Fair Knapsack

Till Fluschnik,[1] **Piotr Skowron,**[2] **Mervin Triphaus,**[1] **Kai Wilker**[1]

[1]Algorithmics and Computational Complexity, Faculty IV, TU Berlin, Berlin, Germany
till.fluschnik@tu-berlin.de, {mervin.triphaus,wilker}@campus.tu-berlin.de
[2]Faculty of Mathematics, Informatics, and Mechanics, University of Warsaw, Warsaw, Poland
p.skowron@mimuw.edu.pl

Abstract

We study the following multiagent variant of the knapsack problem. We are given a set of items, a set of voters, and a value of the budget; each item is endowed with a cost and each voter assigns to each item a certain value. The goal is to select a subset of items with the total cost not exceeding the budget, in a way that is consistent with the voters' preferences. Since the preferences of the voters over the items can vary significantly, we need a way of aggregating these preferences, in order to select the socially best valid knapsack. We study three approaches to aggregating voters' preferences, which are motivated by the literature on multiwinner elections and fair allocation. This way we introduce the concepts of individually best, diverse, and fair knapsack. We study the computational complexity (including parameterized complexity, and complexity under restricted domains) of the aforementioned multiagent variants of knapsack.

1   Introduction

In the classic knapsack problem we are given a set of *items*, each having a cost and a value, and a budget. The goal is to find a subset of items with the maximal sum of the values subject to the constraint that the total cost of the selected items must not exceed the budget. In this paper we are studying the following variant of the knapsack problem: instead of having a single objective value for each item we assume that there is a set of *agents* (also referred to as *voters*) who have potentially different valuations (expressed through *utilities*) of the items. When choosing a subset of items we want to take into account possibly conflicting preferences of the voters with respect to which items should be selected: in this paper we discuss three different approaches to how the voters' valuations can be aggregated.

Multiagent knapsack forms an abstract model for a number of real-life scenarios. First, let us note that if the costs of the items are all the same, then the multiagent knapsack model collapses to the model for multiwinner elections (Faliszewski et al. 2017) (in the literature on multiwinner elections, items are often called candidates). Multiwinner voting rules are applicable in a broad class of scenarios, ranging from selecting a representative committee of experts,

through recommendation systems (Lu and Boutilier 2011)[1], to resource allocation and facility location problems. In each of these settings it is quite natural to consider that different items/candidates can incur different costs. Further, algorithms for multiagent knapsack can be viewed as tools for participatory budgeting (Cabannes 2004), where the authorities aggregate citizens' preferences to decide which of the potential local projects should obtain funding.

Perhaps the most straightforward way to aggregate voters' preferences is to select a subset (a knapsack) that maximizes the sum of the utilities of all the voters over all selected items. This approach—which we call selecting an *individually best knapsack*—subject to differences in methods used for eliciting voters' preferences has been taken by Benabbou and Perny (2016), and in the context of participatory budgeting by Goel et al. (2016) and Benade et al. (2017). However, by selecting an individually best knapsack we can disadvantage even large minorities of voters, which is illustrated by the following example: assume that the set of items can be divided into two subsets A_1 and A_2, that all items have the same cost, and that 51% of the voters like items from A_1 (assigning the utility of 1 to them, and the utility of 0 to the other items) and the remaining 49% of voters like only items from A_2. An individually best knapsack would contain only items from A_1, that is 49% of the voters would be effectively disregarded.

In this paper we introduce two other approaches to aggregating voters' preferences for selecting a multiagent knapsack. One such approach—which we call selecting a *diverse knapsack*—is inspired by the Chamberlin–Courant rule (Chamberlin and Courant 1983) from the literature on multiwinner voting. Informally speaking, in this approach we aim at maximizing the number of voters who have *at least one* preferred item in the selected knapsack. For the second approach—which is the main focus of the paper and which we call selecting a *fair knapsack*—we use the concept of Nash welfare (Nash 1950) from the literature on fair allocation. Nash welfare is a well-established solution concept that implements a tradeoff between having an objectively ef-

---

[1]An example often described in the literature is when an enterprise considers which set of products should be pushed to production—it is natural to view such a problem as an instance of multiwinner elections with products corresponding to the items/candidates and potential customers to the voters.

ficient resource allocation (knapsack, in our case), and having an allocation which is acceptable for a large population of agents. Indeed, extensive recent studies in the domain of fair allocation confirm particularly strong fairness guarantees of Nash welfare (Caragiannis et al. 2016; Moulin 2003; Darmann and Schauer 2015), and this solution concept has been applied e.g. in the context of public decision making (Conitzer, Freeman, and Shah 2017), online resource allocation (Freeman, Zahedi, and Conitzer 2017) or transmission congestion control (Kelly 1997) (therein referred to as *proportional fairness*). Thus, our work introduces a new application domain—now the goal is to select a set of shared items—for the concept of Nash welfare. In particular, as a side note, we will explain that our approach leads to a new class of multiwinner rules, which can be viewed as generalizations of the Proportional Approval Voting rule beyond the approval setting.

Apart from introducing the new class of multiagent knapsack problems, our contributions are as follows:

(1) We study the complexity of computing an optimal individually best (IB), diverse, and fair knapsack. This problem is in general NP-hard, except for the case of IB knapsack with unarily encoded utilities of the voters (as the IB knapsack problem is equivalent to the classic knapsack problem).

(2) We study the parameterized complexity of computing a diverse and fair knapsack, focusing on the number of voters.[2] We show that for unary-encoded utilities of the voters computing a diverse knapsack is fixed-parameter tractable when parameterized by the number of voters. On the contrary, computing a fair knapsack is W[1]-hard for the same parameter.

(3) We study the complexity of the considered problems for single-peaked and single-crossing preferences. We show that (under unary encoding of voters' utilities) a diverse knapsack can be computed in polynomial time when the preferences are single-peaked or single-crossing, while computing a fair knapsack remains NP-hard.

Our results are summarized in Table 1. Our main message is that fairness comes with a surprisingly high computational complexity. Indeed, our most unexpected results are that computing a fair knapsack is W[1]-hard when parameterized by the number of voters, and NP-hard on single-peaked single-crossing domains, with unit-costs, and all utilities coming from the set $\{0, \ldots, 6\}$. This was unforeseen since by using a recent result of Peters (2018), one can show that computing a fair knapsack on single-peaked domains (which are not necessarily single-crossing, i.e., when one of the assumptions is weakened), with unit-costs, and all utilities coming from the set $\{0, 1\}$ (instead of $\{0, \ldots, 6\}$, i.e., when another assumption is strengthened) is polynomial-time solvable. Our result required a complex reduction from the exact set cover problem. Most of our proofs are omitted

---

[2]Considering this parameter is relevant, e.g., for the case when the set of voters is in fact a relatively small group of experts acting on behalf of a larger population of agents.

Table 1: Overview of our results (for the case of utilities encoded in unary): Herein, SP and SC abbreviate single-peaked and single-crossing preferences, respectively, and "# voters" refers to "when parameterized by the number of voters". * (Procaccia, Rosenschein, and Zohar 2008)

Knapsack	general	SP	SC	# voters
IB	P (equivalent to KNAPSACK)			$\leftarrow$
Diverse	NP-hard *	P (Thm. 3)	P (Thm. 3)	FPT (Thm. 4)
Fair	NP-hard (Thm. 5)	NP-hard (Thm. 9)		W[1]-hard (Thm. 6)

due to the space constraints.[3]

Most of the our results are presented for the costs and utilities of the agents given in the unary-encoding. This makes our results more relevant for practical applications of our framework. E.g., for participatory budgeting (PB) one does not really need more than thousands of values to represent utilities/costs. Further, by assuming efficient encoding of the utilities/costs we would make the hardness results less meaningful—the hardness would simply be an artifact of the fact that we admit the values of the utilities/costs that are exponentially large in the number of voters. By assuming unary encoding we make—on the one hand—the hardness results stronger, and—on the other hand—more applicable to the real scenarios that our model represents.

We also show all three problems to be NP-hard in the non-unary case (Theorems 2 and 5) and prove W-hardness with respect to the budget (Proposition 1 and Corollary 1).

2 The Model

For a pair of natural numbers $i, j \in \mathbb{N}$, $i \leq j$, by $[i, j]$ we denote the set $\{i, i+1, \ldots, j\}$. Further, we let $[j] = [1, j]$.

Let $V = \{v_1, \ldots, v_n\}$ be the set of n *voters* and $A = \{a_1, \ldots, a_m\}$ be the set of m *items*. The voters have preferences over the items, which are represented as a *utility profile* $u = (u_i(a) \mid i \in [n], a \in A)$: for each $i \in [n]$ and $a \in A$ we use $u_i(a)$ to denote the utility that v_i assigns to a; this utility quantifies the extent to which v_i enjoys a. We assume that all utilities are nonnegative integers.

Each item $a \in A$ comes with a cost $c(a) \in \mathbb{N}$, and we are given a global budget $B \in \mathbb{N}$. We call a *knapsack* a subset S of items whose total cost does not exceed B, that is $c(S) = \sum_{a \in S} c(a) \leq B$. Our goal is to select a knapsack that would be, in some sense, most preferred by the voters. Below, we describe three representative rules which extend the preferences of the individual voters over individual items to their aggregated preferences over all knapsacks. Each such a rule induces a corresponding method for selecting the best knapsack. Our rules are rooted in concepts from the literature on fair division and on multiwinner elections:

Individually best knapsack: this is the knapsack S which

---

[3]They are available in a long version of this paper under https://arxiv.org/abs/1711.04520.

maximizes the total utility of the voters from the selected items $u_{\text{IB}}(S) = \sum_{a \in S} \sum_{v_i \in V} u_i(a)$. This defines perhaps the most straightforward way to select the knapsack: we call it individually best, because the formula $u_{\text{IB}}(S)$ treats the items separately and does not take into account fairness-related issues. Indeed, such a knapsack can be very unfair, as discussed in the introduction.

Diverse knapsack: this is the knapsack S that maximizes the utility $u_{\text{Div}}(S) = \sum_{v_i \in V} \max_{a \in S} u_i(a)$. In words, in the definition of u_{Div} we assume that each voter cares only about his or her most preferred item in the knapsack. This approach is inspired by the Chamberlin–Courant rule (1983) for multiwinner elections and by classic models in facility location (Farahani and Hekmatfar 2009). We call such a knapsack diverse following the convention from the multiwinner literature (Faliszewski et al. 2017). Intuitively, such a knapsack represents the diversity of the opinions among the population of voters; in particular, if the preferences of the voters are very diverse, such a knapsack tries to incorporate the preferences of as many groups of voters as possible at the cost of containing only one representative item for each "similar" group.

Fair knapsack: we use Nash welfare (Nash 1950) as a solution concept for fairness. Formally, we call a knapsack S fair if it maximizes the product $u_{\text{Fair}}(S) = \prod_{v_i \in V} \left(1 + \sum_{a \in S} u_i(a)\right)$.[4] Alternatively, by taking the logarithm of u_{Fair} we can represent fair knapsack as the one maximizing $\sum_{v_i \in V} \log(1 + \sum_{a \in S} u_i(a))$. The following example intuitively explains the type of fairness guaranteed by using the Nash welfare.

Example 1. Consider six groups of voters, $V_1, \ldots, V_6$, with $|V_1| = 300$, $|V_2| = 200$, $|V_3| = 100$, and $|V_4| = |V_5| = |V_6| = 1$. Assume we have six sufficiently large groups of items, $A_1 \ldots, A_6$. Each voter from group V_i assigns utility 1 to all items from group A_i, and zero to all other items. Finally, assume that the costs of all items are equal to 1, and that the value of the budget is 6. An individually best knapsack would consist only of the items from A_1. A diverse knapsack would contain one item from each set A_i for $i \in \{1, \ldots, 6\}$. A fair knapsack would contain 3 items from A_1, 2 items from A_2 and 1 item from A_3—this solution can be interpreted as fair since the number of items selected from each group is proportional to the number of voters liking

---

[4]Typically, Nash welfare would be defined as $\prod_{v_i \in V} \left(\sum_{a \in S} u_i(a)\right)$. In our definition, we add one to the sum $\sum_{a \in S} u_i(a)$ in order to avoid pathological situations when the sum is equal to zero for some voters. This also allows us to represent the expression we optimize as a sum of logarithms, and thus, to expose the close relation between the fair knapsack and the Proportional Approval Voting rule. When the utilities are normalized our definition results in better properties of the outcome pertaining to fairness (Fain, Munagala, and Shah 2018). Further, all our hardness results can be formulated for a weaker (and perhaps the least disputable) notion of fairness in the following way: it is hard to decide whether an instance of the collective knapsack problem admits a solution where the sum of the utilities of all the agents is the highest (among all valid solutions) and all the agents have the same utility.

items from these groups.

If we assume that the costs of the items from groups A_1, A_2, and A_3 are equal to 3, 2, and 1, respectively, and that our budget is equal to 6, then the fair knapsack will consist of one item from each of the sets A_1, A_2, and A_3—this shows that each group will obtain a share of the cost of the whole knapsack which is proportional to its size.

In Section 1 we referred to the literature supporting the use of Nash welfare in various settings. Let us complement these arguments with one additional observation. When the utilities of the voters come from the binary set $\{0, 1\}$ and the costs of all items are one, then our multiagent knapsack framework boils down to the standard multiwinner elections model with approval preferences. In this case, a very appealing rule, Proportional Approval Voting (PAV), can be expressed as finding a knapsack maximizing $\sum_{v_i \in V} H(\sum_{a \in S} u_i(a))$, where $H(i)$ is the i-th harmonic number. This is almost equivalent to finding a fair knapsack (maximizing the Nash welfare) since the harmonic function can be viewed as a discrete version of the logarithm. Thus, fair knapsack can be considered an adjustment of PAV to the model with cardinal utilities and costs. In particular (as a side note), observe that the notion of fair knapsack combined with positional scoring rules induces rules that can be viewed as adaptations of PAV to the ordinal model.

3   Related Work

Our work extends the literature on the multi-objective (MO) knapsack problem (Kellerer, Pferschy, and Pisinger 2004), i.e., on the variant of the classic knapsack problem with multiple independent functions valuating the items. Typically, in the MO knapsack problem the goal is to find a (the set of) Pareto optimal solution(s) according to multiple objectives defined through given functions valuating items. Our approach is different since we consider specific forms of aggregating objectives; in particular for each of our concepts—for the individually best, diverse, and fair knapsack—there always exists a Pareto optimal solution; further each solution to the individually best and fair knapsack is Pareto optimal. For an overview of the literature on the MO knapsack problem (with the focus on the analysis of heuristic algorithms) we refer to the survey by Lust and Teghem (2012).

Lu and Boutilier (2011) studied a variant of the Chamberlin–Courant rule that includes knapsack constraints and so being very similar to our diverse knapsack problem. The difference is that (i) they consider utilities which are extracted from the voters' preference rankings, thus these utilities have a specific structure, and (ii) in their model the items are not shared; instead, the selected items can be copied and distributed among the voters. Lu and Boutilier consider a model with additional costs of copying a selected item and sending it to a voter. Consequently, their general model is more complex than our diverse knapsack; they also considered a more specific variant of the model, equivalent to winner determination under the Chamberlin–Courant rule.

A variant of the diverse knapsack problem with the utilities satisfying a form of the triangle inequality is known as the knapsack median problem; see the work of

Byrka et al. (2015) for a discussion on the approximability of the problem.

As we discussed in the introduction, the multiagent variant of the knapsack problem has been often considered in the context of participatory budgeting, yet to the best of our knowledge this literature focuses on the simplest aggregation rule corresponding to our individually best knapsack approach (Cabannes 2004; Goel et al. 2016; Benade et al. 2017). Another avenue has been explored by Fain et al. (2016), who studied rules that determine the level of funding provided to different projects (items, in our nomenclature) rather than rules selecting subsets of projects with predefined funding requirements.

4   Computing Multiagent Knapsacks

In this section we investigate the computational complexity of finding individually best, diverse, and fair knapsack. Formally, we define the computational problem for computing a fair knapsack as:

FAIR KNAPSACK

Input: An instance (V, A, u, c) and a budget B.

Task: Compute a knapsack $S \subseteq A$ such that $c(S) \leq B$ and $u_{\mathrm{Fair}}(S)$ is maximum.

We define the computational problems DIVERSE KNAPSACK and INDIVIDUALLY BEST KNAPSACK analogously—the difference is only in the expression to maximize, which for the two problems is u_{Div} and u_{IB}, respectively. We will use the same names when referring to the decision variants of these problems; in such cases we will assume that one additional integer x is given in the input, and that the decision question is whether there exists S with value $u_{\mathrm{IB}}(S)$ (respectively, $u_{\mathrm{Div}}(S)$ or $u_{\mathrm{Fair}}(S)$) at least x, and $c(S) \leq B$.

We observe that the functions u_{IB}, u_{Div}, and u_{Fair} (when represented as a sum of logarithms—the use of the logarithm in the objective is only relevant for the approximation ratio) are submodular. Thus, we can use an algorithm of (Sviridenko 2004) with the following guarantees.

Theorem 1. *There is a polynomial-time* $(1 - \mathsf{1}/e)$-*approximation algorithm for* INDIVIDUALLY BEST KNAPSACK, DIVERSE KNAPSACK, *and* FAIR KNAPSACK *with the objective function* $\log(u_{\mathrm{Fair}})$.

In the remainder of the paper we focus on computing exact solutions for the three problems. In particular, we study the complexity under the following two restricted domains:

Single-peaked preferences. Let top_i denote v_i's most preferred item, and let $\lhd$ be an order of the items. We say that a utility profile u is *single-peaked* with respect to $\lhd$ if for each $a, b \in A$ and each $v_i \in V$ such that $a \lhd b \lhd \mathrm{top}_i$ or $\mathrm{top}_i \lhd b \lhd a$ we have that $u_i(b) \geq u_i(a)$.

Single-crossing preferences. Let $\lhd$ be an order of the voters. We say that a utility profile u is *single-crossing* with respect to $\lhd$ if for each two items $a, b \in A$ the set $\{v_i \in V \mid u_i(b) \geq u_i(a)\}$ forms a consecutive block according to $\lhd$.

We say that a profile u is single-peaked (resp., single-crossing) if there exists an order $\lhd$ of the items (resp., of the

voters) such that u is single-peaked (resp., single-crossing) with respect to $\lhd$. Note that an order witnessing single-peakedness or single-crossingness can be computed in polynomial time (see, e.g., (Elkind, Lackner, and Peters 2017; Fitzsimmons 2015)).

We will also study the parameterized complexity of the three problems. For a given parameter p, we say that a problem is fixed-parameter tractable (FPT) when parameterized by p if there is an algorithm (FPT algorithm) that solves each instance I of the problem in $O\big(f(p) \cdot \mathrm{poly}(|I|)\big)$ time, where f is some computable function. In parameterized algorithmics, FPT algorithms are considered efficient. There is a whole hierarchy of complexity classes, but informally speaking, a problem that is W[1]- or W[2]-hard is assumed not to be FPT and, hence, hard (or fixed-parameter intractable) from the parameterized point of view (see (Downey and Fellows 2013) for more details).

Note that if the utilities are not unarily encoded, then FAIR KNAPSACK is NP-hard even for one voter (see Theorem 5).

Diverse Knapsack

We now turn our attention to the problem of computing a diverse knapsack. Through a straightforward reduction from the standard knapsack problem, we get that DIVERSE KNAPSACK is computationally hard even for profiles which are both single-peaked and single-crossing, unless the utilities are provided in unary encoding.

Theorem 2. DIVERSE KNAPSACK *is* NP-*hard even for single-peaked and single-crossing utility profiles.*

Proof. We present a many-one reduction from KNAPSACK. Let $(X = \{x_1, \ldots, x_n\}, x, y)$ be an instance of KNAPSACK where each x_i comes with value $\nu(x_i) \geq 1$ and weight $\omega(x_i)$; the question is whether there exists $S \subseteq X$ with $\sum_{x_i \in S} \nu(x_i) \geq x$ and $\sum_{x_i \in S} \omega(x_i) \leq y$. We set our set of items $A = \{a_1, \ldots, a_n\}$ with $c(a_i) := \omega(x_i)$ for each $i \in [n]$. We add n voters $v_1, \ldots, v_n$ with

$$u_i(a_j) := \begin{cases} 3n^2 \cdot \nu(a_j), & i = j, \\ j, & i > j, \\ 2n - j + 1, & i < j. \end{cases}$$

It is immediate that for each S we have that $\sum_{a_i \in S} c(a_i) = \sum_{x_i \in S} \omega(x_i)$. Further, $\sum_{v_i \in V} \max_{a_j \in S} u_i(a_j) \geq 3n^2 x$ if and only if $\sum_{x_j \in S} \nu(x_j) \geq x$, which proves the correctness. It is immediate to check that the utility profile is single-peaked and single-crossing. □

Note that DIVERSE KNAPSACK is NP-hard for utilities encoded in unary as it generalizes the Chamberlin–Courant rule, which is computationally hard (Procaccia, Rosenschein, and Zohar 2008). For single-peaked or single-crossing profiles the Chamberlin-Courant rule is computable in polynomial time (Betzler, Slinko, and Uhlmann 2013; Skowron et al. 2015). These known algorithms can be extended to the case of the diverse knapsack.

Theorem 3. DIVERSE KNAPSACK *is solvable in polynomial time when the utility profile is*

(i) single-peaked and encoded in unary;

(ii) single-crossing and encoded in unary.

Theorem 3(i) is proven via straight-forward dynamic programming and omitted due to space constraints. We prove our result for single-crossingness. Let us define a set of useful tools. We will also use these tools later on, when analyzing the parameterized complexity of the problem.

Given a tuple of voters $\vec{V} = (v_1, \ldots, v_n)$ and a subset $S \subseteq A$ of items, we define an *assignment* $\pi_{S,\vec{V}}$ as a surjection $[n] \to S$. An assignment is called *connected*, if for every $s \in S$ it holds that $\pi_{S,\vec{V}}^{-1}(s) := \{i \in [n] \mid s = \pi_{S,\vec{V}}(i)\} = [x, y]$ for some $x, y \in [n]$, $y \geq x$. For our first tool we introduce the following auxiliary problem.

ORDERED DIVERSE KNAPSACK

Input: An instance $(\vec{V}, A, u, c)$ where $\vec{V} = (v_1, \ldots, v_n)$ is ordered and a budget B.

Task: Compute a knapsack $S \subseteq A$ such that $c(S) \leq B$, and $u_{\mathrm{Ord}}(S) = \max_{\text{connected } \pi_{S,\vec{V}}} \sum_{i=1}^{n} u_i(\pi_{S,\vec{V}}(i))$ is maximum.

If $S = \{s_1, \ldots, s_\ell\} \subseteq A$ is a cost-minimal solution to DIVERSE KNAPSACK on (V, A, u, c), then let $V_i := \{v_j \in V \mid s_i \in \arg\max_{a \in S} u_j(a)\}$. Consider an ordering $\vec{V} = (V_1, \ldots, V_\ell)$, where for each $i \in [\ell]$, the voters in V_i are arbitrarily ordered. Then it is not difficult to see that the assignment $\pi_{S,\vec{V}}(i) = \arg\max_{a \in S} u_i(a)$ is connected. Hence, we obtain the following connection between DIVERSE KNAPSACK and ORDERED DIVERSE KNAPSACK.

Observation 1. *There is an ordering $\vec{V}$ on the voters V such that there is an $S \subseteq A$ that forms a cost-minimal solution for ORDERED DIVERSE KNAPSACK and for DIVERSE KNAPSACK.*

Next, we give a dynamic program for computing knapsacks that qualitatively lie "between" optimal solutions for ORDERED DIVERSE KNAPSACK and DIVERSE KNAPSACK (what "lying in between" means is specified later on).

Let us fix an input (V, A, u, c, B) and an ordering $\vec{V} = (v_1, \ldots, v_n)$ of the voters. We set $\hat{u} := \sum_{i=1}^{n} \sum_{a \in A} u_i(a)$. We give a dynamic program with table T, where $T[i, x]$ denotes "some" cost of a knapsack with a value assigned by voters from $(v_1, \ldots, v_i)$ at least equal to x. We set $T[1, x] = \min\{c(a) \mid a \in A, u_1(a) \geq x\}$, if there is an $a \in A$ such that $u_1(a) \geq x$, and $T[1, x] = \infty$ otherwise. We define a helper function

$$f(i, a, x) = \begin{cases} c(a), & \text{if } \sum_{j=1}^{i} u_j(a) \geq x, \\ \infty, & \text{otherwise.} \end{cases}$$

We set

$$T[i, x] = \tag{1}$$
$$\min_{a \in A} \left(\begin{array}{l} f(i, a, x), \\ c(a) + \min_{j \in [i-1]} T[j, \max(0, x - \sum_{\ell=j+1}^{i} u_\ell(a))] \end{array} \right).$$

Observation 2. *When the utilities are unarily encoded, we can compute all entries of T in polynomial time.*

Lemma 1. *Let S be a cost-minimal solution to DIVERSE KNAPSACK on (V, A, u, c) and let $x = u_{\mathrm{Div}}(S)$. Then $T[n, x] \geq c(S)$.*

Proof. Suppose that this is not the case, that is, $T[n, x] < c(S)$. Then we construct a knapsack S' from $T[n, x]$ as follows. Let $a \in A$ be an item that minimizes (1) for $T[n, x]$, then make $a \in S'$. If $T[n, x] = f(n, a, x)$, then $T[n, x] = c(a) < c(S)$, contradicting the fact that S is cost-minimal. Otherwise,

$$T[n, x] = c(a) + T[j, x' := \max(0, x - \sum_{\ell=j+1}^{n} u_\ell(a))],$$

for some $j \in [n - 1]$. Then we proceed towards a contradiction as before: Let $a' \in A$ be an item that minimizes (1) for $T[j, x']$, then make $a' \in S'$, and continue the same reasoning. $\square$

Lemma 2. *Let S be a cost-minimal solution to ORDERED DIVERSE KNAPSACK on $(\vec{V}, A, u, c)$ where $\vec{V}$ is ordered and let $x = u_{\mathrm{Ord}}(S)$. Then $T[n, x] \leq c(S)$.*

We have all ingredients at hand to prove our main results.

Proof of Theorem 3(ii). If $\vec{V}$ is an order witnessing single-crossingness, then there is an $S \subseteq A$ that forms a cost-minimal solution for ORDERED DIVERSE KNAPSACK and for DIVERSE KNAPSACK. Lemmas 1 and 2 guarantee that our algorithm will find it. $\square$

Further, we can use our tools to obtain an FPT algorithm (for the number of voters) for unrestricted domains.

Theorem 4. DIVERSE KNAPSACK *is FPT when parameterized by the number of voters and the utilities are unarily encoded.*

Proof. By Observation 1, we know that there is an order $\vec{V}$ on the voters such that there is an $S \subseteq A$ that forms a cost-minimal solution for ORDERED DIVERSE KNAPSACK and for DIVERSE KNAPSACK. Together with Lemmas 1 and 2 we obtain that for $\vec{V}$ our dynamic program will find such S. Hence, for each ordering of V, we compute $T[n, x]$. Then, we take the minimum over all observed values. Note that x is the largest value such that $T[n, x] \leq B$ for some ordering of the voters. Altogether, this yields a running time of $O(n! \operatorname{poly}(\hat{u} + n + m)) \subseteq O(2^{n \log n} \operatorname{poly}(\hat{u} + n + m))$. $\square$

Finally, we complement Theorem 4 by proving a lower bound on the running time (via reducing from the W[2]-complete DOMINATING SET problem), assuming the Exponential-Time Hypothesis (ETH).

Proposition 1. DIVERSE KNAPSACK *with binary utilities and unary costs is* W[2]-*hard when parameterized by the budget B and, unless the ETH breaks, there is no $2^{o(n+m)} \cdot \operatorname{poly}(n + m)$ algorithm.*

Fair Knapsack

Let us now turn to the problem of computing a fair knapsack. We first prove that the problem is NP-hard, even for restricted cases, and then we study its parameterized complexity.

Theorem 5. FAIR KNAPSACK *is* NP-*hard, even*

1. *for one voter;*
2. *for two voters and when all costs are equal to one;*
3. *if all utilities are in* $\{0,1\}$ *and all costs are equal to one.*

The proof of Theorem 5 (3) uses the reduction from EX-ACT REGULAR SET PACKING (ERSP). Since ERSP is W[1]-hard with respect to the size of the solution (Ausiello, D'Atri, and Protasi 1980), we get the following corollary.

Corollary 1. FAIR KNAPSACK *is* W[1]-*hard when parameterized by the budget, even if all utilities are in* $\{0,1\}$ *and all costs are equal to one.*

Using a different construction, we can show that for the combination of the two parameters—the number of voters and the budget—we still get fixed-parameter intractability.

Theorem 6. FAIR KNAPSACK *is* W[1]-*hard when parameterized by the number of voters and the budget, even if the utilities and the budget are represented in unary encoding and the costs of all items are equal to one.*

Proof. We provide a parameterized reduction from the k-MULTICOLORED CLIQUE problem, which is known to be W[1]-hard with respect to the number of colors. Let I be an instance of k-MULTICOLORED CLIQUE. In I we are given a graph G with the set $V(G)$ of vertices and the set $E(G)$ of edges, a natural number $k \in \mathbb{N}$, and a coloring function $f\colon V(G) \to [k]$ that assigns one of k colors to each vertex. We ask if G contains k pairwise connected vertices, each having a different color. Without loss of generality we assume that $k \geq 2$.

From I we construct an instance I_F of FAIR KNAPSACK as follows (we refer to Figure 1 for an illustration). Let $T = |V(G)|$. We set the set of items to $V(G) \cup E(G)$, that is we associate one item with each vertex and with each edge. We construct the set of voters as follows (unless specified otherwise, by default we assume that a voter assigns utility of zero to an item):

1. For each color we introduce one voter who assigns utility of T to each vertex with this color. Clearly, there are k such voters.
2. For each pair of two different colors we introduce $k-2$ voters, each assigning utility of T to each edge that connects two vertices with these two colors. There are $(k-2)\binom{k}{2}$ such voters.
3. For each ordered pair of colors, c_1 and c_2, with $c_1 \neq c_2$ we introduce two voters, call them a and b, with the following utilities. Consider the set of vertices with color c_1 and rename them in an arbitrary way so that they can be put in a sequence $n_1, n_2, \ldots, n_\ell$. For each $i \in [\ell]$ voter a assigns utility i to vertex n_i and utility $(T-i)$ to each edge that connects n_i with a vertex with color c_2. Voter b assigns utility $(T-i)$ to n_i and utility i to each edge
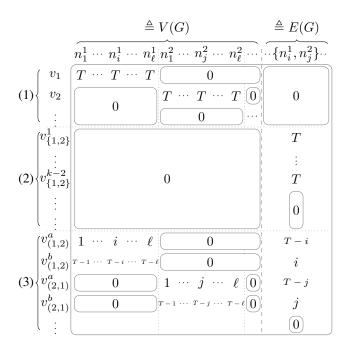

Figure 1: Illustration of the instance obtained in the proof of Theorem 6. Herein, n_b^c denotes vertex b in color class c, where each color class contains ℓ vertices. In the presented example, the vertices n_i^1 and n_j^2 are adjacent. Blocks containing a zero indicate that the corresponding entries are zero.

that connects n_i with a vertex with color c_2. There are $2k(k-1)$ such voters.

We set the cost of each item to one, and the total budget to $B = k + \binom{k}{2}$. By a simple calculation one can check that the total number of voters is equal to $k + (k-2)\cdot\binom{k}{2} + 2k(k-1) = kB$. This completes our construction.

First, observe that in total each item is assigned utility kT from all the voters. Indeed, each item corresponding to a vertex gets utility of T from exactly one voter from the first group, and total utility of $(k-1)\cdot T$ from $2(k-1)$ voters from the third group. Similarly, each item corresponding to an edge gets utility of T from $k-2$ voters from the second group, and total utility of $2\cdot T$ from four voters from the third group. Thus, independently of how we select B items, the sum of the utilities they are assigned from the voters will always be the same, that is BkT. Thus, clearly the Nash welfare would be maximized if the total utility assigned to the selected items by each voter is the same, and equal to T. Only in such case the Nash welfare would be equal to $(T+1)^{kB}$. We will show, however, that each voter assigns to the set of B items utility T if and only if k out of such items are vertices with k different colors, the remaining $\binom{k}{2}$ of such items are edges, and each selected edge connects two selected vertices.

Indeed, it is easy to see that if the selected set of items has the structure as described above, then each voter assigns to this set the utility of T. We will now prove the other im-

plication. Assume that for the set of B items S each voter assigns total utility of T. By looking at the first group of voters, we infer that k items from S correspond to the vertices, and that these k vertices have different colors. By looking at the second group of voters, we infer that for each pair of two different colors, S contains exactly one edge connecting vertices with such colors. Finally, by looking at the third group of voters we infer that each edge from S that connects colors c_1 and c_2 is adjacent to the vertices from S with colors c_1 and c_2. This completes the proof. □

By Theorem 6 we presumably cannot hope for an FPT algorithm for FAIR KNAPSACK when parameterized by the number n of voters. However, each instance I of FAIR KNAPSACK with unarily encoded utilities is solvable in $O(|I|^{f(n)})$ time (that is, it is in XP when parameterized by n), where f is some computable function only depending on n.

Theorem 7. *For unarily encoded utilities,* FAIR KNAPSACK *is in* XP *when parameterized by the number of voters.*

On the positive side, with stronger requirements on the voters' utilities, that is, if the number of different values over the utilities is small, we can strengthen Theorem 7 and prove membership in FPT (using integer linear programming).

Theorem 8. FAIR KNAPSACK *is* FPT *when parameterized by the combination of the number of voters and the number of different values that a utility function can take.*

Proof. We will use the classic result of Lenstra (1983) which says that an integer linear program (ILP) can be solved in FPT time with respect to the number of integer variables. We will also use a recent result of Brederreck et al. (2017) who proved that one can apply concave/convex transformations of certain variables in an ILP, and that such a modified program can be still solved in an FPT time. We construct an ILP as follows. Let U be the set of values that a utility function can take. For each vector $z = (z_1, \ldots, z_n)$ with $z_i \in U$ for each i, we define A_z as the set of items a such that for each voter v_i we have $u_i(a) = z_i$. Intuitively, A_z describes a subcollection of the items with the same "type": such items are indistinguishable when we look only at the utilities assigned by the voters; they may vary only with their costs. For each such a set A_z we introduce an integer variable x_z which intuitively denotes the number of items from the optimal solution that belong to A_z. Further, we construct a function f_z such that $f_z(x)$ is the cost of the x cheapest items from A_z; clearly f_z is convex. We formulate the following program:

$$
\begin{aligned}
\text{maximize:} \quad & \sum_{v_i \in V} \log \left(\sum_{z \in U^n} z_i \cdot x_z \right) \\
\text{subject to:} \quad & \sum_{z \in U^n} f_z(x_z) \leq B \\
& x_z \in \mathbb{Z}, \qquad\qquad z \in U^n
\end{aligned}
$$

The above program uses concave transformations (logarithms) for the maximized expression, and convex transformations (functions f_z) in the left-hand sides of the constraints, so we can use the result of Brederreck et al. (2017) and claim that this program can be solved in an FPT time with respect to the number of integer variables. This completes the proof. □

Fair Knapsack under Restricted Domains

In contrast to INDIVIDUALLY BEST KNAPSACK and DIVERSE KNAPSACK, both being solvable in polynomial time on restricted domains, FAIR KNAPSACK remains NP-hard on utility profiles that are even both, single-peaked and single-crossing.

Theorem 9. FAIR KNAPSACK *is* NP-*hard even on single-peaked single-crossing domains, when the costs of all items are equal to one, and the utilities of each voter come from the set* $\{0, \ldots, 6\}$.

As we discussed in Section 2, if the voters' utilities come from the binary set $\{0, 1\}$ and if the costs of the items are equal to one, then FAIR KNAPSACK is equivalent to computing winners according to Proportional Approval Voting. For this case with single-peaked preferences, Peters (2018) showed that the problem can be formulated as an integer linear program with totally unimodular constraints, and thus it is solvable in polynomial time. This makes our result interesting, as it shows that by allowing slightly more general utilities (coming from the set $\{0, \ldots, 6\}$ instead of $\{0, 1\}$) the problem becomes already NP-hard (even if we additionally assume single-crossingness of the preferences).

5 Conclusion

We studied three variants of the knapsack problem in multiagent settings. One of these variants, selecting an individually best knapsack, has been considered in the literature before, and our work introduces the other two concepts: diverse and fair knapsack. Our paper establishes a relation between the multiagent knapsack model and a broad literature including work on multiwinner voting and on fair allocation. This way, we expose a variety of ways in which the preferences of the voters can be aggregated in different applications that are captured by the abstract model of the multiagent knapsack problem.

Our complexity results are outlined in Table 1. In summary, we showed that computing an individually best or a diverse knapsack can be done efficiently under some constraints. On the contrary, we give multiple evidences that computing a fair knapsack is computationally hard. This also motivates the study of approximation and heuristic algorithms for computing a fair knapsack.

Acknowledgments

References

Ausiello, G.; D'Atri, A.; and Protasi, M. 1980. Structure preserving reductions among convex optimization problems. *Journal of Computer and System Sciences* 21:136–153.

Benabbou, N., and Perny, P. 2016. Solving multi-agent knapsack problems using incremental approval voting. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI-16)*, 1318–1326.

Benade, G.; Nath, S.; Procaccia, A.; and Shah, N. 2017. Preference elicitation for participatory budgeting. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI-17)*, 376–382.

Betzler, N.; Slinko, A.; and Uhlmann, J. 2013. On the computation of fully proportional representation. *J. of Artificial Intelligence Research* 47:475–519.

Bredereck, R.; Faliszewski, P.; Niedermeier, R.; Skowron, P.; and Talmon, N. 2017. Mixed integer programming with convex/concave constraints: Fixed-parameter tractability and applications to multicovering and voting. Technical Report arXiv:1709.02850 [cs.DS], arXiv.org.

Byrka, J.; Pensyl, T.; Rybicki, B.; Spoerhase, J.; Srinivasan, A.; and Trinh, K. 2015. An improved approximation algorithm for knapsack median using sparsification. In *Proceedings of 23rd Annual European Symposium on Algorithms (ESA-2015)*, 275–287.

Cabannes, Y. 2004. Participatory budgeting: a significant contribution to participatory democracy. *Environment and Urbanization* 16(1):27–46.

Caragiannis, I.; Kurokawa, D.; Moulin, H.; Procaccia, A.; Shah, N.; and Wang, J. 2016. The unreasonable fairness of maximum Nash Welfare. In *Proceedings of the 2016 ACM Conference on Economics and Computation (EC-16)*, 305–322.

Chamberlin, B., and Courant, P. 1983. Representative deliberations and representative decisions: Proportional representation and the Borda rule. *American Political Science Review* 77(3):718–733.

Conitzer, V.; Freeman, R.; and Shah, N. 2017. Fair public decision making. In *Proceedings of the 2017 ACM Conference on Economics and Computation (EC-17)*, 629–646.

Darmann, A., and Schauer, J. 2015. Maximizing Nash product social welfare in allocating indivisible goods. *European Journal of Operational Research* 247(2):548–559.

Downey, R., and Fellows, M. 2013. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer.

Elkind, E.; Lackner, M.; and Peters, D. 2017. Structured preferences. In Endriss, U., ed., *Trends in Computational Social Choice*. AI Access.

Fain, B.; Goel, A.; and Munagala, K. 2016. The core of the participatory budgeting problem. In *Proceedings of the 12th Conference on Web and Internet Economics (WINE-16)*, 384–399.

Fain, B.; Munagala, K.; and Shah, N. 2018. Fair public decision making. In *Proceedings of the 2018 ACM Conference on Economics and Computation (EC-18)*, 575–592.

Faliszewski, P.; Skowron, P.; Slinko, A.; and Talmon, N. 2017. Multiwinner voting: A new challenge for social choice theory. In Endriss, U., ed., *Trends in Computational Social Choice*. AI Access.

Farahani, F. Z., and Hekmatfar, M., eds. 2009. *Facility Location: Concepts, Models, and Case Studies*. Springer.

Fitzsimmons, Z. 2015. Single-peaked consistency for weak orders is easy. In *Proceedings of the 17th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-15)*, 127–140.

Freeman, R.; Zahedi, S.; and Conitzer, V. 2017. Fair and efficient social choice in dynamic settings. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI-17)*, 4580–4587.

Goel, A.; Krishnaswamy, A.; Sakshuwong, S.; and Aitamurto, T. 2016. Knapsack voting: Voting mechanisms for participatory budgeting. Manuscript.

Kellerer, H.; Pferschy, U.; and Pisinger, D. 2004. *Knapsack problems*. Springer.

Kelly, F. 1997. Charging and rate control for elastic traffic. *European Transactions on Telecommunications* 8:33–37.

Lenstra, Jr., H. 1983. Integer programming with a fixed number of variables. *Mathematics of Operations Research* 8(4):538–548.

Lu, T., and Boutilier, C. 2011. Budgeted social choice: From consensus to personalized decision making. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-11)*, 280–286.

Lust, T., and Teghem, J. 2012. The multiobjective multidimensional knapsack problem: a survey and a new approach. *International Transactions in Operational Research* 19(4):495–520.

Moulin, H. 2003. *Fair Division and Collective Welfare*. The MIT Press.

Nash, J. 1950. The bargaining problem. *Econometrica* 18(2):155–162.

Peters, D. 2018. Single-peakedness and total unimodularity: New polynomial-time algorithms for multi-winner elections. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI-18)*. To appear.

Procaccia, A.; Rosenschein, J.; and Zohar, A. 2008. On the complexity of achieving proportional representation. *Social Choice and Welfare* 30(3):353–362.

Skowron, P.; Yu, L.; Faliszewski, P.; and Elkind, E. 2015. The complexity of fully proportional representation for single-crossing electorates. *Theoretical Computer Science* 569:43–57.

Sviridenko, M. 2004. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters* 32(1):41–43.