical evaluations of each item in the portfolio as the evaluation criterion. However, the quality manager and the production manager, according to their own beliefs, propose two different numerical evaluations \( v' \) and \( v'' \), as follows:

<table>
<thead>
<tr>
<th>Evaluation</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
<th>Very High</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v' )</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>( v'' )</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

According to the numerical evaluation \( v' \) of the quality manager, it holds that \( v'(S_3) = 2 + 4 = 6 > 5 = 1 + 2 + 2 = v'(S_1) \) and thus, set \( S_3 \) should be preferred over set \( S_1 \). In contrast, considering the numerical evaluation \( v'' \) of the production manager, it holds that \( v''(S_3) = 4 + 6 = 10 < 11 = 3 + 4 + 4 = v'(S_1) \) and thus, set \( S_1 \) is preferred over the set \( S_3 \). Therefore, one can see that project preferences crucially depend on the particular numerical evaluation.
chosen, and it is certainly based on personal judgement. To overcome this issue, in this article, we introduce a preference system that does not rely on a particular numerical representation, instead it considers every possible numerical evaluation.

Note that the presence of such qualitative evaluations emerges in different contexts. For example, for the selection of research and development projects, it is important to consider not only the financial benefits of the projects but also their sustainability in terms of human and environmental values [27]. Similarly, when selecting infrastructures or urban planning projects their benefits, in terms of social aspects or environmental impact of the projects, can be hard to quantify [20]. Furthermore, when selecting technologies in a company the adoption of qualitative evaluations can be necessary given the scarcity of information to produce exact quantitative evaluations [22]. Therefore, given the plethora of potential applications, it is crucial to define a formulation of the knapsack problem with qualitative levels.

The paper is organized as follows. In Section 2, we review the literature on the knapsack problem with evaluations for the profit and the weights different from the classical knapsack problem. In Section 3, we define preliminaries and introduce the notation necessary to formulate the model. In Section 4, two greedy algorithms computing a single efficient solution and a dynamic programming procedure to list all the non-dominated rank cardinality vectors for the knapsack problem with qualitative levels are proposed. Finally, Section 5 concludes the paper.
2. Literature review

Although there exists a vast literature on the knapsack problem, a formulation that considers the qualitative benefits of the items is still missing. Nevertheless, the need to introduce imprecise or stochastic evaluations has been treated in different works, in which profits and weights associated with the items are different from a generic positive quantitative measure. Among those, fuzzy approaches are quite popular. In this sense, [17] formulated a fuzzy knapsack problem where the weight of each item is imprecise in the sense that it can be less than or greater than a fixed value. Later, they have proposed genetic algorithms to handle such problems [16]. Similarly, trapezoidal fuzzy intervals are used to model imprecision in weights and profits of the items, see [9]. Alike, fuzzy triangular numbers were exploited for defining the profits and the weights of the constrained knapsack problem, where a discount applies when a given quantity of an item is inserted in the knapsack, cf. [4].

Likewise, stochastic considerations have also been taken into account for the profits and the weights of items. For example, in [5] the profits are deterministic but the weights are independent random variables. The items are selected sequentially, and when inserted in the knapsack their size is determined. Then, the expected value of the profit is maximized. On the same perspective, [29] formulates the non-linear equality fractional knapsack problem where the profit associated with each item depends on the quantity that is inserted in the knapsack. Alternatively, [24] assumed that items could have a different profit value according to different scenarios. They have analyzed a set of feasible solutions for all the possible scenarios. Then,
the worst scenario, i.e., the knapsack maximizing the worst possible outcome, is identified.

Additionally, an alternative approach is to define a so-called parametric knapsack problem where the profits of the items are formulated as affine-linear functions of real-valued variables [8]. The authors proposed an approximation scheme in order to obtain the optimal solutions of the problem for all the values of the parameter within a given interval.

Further, the introduction of stochastic or imprecise weights and profits has been approached in a multicriteria formulation where the selection of items happens according not only to one single criterion but to more criteria, see e.g., [2]. In this sense, criteria with qualitative benefits are often employed for the selection of a set (portfolio) of items (projects) [23]. In this context, there are two main approaches. One is to assign a numerical evaluation to qualitative benefits through elicitation of a value function, cf. [11]. Another approach is to assign projects to ordered classes and select the projects in the best classes subject to some given constraints and requirements [26]. Therefore, in the former case, after defining the value function, the problem can be reformulated in terms of the classical knapsack problem, while in the latter case, an approach quite different from the classical knapsack formulation is adopted. For an interesting extension of the former approach see [13], which describes a stochastic multicriteria acceptability analysis for group decision making with the aim of ranking and selecting a set of waste management projects optimizing both quantitative and qualitative criteria. Later, the evaluations on the criteria were modeled through score intervals which are assumed to include the true value and all the non-dominated portfolios were
computed by means of a preference programming algorithm [14]. Following that, some modifications on the interdependencies of the projects were added in [15].

Besides, fuzzy numbers were used in a multicriteria context as in [21] where a fuzzy weighted average approach is employed to select new product development projects. Alternatively, in [3], the fuzziness of the profits for the projects are expressed through the definition of a data envelopment analysis approach. Moreover, the use of ordered weighted averaging operators is endorsed to deal with the vagueness of the contribution of several research funding programs in [28].

Finally, a promising approach strongly related to the qualitative optimization knapsack problem considered in this paper, was introduced in [1] where the evaluations of the items are linked to a set of predefined qualitative benefit levels. More precisely, an item can attain a certain benefit level according to the fact that its evaluation is above or below a given threshold; in this way, the evaluation of the items becomes an ordinal evaluation. Then, the preferred knapsack is obtained by means of a multiobjective optimization problem whose objectives to optimize are the number of items that attain the considered benefit levels. The differences with respect to the approach presented in this paper are mainly the following three: First, while in [1] a multiobjective knapsack problem, cf. [18], is considered, we consider in this paper a single objective knapsack problem. Second, in [1] a single “best” solution is searched through interaction with the decision maker, while in our approach we look for the entire set of solutions that are optimal with respect to numerical representations preserving the order of the ordinal evaluations.
Third, in [1], in the context of interactive multiobjective optimization, constraints can be added representing specific requirements expressing preferences of the decision maker in terms of minimum satisfaction levels on the considered objectives. In this paper, we consider simply the usual capacity constraint (which, of course, do not prevent to extend the approach we are proposing also in case of further constraints beyond the capacity).

From the above literature review, the necessity of dealing with imprecise or vague evaluations is evident. However, the adoption of ordinal evaluations in combinatorial optimization is very rare [25] and, to the best of our knowledge, a formulation of a knapsack problem with qualitative evaluations has not yet been provided. Our paper addresses this gap and, therefore, further expands the potential applications of the knapsack problem.

3. Preliminaries and Notation

Let $S = \{s_1, \ldots, s_n\}$ denote a set of items, let $L = \{\ell_1, \ldots, \ell_k\}$ denote a set of qualitative levels with $\ell_1 < \ell_2 < \cdots < \ell_k$ and let $W \in \mathbb{N}$ denote the knapsack capacity. With $\ell_i < \ell_{i+j}$, $i = 1, \ldots, k-1$ and $j = 1, \ldots, k-i$, we indicate that $\ell_{i+j}$ is strictly better than $\ell_i$ for all $j = 1, \ldots, k-i$. Furthermore, let $w: S \to \mathbb{N}$ denote a function assigning a weight to each item $s_i \in S$ and $r: S \to L$ denote a rank function assigning a qualitative level to each item $s_i \in S$. Finally, $S(W) = \{S' \subseteq S \mid w(S') \leq W\}$ is used to denote the set of all feasible subsets of $S$ satisfying the capacity $W \in \mathbb{N}$, where $w(S') = \sum_{s \in S'} w(s)$.

For the purpose of item sets with qualitative levels, we recall some definitions of binary relations, cf. [6]. A binary relation on $L$ is a subset $R$ on $L \times L$. 

8
For any $\ell, \ell' \in \mathcal{L}$, $(\ell, \ell') \in \mathcal{R}$ can also be denoted as $\ell \mathcal{R} \ell'$.

**Definition 3.1** (Properties of a binary relation). A binary relation $\mathcal{R}$ on $\mathcal{L}$ is called

1) reflexive, if $(\ell, \ell) \in \mathcal{R}$ for all $\ell \in \mathcal{L}$

2) transitive, if $(\ell_1, \ell_2) \in \mathcal{R}$ and $(\ell_2, \ell_3) \in \mathcal{R}$ implies $(\ell_1, \ell_3) \in \mathcal{R}$ for all $\ell_1, \ell_2, \ell_3 \in \mathcal{L}$

3) antisymmetric, if $(\ell_1, \ell_2) \in \mathcal{R}$ and $(\ell_2, \ell_1) \in \mathcal{R}$ implies $\ell_1 = \ell_2$ for all $\ell_1, \ell_2 \in \mathcal{L}$

A binary relation $\mathcal{R}$ on $\mathcal{L}$ is called a preorder, if it is reflexive and transitive and it is called a partial order, if it is reflexive, transitive and antisymmetric. Given a preorder $\preceq$ on $\mathcal{L}$, two additional relations can be defined as follows.

\[ \ell_1 \prec \ell_2 :\iff \ell_1 \preceq \ell_2 \text{ and } \ell_2 \not\preceq \ell_1 \text{ (asymmetric part of $\preceq$) } \]

\[ \ell_1 \sim \ell_2 :\iff \ell_1 \preceq \ell_2 \text{ and } \ell_2 \preceq \ell_1 \text{ (symmetric part of $\preceq$) } \]

In the following, we use the subsequent definition of a numerical representation. For a survey on preference structures and their numerical representations see [7].

**Definition 3.2** (Numerical representation). Let $S \subseteq \mathcal{S}$ be a subset of the item set, $\mathcal{L}$ a set of qualitative levels and let $r : \mathcal{S} \to \mathcal{L}$ be a rank function. A function $v : \mathcal{L} \to \mathbb{Q}^+$ is called a numerical representation with respect to the rank function $r$ if

\[ r(s_1) > r(s_2) \iff v(r(s_1)) > v(r(s_2)), \text{ for all } s_1, s_2 \in S \text{ and} \]

\[ r(s_1) \sim r(s_2) \iff v(r(s_1)) = v(r(s_2)), \text{ for all } s_1, s_2 \in S. \]
Every numerical representation preserves the order of the rank function. Let \( V_r \) denote the set of all numerical representations with respect to \( r \).

**Definition 3.3** (Rank cardinality function). Let \( S \subseteq \mathcal{S} \) be a subset of the item set, let \( r : \mathcal{S} \to \mathcal{L} \) be a rank function and let \( v : \mathcal{L} \to \mathbb{Q}^+ \) be a numerical representation with \( k \) being the number of qualitative levels. The rank cardinality function \( g_i : 2^\mathcal{S} \to \mathbb{N} \) is given by

\[
g_i(S) = |\{ s \in S | r(s) = \ell_i \}| \quad \text{for} \quad i = 1, \ldots, k.
\]

and denotes the number of items in \( S \) with \( \ell_i \) being its qualitative level. We call \( g(S) := (g_1(S), \ldots, g_k(S))^\top \) the rank cardinality vector of \( S \subseteq \mathcal{S} \). Further, for \( S \subseteq \mathcal{S} \), we define \( v(S) := \ell_v \cdot g(S) \), where \( \ell_v := (v(\ell_1), \ldots, v(\ell_k)) \), which denotes the total value of \( S \) with respect to the numerical representation \( v \).

**Definition 3.4** (Efficiency/ Dominance). Let \( S_1, S_2 \in \mathcal{S}(W) \) be feasible subsets of the item set for some \( W \in \mathbb{N} \). Then,

1) \( S_1 \) weakly dominates \( S_2 \), denoted by \( S_1 \succeq S_2 \), if and only if for every \( v \in V_r \), it holds that \( v(S_1) \geq v(S_2) \).

2) \( g(S_1) \) weakly dominates \( g(S_2) \), denoted by \( g(S_1) \succeq g(S_2) \), if \( S_1 \succeq S_2 \),

3) \( S_1 \) dominates \( S_2 \), denoted by \( S_1 \succ S_2 \), if and only if \( S_1 \) weakly dominates \( S_2 \) and there exists \( v^* \in V_r \) such that \( v^*(S_1) > v^*(S_2) \).

4) \( S^* \in \mathcal{S}(W) \) is called efficient, if there does not exist any \( S \in \mathcal{S}(W) \) with \( S \succ S^* \),

5) \( g(S^*) \) is called non-dominated rank cardinality vector, if \( S^* \) is efficient.
By Definition 3.4, it becomes clear that we aim to avoid using a specific numerical representation to decide on the dominance relation between two item sets. Instead, we require a set $S_1$ to be at least as good as another set $S_2$ for every possible numerical representation $v \in V_r$ to weakly dominate $S_2$.

**Definition 3.5** (Equivalence). Let $S_1, S_2 \subseteq S$ be two subsets of the item set. $S_1$ and $S_2$ are called equivalent if and only if $v(S_1) = v(S_2)$ for all $v \in V_r$.

**Remark 3.6.** Note that we can rewrite the definition of dominance in the following way. $S_1$ dominates $S_2$ if and only if $S_1$ weakly dominates $S_2$ and $S_2$ does not weakly dominate $S_1$.

Further, $S_1$ and $S_2$ are equivalent, if $S_1 \succeq S_2$ and $S_1 \not\succ S_2$.

**Lemma 3.7.** The dominance relation $\succeq$ defined on the set of feasible subsets $S(W)$ for some $W \in \mathbb{N}$ is a preorder.

*Proof.* Obviously $\succeq$ is reflexive since $v(S) \geq v(S)$ for every $v \in V_r$ and all feasible subsets $S \in S(W)$. Further, $\succeq$ is transitive, since for $S_1, S_2, S_3 \in S(W)$ with $v(S_1) \geq v(S_2)$ and $v(S_2) \geq v(S_3)$ for every $v \in V_r$, it holds that $v(S_1) \geq v(S_3)$ due to Definition 3.3. Consequently, $S_1 \succeq S_3$. \hfill $\square$

4. **The binary knapsack problem with qualitative levels**

In this section, we introduce the knapsack problem with qualitative levels. An instance $I = (S, r, w, W)$ of that problem is given by a set of items $S$, a rank function $r$, a weight function $w$ and a knapsack capacity $W$. Given such an instance $I$, we aim to find all non-dominated rank cardinality vectors, see
Definition 3.4. Further, we assume the number of qualitative levels $k$ to be fixed.

Lemma 4.1. Let $S_1, S_2 \subseteq S(W)$ be two feasible subsets of the item set for some $W \in \mathbb{N}$. Then $S_1$ weakly dominates $S_2$, i.e., $S_1 \succeq S_2$, if and only if

$$\sum_{i=j}^k g_i(S_1) \geq \sum_{i=j}^k g_i(S_2) \text{ for all } j = 1, \ldots, k.$$  

Proof. Let $S_1, S_2 \subseteq S(W)$ for some $W \in \mathbb{N}$ and let $j^* \in \{1, \ldots, k\}$ with $\sum_{i=j^*}^k g_i(S_1) < \sum_{i=j^*}^k g_i(S_2)$. Further, we set $M = 4 \cdot |S| \cdot k$ and define the following numerical representation $v : L \rightarrow \mathbb{Q}_+$:

$$v(\ell_i) = \begin{cases} i + M & \text{if } i \geq j^*, \quad i = 1, \ldots, k, \\ i & \text{if } i < j^*. \end{cases} \quad (1)$$

To obtain that $S_1$ does not weakly dominate $S_2$, we observe that

$$\frac{M}{2} \geq \left( \sum_{i=j^*}^k g_i(S_2) + \sum_{i=1}^{j^*-1} g_i(S_1) \right) k > k \sum_{i=j^*}^k g_i(S_2) + (j^* - 1) \sum_{i=1}^{j^*-1} g_i(S_1),$$

where the last inequality follows from the fact that $j^* - 1 < k$. In particular it holds

$$\frac{M + k}{2} \geq k \sum_{i=j^*}^k g_i(S_2) + (j^* - 1) \sum_{i=1}^{j^*-1} g_i(S_1). \quad (2)$$
Then, it holds

\[ v(S_2) = \sum_{i=1}^{k} g_i(S_2) \ell_i \]

\[ = \sum_{i=1}^{j^*-1} g_i(S_2)i + \sum_{i=j^*}^{k} g_i(S_2)(i + M) \]

\[ \geq M \sum_{i=j^*}^{k} g_i(S_2) \]

\[ \geq M \sum_{i=j^*}^{k} g_i(S_2) + k \sum_{i=j^*}^{k} g_i(S_2) + (j^* - 1) \sum_{i=1}^{j^*-1} g_i(S_1) - \frac{M+k}{2} \]

\[ = (M + k) \left( \sum_{i=j^*}^{k} g_i(S_2) - \frac{1}{2} \right) + (j^* - 1) \sum_{i=1}^{j^*-1} g_i(S_1) \]

\[ > (M + k) \sum_{i=j^*}^{k} g_i(S_1) + (j^* - 1) \sum_{i=1}^{j^*-1} g_i(S_1) \]

\[ \geq v(S_1). \]

Hence, it follows that \( S_1 \not\subseteq S_2 \). Thus, we proved that \( S_1 \geq S_2 \) implies that

\[ \sum_{i=j}^{k} g_i(S_1) \geq \sum_{i=j}^{k} g_i(S_2) \] for all \( j = 1, \ldots, k \).

For the other direction, let \( \tilde{g}_i(S), S \subseteq S(W), i = 1, \ldots, k \), denote the number of items \( s \in S \) to which the rank function \( r \) assigns a level not smaller than \( \ell_i \), i.e.,

\[ \tilde{g}_i(S) = |\{ s \in S | r(s) \geq \ell_i \}| = \sum_{j=i}^{k} g_j(S). \]
Observe that $g_i(S) = \tilde{g}_i(S) - \tilde{g}_{i+1}(S)$ for all $i = 1, \ldots, k - 1$. Thus, we get

$$v(S) = \sum_{i=1}^{k} v(\ell_i)g_i(S) = \sum_{i=1}^{k-1} v(\ell_i)(\tilde{g}_i(S) - \tilde{g}_{i+1}(S)) + v(\ell_k)\tilde{g}_k(S)$$

$$= v(\ell_1)\tilde{g}_1(S) + \sum_{i=2}^{k} (v(\ell_i) - v(\ell_{i-1})) \tilde{g}_i(S).$$

(3)

Consequently, if $\sum_{i=j}^{k} g_i(S_1) \geq \sum_{i=j}^{k} g_i(S_2)$ for all $j = 1, \ldots, k$, for some $S_1, S_2 \in S(W)$ or equivalently, $\tilde{g}_j(S_1) \geq \tilde{g}_j(S_2)$ for all $j = 1, \ldots, k$, and using (3) and the fact that

$$0 \leq v(\ell_1) \leq \ldots v(\ell_{k-1}) \leq v(\ell_k) \text{ for all } v \in V,$$

we get that for all $v \in V$, it holds:

$$v(S_1) = v(\ell_1)\tilde{g}_1(S_1) + \sum_{i=2}^{k} (v(\ell_i) - v(\ell_{i-1})) \tilde{g}_i(S_1)$$

$$\geq$$

$$v(\ell_1)\tilde{g}_1(S_2) + \sum_{i=2}^{k} (v(\ell_i) - v(\ell_{i-1})) \tilde{g}_i(S_2) = v(S_2)$$

Thus, we proved that $\sum_{i=j}^{k} g_i(S_1) \geq \sum_{i=j}^{k} g_i(S_2)$ for all $j = 1, \ldots, k$ implies $S_1 \succeq S_2$, which concludes the proof.

\[\square\]

**Remark 4.2.** Lemma 4.1 provides a criterion to decide in constant time on the dominance relation between two feasible subsets of the item set without considering numerical representations. Thus, by using Lemma 4.1 we can treat the knapsack problem with qualitative levels in a purely qualitative manner without violating reasonable assumptions on the preference structure, see Definition 3.2.
Next, we show how to obtain a single efficient solution by running a greedy algorithm. Therefore, we need the following definition of a lexicographically ordered item set.

**Definition 4.3** (*r*-lexicographical order). Let $S' \subseteq S$ be a subset of the item set with $S' = \{s'_1, \ldots, s'_p\}$ and let $\pi : \{1, \ldots, p\} \to \{1, \ldots, p\}$ be a permutation. Then, the lexicographically ordered set $S'_{r\text{-lex}}$ with respect to $r$ is defined as $S'_{r\text{-lex}} = [s'_{\pi(1)}, \ldots, s'_{\pi(p)}]$ with $r(s'_{\pi(1)}) \geq \cdots \geq r(s'_{\pi(p)})$ and if $r(s'_{\pi(i)}) = r(s'_{\pi(i+1)})$, then $w(s'_{\pi(i)}) \leq w(s'_{\pi(i+1)})$ for $i = 1, \ldots, p - 1$.

**Definition 4.4** (*w*-lexicographical order). Let $S' \subseteq S$ be a subset of the item set with $S' = \{s'_1, \ldots, s'_p\}$ and let $\pi : \{1, \ldots, p\} \to \{1, \ldots, p\}$ be a permutation. Then the lexicographically ordered set $S'_{w\text{-lex}}$ with respect to $w$ is defined as $S'_{w\text{-lex}} = [s'_{\pi(1)}, \ldots, s'_{\pi(p)}]$ with $w(s'_{\pi(1)}) \leq \cdots \leq w(s'_{\pi(p)})$ and if $w(s'_{\pi(i)}) = w(s'_{\pi(i+1)})$, then $r(s'_{\pi(i)}) \succeq r(s'_{\pi(i+1)})$ for $i = 1, \ldots, p - 1$.

**Algorithm 1** Greedy Algorithm w.r.t. $r$ or $w$

**Input:** An instance $I = (S, r, w, W)$

**Output:** An efficient solution $S^* \subseteq S$

1: Sort items $s_i \in S$ *r*- or *w*-lexicographically, see Definition 4.3 or 4.4
2: $S^* \leftarrow \emptyset$
3: for $i = 1, \ldots, n$ do
4: \hspace{1em} if $w(s_i) \leq W$ then
5: \hspace{2em} $S^* \leftarrow S^* \cup \{s_i\}$
6: \hspace{2em} $W \leftarrow W - w(s_i)$
7: return $S^*$
Theorem 4.5. The solution \( S^* \) returned by Algorithm 1 with respect to \( r \) is efficient.

Proof. Let \( S' \in \mathcal{S}(W) \) be an arbitrary feasible subset of \( \mathcal{S} \). We show that \( S' \nless S^* \). Therefore, we distinguish two cases.

**Case 1:** \( g(S') = g(S^*) \).

It follows that \( v(S') = v(S^*) \) for all \( v \in \mathcal{V}_r \) and nothing remains to show.

**Case 2:** \( g(S') \neq g(S^*) \).

Let \( j^* \) be maximal such that \( g_j(S') \neq g_j(S^*) \). Thus, it holds that \( g_i(S') = g_i(S^*) \) and, due to construction of Algorithm 1 with respect to \( r \), \( w(S'_i) \geq w(S^*_i) \) for all \( i = j^* + 1, \ldots, k \), where \( S'_i := \{ s \in S' \mid r(s) = \ell_i \} \). The latter inequality follows from the fact that \( g_i(S') = g_i(S^*) \) and that Algorithm 1 with respect to \( r \) picks among all items in \( \mathcal{S} \) with rank \( \ell_i, i = j^* + 1, \ldots, k \), the items of least weight first. Note that \( S^*_i \) is analogously defined. Consequently, it follows that

\[
g_{j^*}(S') < g_{j^*}(S^*). \tag{4}
\]

Next, we define a numerical representation \( v : \mathcal{L} \to \mathbb{Q}_+ \) as follows.

\[
v(\ell_i) = \begin{cases} 
\frac{1}{2^{j^*-i}} & \text{if } i < j^* \\
1 & \text{if } i = j^* \\
n + i & \text{if } i > j^*
\end{cases} \tag{5}
\]

It follows:
\[ v(S') = \sum_{i=1}^{k} g_i(S') v(\ell_i) \]

\[ = \sum_{i=1}^{j^*-1} g_i(S') \frac{1}{2^{j^*-i}} + g_{j^*}(S') n + \sum_{i=j^*+1}^{k} g_i(S')(n+i) \]

\[ < n(g_{j^*}(S') + 1) + \sum_{i=j^*+1}^{k} g_i(S')(n+i) \]

\[ = n(g_{j^*}(S') + 1) + \sum_{i=j^*+1}^{k} g_i(S')(n+i) \]

\[ \leq n \cdot g_{j^*}(S^*) + \sum_{i=j^*+1}^{k} g_i(S^*)(n+i) \]

\[ \leq \sum_{i=1}^{j^*-1} g_i(S^*) \frac{1}{2^{j^*-i}} + ng_{j^*}(S^*) + \sum_{i=j^*+1}^{k} g_i(S^*)(n+i) \]

\[ = v(S^*) \]

Consequently, it is \( S' \not\succ S^* \), which concludes the proof. \( \square \)

**Theorem 4.6.** Algorithm 1 with respect to \( r \) runs in \( O(n \log n) \).

**Proof.** The sorting of the items can be done in \( O(n \log n) \) time. The amount of work in the for-loop of Algorithm 1 with respect to \( r \) is in \( O(n) \), since only constant time operations are performed within each iteration. Thus, the running time follows. \( \square \)

**Theorem 4.7.** The solution \( S^* \) returned by Algorithm 1 with respect to \( w \) is efficient, if \( w(S^*) = W \).

**Proof.** Let \( S^* \) denote the solution returned by Algorithm 1 with respect to \( w \) with \( w(S^*) = W \). It holds that \( \sum_{i=1}^{k} g_i(S^*) \geq \sum_{i=1}^{k} g_i(S') \) for all \( S' \in \mathcal{S}(W) \). If \( \sum_{i=1}^{k} g_i(S^*) > \sum_{i=1}^{k} g_i(S') \), then there is nothing to show.
In case of $\sum_{i=1}^{k} g_i(S^*) = \sum_{i=1}^{k} g_i(S')$, it holds that $w(S') \geq w(S^*)$. If $w(S') > w(S^*) = W$, we know that $S' \notin S(W)$. In the case of $w(S') = w(S^*)$, we know due to construction of the algorithm that either $S'$ and $S^*$ are both efficient or that $S^*$ dominates $S'$. Consequently, it follows that $S^*$ is efficient.

Corollary 4.8. Algorithm 1 with respect to $w$ runs in $O(n \log n)$.

Example 4.9. To illustrate Algorithm 1 with respect to $r$ and $w$, consider the following knapsack instance $I = (S, r, w, W)$ with $S = \{s_1, s_2, s_3, s_4, s_5\}$ and $W = 6$. The corresponding item weights and qualitative levels are as follows:

<table>
<thead>
<tr>
<th>item</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>$r$</td>
<td>$\ell_1$</td>
<td>$\ell_2$</td>
<td>$\ell_2$</td>
<td>$\ell_3$</td>
<td>$\ell_4$</td>
</tr>
</tbody>
</table>

Due to Definition 4.3, it holds that $S_{r-\text{lex}} = [s_5, s_4, s_2, s_3, s_1]$ reveals the $r$-lexicographical order of $S = \{s_1, s_2, s_3, s_4, s_5\}$. Consequently, the solution $S^*$ returned by Algorithm 1 with respect to $r$ contains the fifth and the second item, i.e., $S^* = \{s_5, s_2\}$. For the $w$-lexicographical order it holds that $S_{w-\text{lex}} = [s_1, s_2, s_3, s_5, s_4]$ and the corresponding solution of Algorithm 1 with respect to $w$ contains the first, second and third item, i.e., $S^* = \{s_1, s_2, s_3\}$ with $w(S^*) = 6 = W$. Both solutions returned by Algorithm 1 with respect to $r$ and $w$, respectively, can be verified to be efficient, see Example 4.14.

Example 4.10. To show that the solution $S^*$ after $n$ iterations of Algorithm 1 with respect to $w$ does not have to be efficient in case of $w(S^*) < W$,
consider the following example. Let $S = \{s_1, s_2\}$ and $W = 3$. Table 1 shows the different items with their corresponding weights and qualitative levels.

<table>
<thead>
<tr>
<th>item</th>
<th>$w$</th>
<th>$r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>2</td>
<td>$\ell_1$</td>
</tr>
<tr>
<td>$s_2$</td>
<td>3</td>
<td>$\ell_2$</td>
</tr>
</tbody>
</table>

Table 1: Example of Algorithm 1 with respect to $w$ with $w(S^*) < W$

The solution returned by Algorithm 1 with respect to $w$ contains only the first item, i.e., $S^* = \{s_1\}$. However, note that $S' = \{s_2\}$ weakly dominates $S^*$, i.e., $S' \succeq S^*$, since $\sum_{i=j}^2 g_i(S') \geq \sum_{i=j}^2 g_i(S^*)$ for all $j = 1, 2$, and $S^*$ does not weakly dominate $S'$, i.e., $S'$ dominates $S^*$.

Next, we present a dynamic programming algorithm for the knapsack problem with qualitative levels that computes the entire set of non-dominated rank cardinality vectors. Again, let $I = (S, r, w, W)$ denote an instance of our problem. For all $i \in \{0, 1, \ldots, n\}$, and for all $x \in \{0, 1, \ldots, W\}$, we introduce label sets $\mathcal{L}_{i,x}$ referring to those non-dominated rank cardinality vectors that use only the first $i$ items with a total size smaller or equal to $x$. Initially, we set $\mathcal{L}_{0,x}$ to be equal to the empty set, i.e., $\mathcal{L}_{0,x} = \emptyset$ for all $x \in \{0, 1, \ldots, W\}$. Next, we compute $\mathcal{L}_{i,x}$ for all $i \in \{1, \ldots, n\}$ and for all $x \in \{0, 1, \ldots, W\}$ by using the following procedure:

$$
\mathcal{L}_{i,x} = \begin{cases} 
\max \left\{ \mathcal{L}_{i-1,x} \cup (g(\{s_i\}) \oplus \mathcal{L}_{i-1,x-w(s_i)}) \right\}, & \text{if } w(s_i) \leq x \\
\mathcal{L}_{i-1,x}, & \text{else}
\end{cases}
$$
where \( g(\{s_i\}) \oplus \mathcal{L}_{i-1,x-w(s_i)} := \{g(\{s_i\}) + L \mid L \in \mathcal{L}_{i-1,x-w(s_i)}\} \). The “\( \max \geq \)” operator finds all non-dominated rank cardinality vectors with respect to the dominance relation “\( \geq \)” defined in Definition 3.4.

**Algorithm 2** Exact Algorithm

**Input:** An instance \( I = (S, r, w, W) \)

**Output:** All non-dominated rank cardinality vectors

1: for \( x = 0, 1, \ldots, W \) do
2: \( \mathcal{L}_{0,x} \leftarrow \emptyset \)
3: for \( i = 1, \ldots, n \) do
4: for \( x = 0, 1, \ldots, W \) do
5: \( \mathcal{L}_{i,x} \leftarrow \begin{cases} 
\max \geq \{ \mathcal{L}_{i-1,x} \cup (g(\{s_i\}) \oplus \mathcal{L}_{i-1,x-w(s_i)}) \}, & \text{if } w(s_i) \leq x \\
\mathcal{L}_{i-1,x}, & \text{else}
\end{cases} \)
6: return \( \mathcal{L}_{n,W} \)

**Theorem 4.11.** Algorithm 2 correctly computes the set of non-dominated rank cardinality vectors.

**Proof.** Follows immediately by induction over the number of iterations \( i \). \( \Box \)

Note that the knapsack problem with qualitative levels still follows Bellman’s principle of optimality.

**Theorem 4.12** (see [25]). Throughout the execution of Algorithm 2 the number of labels in \( \mathcal{L}_{i,x} \) is polynomially bounded for all \( i \in \{1, \ldots, |S|\} \) and for all \( x \in \{0, 1, \ldots, W\} \).
Proof. A non-dominated rank cardinality vector \( g(S) \) in \( L_{i,x} \) corresponds to an item set \( S \) containing at most \( i \) items with a total weight of at most \( x \). Each component in \( g(S) \) can attain at most \( i + 1 \) values, i.e., \( g_j(S) \in \{0, 1, \ldots, i\} \) for all \( j \in \{1, \ldots, k\} \). Thus, there are at most \((i + 1)^k\) different rank cardinality vectors in \( L_{i,x} \). Consequently, \( L_{i,x} \in \mathcal{O}(i^k) \), which concludes the proof.

Theorem 4.13. Algorithm 2 runs in \( \mathcal{O}(n^{2k+1}W) \).

Proof. Clearly, the first for-loop is in \( \mathcal{O}(W) \). The nested for-loop indicates that there are \( nW \) subproblems that have to be solved. As the algorithm proceeds (increasing \( i \)), the worst-case size of the label sets also increases. The amount of work for the \( i \)-th iteration is in \( \mathcal{O}(i^{2k}W) \), since two label sets of size bounded by \( \mathcal{O}(i^k) \), see Theorem 4.12, have to be searched for non-dominance. Consequently, the overall amount of work is in \( \mathcal{O}(n^{2k+1}W) \), which concludes the proof.

Example 4.14. Let \( I = (S, r, w, W) \) be the same instance as in Example 4.9. Table 2 shows the result of the dynamic programming algorithm, see Algorithm 2. The table has to be read from the bottom left to the top right.
Note that $L_{5,6}$ refers to the set of non-dominated rank cardinality vectors of
the knapsack instance as described in Example 4.9. One can easily recover
the item sets corresponding to the rank cardinality vectors in $L_{5,6}$ by taking
for all $i \in \{1, \ldots, k\}$ those $g_i(S)$ many items with rank $\ell_i$ of least weight.
Consider e.g. the rank cardinality vector $l = (0, 1, 0, 1) \in L_{5,6}$. The item
set corresponding to $l$ contains items $s_2$ and $s_5$, since there are two items in
$S$, i.e., $s_2$ and $s_3$, of rank $\ell_2$, from which we can only choose one. Thus,
we take item $s_2$, since $w(s_2) < w(s_3)$. Consequently, $S_1 = \{s_1, s_2, s_3\}$ and
$S_2 = \{s_2, s_3\}$ denote the efficient solutions of the knapsack instance.

Connection to multiobjective optimization
In this paper, we have shown how the binary knapsack problem with qualita-
tive levels can be formulated. Although a detailed and thorough investigation of connections and possible extensions to multiobjective optimization is not within the scope of this article, we want to point out that the proposed formulation could also be adapted in case of more objectives. Therefore, assume that every qualitative level of rank at least $\ell_i$ in the knapsack problem with qualitative levels corresponds to one criterion in a multiobjective optimization problem (MOP), i.e., $\max\{(\tilde{g}_1(S), \ldots, \tilde{g}_k(S))^\top \mid S \in \mathcal{S}(W)\}$, cf. [1], where $\tilde{g}_i(S) = |\{s \in S \mid r(s) \geq \ell_i\}|$. Then, any efficient solution of the knapsack problem with qualitative levels denotes an efficient solution for (MOP) and vice versa. This can easily be shown: Let $S^*$ be an efficient solution for the knapsack problem with qualitative levels and let $S' \in \mathcal{S}(W)$ be a feasible solution that is not equivalent to $S^*$. Due to Lemma 4.1, there exists some $j$ such that $\tilde{g}_j(S^*) > \tilde{g}_j(S')$. Thus, $S^*$ is efficient for (MOP). On the other hand, let $S^*$ be an efficient solution for (MOP). Then, for every feasible $S' \in \mathcal{S}(W)$ that is not equivalent to $S^*$, it holds that there exists some $j$ such that $\tilde{g}_j(S^*) > \tilde{g}_j(S')$. In particular, Lemma 4.1 implies that $S'$ does not dominate $S^*$. Thus, $S^*$ is efficient for the knapsack problem with qualitative levels. Consequently, instead of computing the set of efficient solutions of an instance of the knapsack problem with qualitative levels, one could compute the set of efficient solutions of the corresponding (MOP). Further, observe that every efficient solution of the knapsack problem with qualitative levels denotes an efficient solution of the following simpler multiobjective problem (MOP'), i.e., $\max\{(g_1(S), \ldots, g_k(S))^\top \mid S \in \mathcal{S}(W)\}$. Again, this can easily be shown, since if $S^*$ is an efficient solution for the knapsack problem with qualitative levels, then for any $S' \in \mathcal{S}(W)$ there exists some numerical rep-
presentation \( v \in \mathcal{V}_r \) such that \( v(S^*) > v(S') \), or for all \( v \in \mathcal{V}_r \) it holds that \( v(S^*) = v(S') \). Choosing a weighted sum vector corresponding to the numerical representation \( v \), i.e., \( \lambda_i := v(\ell_i) \), yields \( \sum_{i=1}^{k} \lambda_i \tilde{g}_i(S^*) \geq \sum_{i=1}^{k} \lambda_i \tilde{g}_i(S') \), i.e., \( S^* \) is efficient for (MOP'). We have seen in Example 4.14 that there are two non-dominated rank cardinality vectors corresponding to the sets \( S_1 = \{s_1, s_2, s_3\} \) and \( S_2 = \{s_2, s_5\} \), respectively. One can check that the vector \( q = (0, 1, 1, 0)^\top \) corresponding to the item set \( S_3 = \{s_2, s_4\} \) would be efficient for (MOP'), although \( q \) does not denote a non-dominated rank cardinality of vector of the knapsack instance. Thus, the converse is in general not true. Further, note that the efficient solution returned by Algorithm 1 with respect to \( r \) equals the \( r \)-lexicographic optimal solution of the corresponding (MOP').

5. Conclusion

In this paper, we investigated the binary knapsack problem with qualitative levels. We introduced a concept of dominance for item sets with qualitative levels. We showed that this concept defines a preorder on the set of feasible subsets of a given item set. We proved that the number of non-dominated rank cardinality vectors is polynomially bounded for a fixed number of qualitative levels. We provided a dynamic programming algorithm, which computes the entire set of non-dominated rank cardinality vectors in pseudo-polynomial time and two greedy algorithms, which efficiently compute a single efficient solution.
Acknowledgements

This work was partially supported by the Bundesministerium für Bildung und Forschung (BMBF) under Grant No. 13N14561 and by Deutscher Akademischer Austauschdienst (DAAD) under Grant No. 57518713. José Rui Figueira also acknowledges the support of national funds through FCT research DOME Project with reference PTDC/CCI-COM/31198/2017. José Rui Figueira was partially supported under the Isambard Kingdom Brunel Fellowship Scheme during a one-month stay (April-May 2018) at the Portsmouth Business School (April-May 2018), U.K.
References


