A Preliminary Study of Multi-Resource Participatory Budgeting

Gerson Foks, **Nima Motamed** and **Arie Soeteman**
University of Amsterdam
gerson.foks@gmail.com, nima@motamed.nl, awsoeteman@hotmail.com

Abstract

We initiate the formal study of participatory budgeting elections with multiple resources. We propose a model for such elections and show it can capture constraints like dependencies between projects. After defining desirable properties for mechanisms in this model (discount-monotonicity, proportionality, and immunity to manipulation), we examine two mechanisms (greedy- and max-approval) and find that they fail to satisfy all of these except discount-monotonicity. Also taking a computational approach towards these mechanisms, we find a general hardness result for the computation of max-approval, but also a pseudo-polynomial algorithm for fixed amounts of resources.

1 Introduction

Participatory budgeting (PB) is a central development in deliberative grassroots democracy that has currently been implemented in hundreds of cities spread across the globe [9]. It enables citizens to vote directly on the distribution of a budget over various projects, after which an aggregation mechanism is applied to decide on a final outcome. In practice however, this process is often not in itself sufficient to realise governmental policy. Goldfrank [6] and Rose and Omolo [8] mention that interference of government officials is often required to determine both the technical feasibility of projects and their value within the government policy as a whole, thereby reducing the transparency of the process. Feasibility of projects does not solely depend on monetary funds, but on a variety of constraints, such as completion time, spatial demands, and project dependencies or incompatibilities.

Most models for PB elections formulated within the field of social choice theory (c.f. Aziz, Lee, and Talmon [2], Goel et al. [5], and Talmon and Faliszewski [10]) are incapable of dealing with the presence of such additional constraints. As a solution to this shortcoming, we propose PB elections with multiple resources to be spent on projects, as also considered by Aziz and Shah [3]. By including these constraints as explicit parts of the election, the need for governmental inference in determining project feasibility could potentially be reduced.

As a first step towards a better understanding of our proposal, we initiate the formal analysis of PB elections with multiple resources. As we will show, this model is quite rich, being able to even express additional constraints like dependencies and incompatibilities between projects. We define desirable axiomatic properties for mechanisms in this model, based on known axioms for single-resource elections. Furthermore, we analyse two intuitive and easily-defined aggregation mechanisms with respect to these axioms. Finally, we consider some of the computational (i.e. complexity-theoretic) ramifications of adding multiple resources to PB elections w.r.t. the aforementioned mechanisms.

Paper outline. We introduce the multi-resource model along with the mechanisms and axioms we consider in Section 2, and formalise the capacity of the model to express additional constraints in Section 3. In Sections 4 and 5 we analyse the axiomatic and computational properties, respectively, of the defined mechanisms. Finally, in Section 6 we conclude with a brief discussion of our findings and directions for future work.

2 The Multi-Resource Setting

In this section we introduce our model, largely based on the bounded discrete PB model described by Aziz and Shah [3], and subsequently define the mechanisms and axioms to be analysed in Sections 4 and 5.

2.1 Model Definition

A *d-resource PB election* (also called a *multi-resource PB election* when we do not care about the specific d) consists of a finite set of *projects* $P = \{p_1, \dots, p_m\}$, a *cost function* vector $\boldsymbol{c} = \langle c_1, \dots, c_d \rangle$ consisting of cost functions $c_j : P \to \mathbb{Z}$, and a *budget* vector $\boldsymbol{b} = \langle b_1, \dots, b_d \rangle \in \mathbb{N}^d$. We often speak of the set of resources R, defined as $R := \{1, \dots, d\}$. A set of projects $S \subseteq P$ (usually referred to as an *outcome*) is said to be *feasible* in such an election (and we denote the set of feasible outcomes by $\textsc{Feas}(P, \boldsymbol{c}, \boldsymbol{b})$) if $c_j(S) \leqslant b_j$ for all $j \in R$, where we write $c_j(S) := \sum_{p \in S} c_j(p)$ by abuse of notation. Intuitively, feasible outcomes are those we can actually realise. Throughout this paper, we make the implicit and natural assumption that we are only working with elections in which $\textsc{Feas}(P, \boldsymbol{c}, \boldsymbol{b}) \neq \emptyset$.

Voting is performed by letting each voter i in a community of *voters* $N = \{1, \dots, n\}$ submit a non-empty *approval ballot* $A_i \subseteq P$, giving rise to an *approval profile* $\boldsymbol{A} = \langle A_1, \dots, A_n \rangle$. Given an outcome $S \subseteq A$, we refer to its *approval score* w.r.t $\boldsymbol{A}$ as the quantity $\sum_{i \in N} |S \cap A_i|$. We also sometimes refer to the approval score of an individual project p, which is the quantity $|\{i \in N \,;\, p \in A_i\}|$. Outcomes of elections are determined by ***PB mechanisms***, which are functions $F : (2^P)^n \to 2^{\textsc{Feas}(P, \boldsymbol{c}, \boldsymbol{b})} \setminus \{\emptyset\}$ that select one or more feasible outcomes given an approval profile for some election. Throughout this paper we consider only *resolute* PB mechanisms, which are functions $F : (2^P)^n \to \textsc{Feas}(P, \boldsymbol{c}, \boldsymbol{b})$ that choose exactly one outcome. Additionally, to specify the election under consideration we occasionally write $F(\boldsymbol{A}, \langle P, \boldsymbol{c}, \boldsymbol{b} \rangle)$ instead of $F(\boldsymbol{A})$, by abuse of notation.

2.2 Mechanisms and Axioms

The mechanisms we consider are generalisations of the *greedy-* and *max-approval* mechanisms known for single-resource PB elections (c.f. Goel et al. [5] and Talmon and Faliszewski [10]). Greedy-approval, which we will sometimes refer to as F_{greedy}, works by iteratively accepting projects in descending order of their approval score, whilst rejecting projects that would cause the outcome up until that point to become infeasible. The max-approval mechanism, which we will sometimes refer to as $F_{\max}$, instead directly chooses the outcome $S \in \textsc{Feas}(P, \boldsymbol{c}, \boldsymbol{b})$ with the highest approval score. Both mechanisms break ties using a predefined lexicographical ordering of individual projects, where sets are ordered by their first non-shared project.

We evaluate these PB mechanisms with respect to desirable axiomatic properties, which are based on known axioms for the single-resource setting. Talmon and Faliszewski [10] have formulated the axiom of *discount-monotonicity* for single-resource PB elections, which expresses that every project accepted for some profile should still be accepted if that project's cost decreases unilaterally. This is desirable, since mechanisms without this property can encourage increasing the cost of projects for strategic reasons. Extending to our model, we obtain the following definition.

Definition 1. A PB mechanism F is called *discount-monotonic* if $p \in F(\boldsymbol{A}, \langle P, \boldsymbol{c}, \boldsymbol{b} \rangle)$ implies $p \in F(\boldsymbol{A}, \langle P, \boldsymbol{c}', \boldsymbol{b} \rangle)$, whenever:

i. $c_j(p) \geqslant c'_j(p)$ for all $j \in R$,

ii. there is some $j \in R$ such that $c_j(p) > c'_j(p)$, and

iii. $c_j(p') = c'_j(p')$ for all projects $p' \neq p$ and $j \in R$.

Aziz, Lee, and Talmon [2] define the axiom of *proportionality* to express that large enough groups of like-minded voters should be represented in the outcomes of a PB mechanism. Again extending these concepts, we obtain the following definition.

Definition 2. A PB mechanism F is called *proportional* if for all $C \subseteq N$, project sets $S \subseteq P$ with $c_j(S) \leqslant \frac{|C|}{n} \cdot b_j$ for all $j \in R$, and profiles $\boldsymbol{A}$ with $A_i = S$ for all $i \in C$, we have that $S \subseteq F(\boldsymbol{A})$.

Finally, we consider axioms expressing mechanisms' immunity to strategic manipulation by voters. Finding a single intuitive formulation of this property in the multi-resource setting proves to be trickier than for the previous properties, since the voters' preferences over outcomes is unknown. Following the conceptual structure of the Duggan-Schwartz theorem (c.f. Taylor [11]), we consider two formulations of immunity to strategic manipulation, based on different assumptions about the voters' mentality.

We represent voters' preferences by assuming each voter i has some weak preference order over outcomes, with some truly preferred outcome $S_i^* \subseteq P$. Furthermore, in order to distinguish those resources that a voter finds relevant, we parameterise our definitions of voters' mentalities by sets of resources. We firstly consider voters that only prefer an outcome over another, if for every resource they are interested in, the total costs spent on and profits obtained from their truly preferred projects in that outcome is higher than that of the other outcome. We call these voters *Paretian*.

Definition 3. A voter $i \in N$ is called *K-Paretian* for some non-empty $K \subseteq R$ if she weakly prefers outcome S to S' iff $\sum_{p \in S \cap S_i^*} |c_j(p)| \geqslant \sum_{p \in S' \cap S_i^*} |c_j(p)|$ for all $j \in K$.

We secondly consider voters that make finer distinctions between outcomes than Paretian voters. They prefer outcomes that assign a higher *average* proportion of the costs and profits to the projects they truly prefer, w.r.t. to the resources they are interested in. We call these voters *averagists*.

Definition 4. A voter $i \in N$ is called *K-averagist* for some non-empty $K \subseteq R$ if she weakly prefers outcome S to S' iff

$$\sum_{j \in K} \frac{\sum_{p \in S \cap S_i^*} |c_j(p)|}{b_j} \geqslant \sum_{j \in K} \frac{\sum_{p \in S' \cap S_i^*} |c_j(p)|}{b_j}.$$

Intuitively, averagists might still prefer an outcome in which they are less content with the way one resource is spent, if this is sufficiently compensated by another resource. Note that averagists truly make finer distinctions between outcomes than Paretian voters - in fact, it can be checked quite easily that if a Paretian voter prefers outcome S over S', then that same voter would still have this preference if she were an averagist.

We can now define immunity to manipulation by these types of voters.

Definition 5. A PB mechanism F is called *immune to manipulation by K-Paretian voters* (resp. *K-averagist voters*) for some non-empty $K \subseteq R$, if for any profile $\boldsymbol{A}$ it holds that every K-Paretian (resp. K-averagist) voter i weakly prefers $F(\boldsymbol{A}_{-i}, S_i^*)$ to $F(\boldsymbol{A})$.

We further define a relaxation of full immunity, based on ideas from Goel et al. [5].

Definition 6. A PB mechanism F is called *approximately immune to manipulation by K-Paretian voters* (resp. *K-averagist voters*) for some non-empty $K \subseteq R$, if for any profile $\boldsymbol{A}$ and K-Paretian (resp. K-averagist) voter i, there exists some $p^* \in P$ such that i weakly prefers $F(\boldsymbol{A}_{-i}, S_i^*) \cup \{p^*\}$ to $F(\boldsymbol{A})$.

3 Expressing Additional Constraints

As mentioned in Section 1, part of the richness of the multi-resource PB model comes from it being able to express additional constraints. We consider three of these constraints here, and show how our model captures them.

Incompatibility constraints These express the impossibility of realising some projects simultaneously (e.g. two buildings with overlapping requirements). These constraints can be seen as sets $L \subseteq P$ of at least two projects that are not simultaneously contained in any feasible outcome. Our model can capture such constraints L by introducing an additional resource r_L. This resource is assigned budget $b_{r_L} := |L| - 1$, and each project p is given cost $c_{r_L}(p) := 1$ if $p \in L$, and $c_{r_L}(p) := 0$ otherwise. This way, not all projects in L can be realised simultaneously.

Dependency constraints These express that some projects can only be realised when some other projects are as well (e.g. a shopping mall that can not be built without a parking lot). These constraints can be seen as pairs of non-empty disjoint sets $L, M \subseteq P$ such that L is contained in a feasible outcome only if M is as well. We capture this in our model by introducing an additional resource $r_{L,M}$. This resource is given a budget $b_{r_{L,M}} := (|L| - 1) \cdot |M|$, and projects p are given cost $c_{r_{L,M}}(p) := |M|$ if $p \in L$, cost $c_{r_{L,M}}(p) := -1$ if $p \in M$, and $c_{r_{L,M}}(p) := 0$ otherwise. This enforces that outcomes that include *all* of L also require all of M in order to be feasible.

Distributional constraints These impose upper bounds on the amount of funding which can be allocated to specific categories of projects (e.g. education or healthcare) [3]. These constraints can be seen as categories of projects $C \subseteq P$ with upper bound a on resource $j \in R$. Our model can be easily seen to capture this by yet again introducing an additional resource $r_{C,j}$ with budget $b_{r_{C,j}} := a$ and costs $c_{r_{C,j}}(p) := c_j(p)$ if $p \in C$ and $c_{r_{C,j}}(p) := 0$ otherwise.

Note that it is not at all clear how these constraints should be expressed in the single-resource model, if they can even be expressed there at all, while their encoding in the multi-resource model is quite simple.

4 Axiomatic Properties of PB Mechanisms

In this section we test the greedy- and max-approval mechanisms as defined in Section 2 w.r.t. the defined axioms. First off, discount-monotonicty.

Theorem 1. *Both greedy- and max-approval are discount-monotonic.*

Proof. Consider a multi-resource PB election $\langle P, \boldsymbol{c}, \boldsymbol{b}\rangle$ with voters N, profile $\boldsymbol{A}$ and an alternative cost vector $\boldsymbol{c}'$ satisfying conditions i. to iii. of Definition 1 with respect to some $p \in P$. The discount-monotonicity of F_{greedy} follows almost immediately from its procedure, since the approval score of p does not change when switching to $\boldsymbol{c}'$. For $F_{\max}$, suppose $p \in S := F_{\max}(\boldsymbol{A}, \langle P, \boldsymbol{c}, \boldsymbol{b}\rangle)$. Note that $S \in \textsc{Feas}(P, \boldsymbol{c}', \boldsymbol{b})$, since costs do not increase under $\boldsymbol{c}'$. As the costs of projects other than p stay the same, any outcome $S' \in \textsc{Feas}(P, \boldsymbol{c}', \boldsymbol{b})$ not containing p is also in $\textsc{Feas}(P, \boldsymbol{c}, \boldsymbol{b})$. This means that the approval score of such S' is at most that of S, and that any ties between them and S are broken in the latter outcome's favour. It immediately follows that $p \in F_{\max}(\boldsymbol{A}, \langle P, \boldsymbol{c}', \boldsymbol{b}\rangle)$, since no $S' \in \textsc{Feas}(P, \boldsymbol{c}', \boldsymbol{b})$ without p can beat S. □

Sadly, the results from this point onward are less positive.

Theorem 2. *Neither greedy- nor max-approval is proportional.*

Proof. We show a stronger result: for every $d \geqslant 1$, there exists a d-resource PB election forming a counterexample against the proportionality of both mechanisms. Consider the election with projects $P = \{p_1, p_2\}$, budgets $b_j = 3$, and costs $c_j(p_1) = 1$ and $c_j(p_2) = 3$ for all $j \in R$. Three voters participate, giving rise to the profile $\boldsymbol{A} = \langle \{p_1\}, \{p_2\}, \{p_2\}\rangle$. Then voter 1 approves of a project set costing $\frac{1}{3}$ for each $j \in R$, and as such her chosen project should be part of the outcome. But $F_{\text{greedy}}(\boldsymbol{A}) = F_{\max}(\boldsymbol{A}) = \{p_2\}$. □

We now turn towards the susceptibility of the two mechanisms to strategic manipulation.

Theorem 3. *There is no K for which either greedy- or max-approval is immune to manipulation by either K-Paretian or K-averagist voters.*

Proof. We show that for every $d \geqslant 1$, there exists a d-resource PB election such that for all non-empty $K \subseteq R$, some K-Paretian voter can successfully manipulate the election - the result for averagists follows from this. Consider projects $P := \{p_1, p_2, p_3\}$, budgets $b_j := 3$ and costs defined as $c_j(p_1) = c_j(p_3) := 1$ and $c_j(p_2) := 2$ for all $j \in R$. Two voters participate, the first of whom is a R-Paretian voter with truly preferred outcome $S_1^* := \{p_2, p_3\}$. The ballots they submit give rise to the profile $\boldsymbol{A} = \langle \{p_1, p_2\}, \{p_1, p_3\}\rangle$. Note that $F_{\text{greedy}}(\boldsymbol{A}_{-1}, S_1^*) = F_{\max}(\boldsymbol{A}_{-1}, S_1^*) = \{p_1, p_3\}$, while $F_{\text{greedy}}(\boldsymbol{A}) = F_{\max}(\boldsymbol{A}) = \{p_1, p_2\}$ (keeping tie-breaking in mind). We see for either $F \in \{F_{\text{greedy}}, F_{\max}\}$ and any $j \in R$ that $c_j(F(\boldsymbol{A}_{-1}, S_1^*) \cap S_1^*) = 1$, while $c_j(F(\boldsymbol{A}) \cap S_1^*) = 2$, showing that voter 1 strongly prefers the manipulated outcome. □

Note that the election in the proof of Theorem 3 sees voters submitting feasible ballots, showing that restricting voters in such a way still offers no immunity to manipulation. There are however other restricted elections that fare better.

Theorem 4. *Both greedy- and max-approval are immune to manipulation by both K-Paretian and K-averagist voters for any K, when restricting to elections with unit costs for all projects and resources.*

Proof. We show immunity to manipulation by K-averagists - the result for K-Paretian voters follows as a result. Our arguments work for either $F \in \{F_{\text{greedy}}, F_{\max}\}$. **'Weakly prefers' klopt hier niet** Under our restriction, both mechanisms choose the $\min\{b_1, \ldots, b_d\}$ most approved projects, keeping tie-breaking in mind. Now suppose for the sake of contradiction that some K-averagist voter i weakly prefers $F(\boldsymbol{A})$ to $F(\boldsymbol{A}_{-i}, S_i^*)$ for some profile $\boldsymbol{A}$. Since all projects have unit costs, it follows that $|F(\boldsymbol{A}_{-\boldsymbol{i}}, S_i^*) \cap S_i^*| < |F(\boldsymbol{A}) \cap$

$S_i^*|$. So there is some $p \in S_i^*$ that is in $F(\boldsymbol{A})$ but not in $F(\boldsymbol{A_{-i}}, S_i^*)$. Furthermore, since $|F(\boldsymbol{A})| = |(F(\boldsymbol{A_{-i}}, S_i^*)|$, there is also some project $p' \notin S_i^*$ that is in $F(\boldsymbol{A_{-i}}, S_i^*) \setminus F(\boldsymbol{A})$. It follows that the difference in approval score between p' an p is higher for $\langle \boldsymbol{A}_{-i}, S_i^* \rangle$ than for $\boldsymbol{A}$. But this is contradictory, since projects in S_i^* are approved at least as often in $\langle \boldsymbol{A}_{-i}, S_i^* \rangle$ as in $\boldsymbol{A}$, while the converse holds for projects not in S_i^*. □

While we now know that these mechanisms are generally not immune to manipulation except for in some restricted cases, one could ask whether they are possibly approximately immune in the general case. This does not hold, unfortunately.

Theorem 5. *There is no K for which either greedy- or max-approval is approximately immune to manipulation by either K-Paretian or K-averagist voters.*

Proof. We show that for every $d \geqslant 2$, there is some d-resource PB election such that for all non-empty $K \neq R$ there exists some K-Paretian voter that can successfully manipulate the election. The result for averagists follows from this again.

Consider the election with projects and costs as defined in Table 1, with budgets $b_j := 10$ for all $j \in R$. Three voters participate, the first of whom is K-Paretian. The truly preferred outcome of voter 1 along with the ballots submitted by all voters is in Table 2.

Table 1: Projects and costs of the election in the proof of Theorem 5.

Project	Cost for all $j \in K$	Cost for all $j \in R \setminus K$
p_1	1	2
p_2	1	2
p_3	0	3
p_4	0	3
p_5	4	2
p_6	4	2

Table 2: Ballots in the proof of Theorem 5.

S_1^*	A_1	A_2	A_3
p_1	p_3	p_1	p_3
p_2	p_4	p_2	p_4
p_5	p_5	p_3	p_5
p_6	p_6	p_4	p_6

The costs for resources in $R \setminus K$ enforce that no feasible outcome contains more than 4 projects. It follows for either $F \in \{F_{\text{greedy}}, F_{\max}\}$ that $F(\boldsymbol{A_{-1}}, S_1^*) = \{p_1, p_2, p_3, p_4\}$ while $F(\boldsymbol{A}) = \{p_3, p_4, p_5, p_6\}$. But then it follows for all $j \in K$ and $p^* \in P$ that $c_j((F(\boldsymbol{A}_{-1}, S_1^*) \cap S_1^*) \cup \{p^*\}) = c_j(\{1, 2\} \cup \{p^*\}) \leqslant 6$, while $c_j(F(\boldsymbol{A}) \cap S_1^*) = c_j(\{5, 6\}) = 8$. This shows that voter 1 strongly prefers the manipulated outcome. □

5 Computational Properties of PB Mechanisms

In this section, we analyse the complexity of the computation of the greedy- and max-approval outcomes in the multi-resource setting. We assume the reader to be familiar with the basic notions of computational complexity (particularly weak and strong NP-completeness, see Arora and Barak [1] and Garey and Johnson [4]). While computing the greedy-approval outcome of an election is straightforward and possible in polynomial time, the same cannot necessarily be said of max-approval.

In order to study the complexity of max-approval, we first formulate it as the following decision problem which is at most as hard as the original optimisation problem.

MULTI-MAX-APPROVAL
Input: A d-resource PB election $\langle N, P, \boldsymbol{c}, \boldsymbol{b} \rangle$ for some $d \geqslant 1$, a preference profile $\boldsymbol{A}$, and an approval goal $k \leqslant n \cdot |P|$.
Question: Is there some $S \in \text{FEAS}(P, \boldsymbol{c}, \boldsymbol{b})$ such that $\sum_{i \in N} |S \cap A_i| \geqslant k$?

Perhaps unsurprisingly, this problem is strongly NP-complete when we place no restrictions on d.

Theorem 6. MULTI-MAX-APPROVAL *is strongly* NP-*complete, even when restricted to elections with non-negative costs.*

Proof. Membership in NP is obvious (use the set S as certificate). We show NP-hardness by reduction from a variation on the NP-hard problem IND-SET (see Garey and Johnson [4] for this specific variation), asking one to determine whether an undirected and connected graph $G = \langle V, E \rangle$ with $E = \{e_1, \ldots, e_d\}$ contains an independent set of size at least k for some $K \leqslant |V|$.

Noting that independent sets require vertices sharing edges to be incompatible with one another, we apply the idea behind modelling incompatibility constraints from Section 3 here. Define an $|E|$-resource PB election with a single voter, projects $P := V$, budgets $b_j := 1$, and costs defined for all $v \in P$ as $c_j(v) := 1$ if $v \in e_j$ and $c_j(v) := 0$ otherwise. Let the voter's ballot be $A := P$, and set the approval goal to $k := K$. Clearly this reduction is correct: the feasible outcomes are precisely those subsets of V in which no two vertices are connected by an edge, and the approval score of any outcome is just its size. Since this reduction is obviously polynomial-time computable, this proves the problem is NP-hard. It is in fact strongly NP-hard, since the magnitudes of all numerical parameters are clearly polynomially bounded by the instance size. □

While this may seem discouraging, it only implies that there cannot be a polynomial-time algorithm that computes the max-approval outcome for *all* multi-resource PB elections, assuming P $\neq$ NP. When we fix the amount of resources d, we obtain a slightly better result: weak NP-completeness. Weak completeness results are more promising, since they leave open the possibility of there being a pseudo-polynomial algorithm for the problem.

Theorem 7. MULTI-MAX-APPROVAL *with fixed d is weakly* NP*-complete for any $d \geqslant 1$, even when restricted to elections with non-negative costs.*

Proof. We show this for the case of $d = 1$. The result for higher values of d follows by reducing the problem for a single resource to the problem with d resources by adding $d-1$ resources with unit budgets and costs 0.

Membership in NP is again trivial. We show NP-hardness by reduction from KNAPSACK (again see Garey and Johnson [4] for proof of NP-hardness), which is the problem of determining whether a set of items U with values $v(u) \in \mathbb{Z}^+$ and sizes $s(u) \in \mathbb{Z}^+$ for $u \in U$ contains a subset U' with total size $\sum_{u \in U'} s(u) \leqslant B$ and value $\sum_{u \in U'} v(u) \geqslant K$, where $B, K \in \mathbb{Z}^+$ are the knapsack capacity and target value, respectively. We define a single-resource PB election with voters $N := \{\langle u, i\rangle\,; u \in U, 1 \leqslant i \leqslant v(u)\}$, projects $P := U$, cost function $c := s$, and budget $b := B$. The approval goal is $k := K$. We let each voter $\langle i, u\rangle$ submit the singleton ballot $A_{i,u} := \{u\}$. It is easy to see that feasibility in this election encodes knapsack capacity, while approval scores encode total value. Thus, this polynomial-time reduction shows NP-completeness. This proof however only shows weak completeness, since the value of the costs $c(p) = s(p)$ is not polynomially bounded by the max-approval instance size. □

Now that we have shown that computing the max-approval outcome for fixed d is weakly NP-complete, we move on to the question of whether there in fact is a pseudo-polynomial algorithm for doing so. As shown by Talmon and Faliszewski [10], there exists a pseudo-polynomial dynamic programming algorithm in the single-resource setting which is based on similarities between max-approval and the knapsack problem. Under the restriction to non-negative costs, these similarities extend to the multi-resource setting with the multi-dimensional knapsack problem MULTI-KNAPSACK. This problem is defined analogously to the one considered in Theorem 7, but instead with d knapsack capacities and d sizes for each item, allowing non-negative costs, values and knapsack capacities.

Lemma 1. MULTI-MAX-APPROVAL *(with some fixed d) reduces to* MULTI-KNAPSACK *(with the same d) in polynomial time, with the restriction that all costs are non-negative.*

Proof. Given a d-resource PB election $\langle N, P, \boldsymbol{c}, \boldsymbol{b}\rangle$ with approval profile $\boldsymbol{A}$ and approval goal k, we define a d-dimensional knapsack instance as follows. Let the set of items be $U := P$, the j-th size function be $s_j := c_j$, and the j-th knapsack capacity be $B_j := b_j$. Additionally, we define the target value as $K := k$ and the value function by putting $v(u) := |\{i \in N\,; u \in A_i\}|$. Knapsack capacities in this constructed instance encode feasibility, while total values encode approval scores, showing the correctness of the reduction, which is clearly computable in polynomial time. □

Given the reduction in Lemma 1, we can now show that there is a pseudo-polynomial algorithm computing the max-approval outcome of d-resource PB elections with fixed d and non-negative costs, by using a pseudo-polynomial DP algorithm solving multi-dimensional knapsack instances, described by Kellerer, Pferschy, and Pisinger [7].

Theorem 8. *A max-approval outcome of a d-resource PB election with m projects, n voters, and highest budget $b_{\max}$ can be computed in pseudo-polynomial time $O(m(n+b_{\max}^d))$, when restricted to elections with non-negative costs.*

Proof. As explained, we apply the reduction from Lemma 1 to the election. Since the DP algorithm is defined for positive knapsack capacities, we need to do some pre-processing to enforce that our election has positive budgets. So for every resource $j \in R$ with $b_j = 0$, we remove all projects p with $c_j(p) > 0$, since these could not be part of any feasible outcome. Afterwards, we change the budget of these resources to 1. Note that the resulting election still has some projects left, since we assume there to be at least one feasible outcome in each election. This suffices for us to be able to apply the reduction. We then apply the DP algorithm to the resulting knapsack instance, which correctly computes the max-approval outcome by virtue of our reduction.

We leave out a description of the full algorithm (consisting of pre-processing, the reduction and the DP) for the sake of brevity, and instead describe the intuition behind the main part. The DP algorithm recursively computes for each $1 \leqslant i \leqslant m$ and for all partial budgets for each resource, the highest achievable approval score for feasible outcomes w.r.t these partial budgets, using only the first i projects in P. Based on all these scores, the algorithm can then determine which projects were needed to obtain the optimal approval score using all projects and the full budgets.

The DP algorithm runs in time $O(|U| \cdot B_{\max}^d)$, where U is the set of items and $B_{\max}$ is the highest knapsack capacity. In the knapsack instance constructed by the reduction, we have that $U = P$ and $B_{\max} = b_{\max}$. Since the reduction can easily be seen to run in time $O(mn)$, we conclude that the combination of pre-processing, the reduction and DP algorithm runs in time $O(m(n + b_{\max}^d))$. □

6 Conclusion

We have presented a preliminary formal analysis of multi-resource PB. Providing an intuitive translation of various constraints, we have shown how the multi-resource setting can facilitate a more self-contained implementation of PB. We have further provided an example of how axiomatic and complexity-theoretic analyses can be realised in this setting, by applying these to greedy- and max-approval.

However, our definitions and results still provide room for sharpening and general improvement, to potentially be addressed in future work. For instance, our definition of averagists uses budgets to assess the relevance of resource spending, while more complex measures might be more adequate. Additionally, most of the defined axiomatic properties are not satisfied by either of the analysed mechanisms, requiring further research to find which mechanisms do. Also of interest is bypassing the restriction to non-negative costs in our max-approval algorithm, considering negative costs are needed to capture dependency constraints. Setting these weaknesses aside, the model also has room for extensions and variations. For example, elections could see voters submitting other types of ballots like project rankings, or they could allow projects to be partially realised.

References

[1] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. 1st ed. New York, NY, USA: Cambridge University Press, 2009. ISBN: 9780521424264.

[2] Haris Aziz, Barton E. Lee, and Nimrod Talmon. "Proportionally Representative Participatory Budgeting: Axioms and Algorithms". In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. Richland, SC, USA: International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 23–31.

[3] Haris Aziz and Nisarg Shah. *Participatory Budgeting: Models and Approaches*. 2020. arXiv: 2003.00606v1 [cs].

[4] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. 1st ed. New York, NY, USA: W. H. Freeman & Company, 1979. ISBN: 0716710447.

[5] Ashish Goel, Anilesh K Krishnaswamy, Sukolsak Sakshuwong, and Tanja Aitamurto. "Knapsack Voting for Participatory Budgeting". In: *ACM Transactions on Economics and Computation* 7.2 (2019), pp. 1–27. DOI: 10.1145/3340230.

[6] Benjamin Goldfrank. "Lessons from Latin America's Experience in Participatory Budgeting". In: *Participatory Budgeting*. Ed. by Anwar Shah. Washington, DC, USA: World Bank, 2007, pp. 91–126.

[7] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack Problems*. 1st ed. New York, NY, USA: Springer-Verlag Berlin Heidelberg, 2004, pp. 235–236, 248–249. ISBN: 9783642073113. DOI: 10.1007/978-3-540-24777-7.

[8] Jonathan Rose and Annette Omolo. *Six Case Studies of Local Participation in Kenya*. Washington, DC, USA, 2013. URL: https://openknowledge.worldbank.org/handle/10986/17556.

[9] Anwar Shah, ed. *Participatory Budgeting*. Washington, DC, USA: World Bank, 2007. DOI: 10.1596/978-0-8213-6923-4. URL: https://elibrary.worldbank.org/doi/abs/10.1596/978-0-8213-6923-4.

[10] Nimrod Talmon and Piotr Faliszewski. "A Framework for Approval-Based Budgeting Methods". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. Association for the Advancement of Artificial Intelligence, 2019, pp. 2181–2188.

[11] Alan D. Taylor. "The Manipulability of Voting Systems". In: *The American Mathematical Monthly* 109.4 (2002), pp. 321–337. DOI: 10.1080/00029890.2002.11920895.

A Appendix: Full Proof of Theorem 8

Proof. As explained, we apply the reduction from Lemma 1 to the election. Since the DP algorithm is defined for positive knapsack capacities, we need to do some pre-processing to enforce that our election has positive budgets. So for every resource $j \in R$ with $b_j = 0$, we remove all projects p with $c_j(p) > 0$, since these could not be part of any feasible outcome. Afterwards, we change the budget of these resources to 1. Note that the resulting election must still have some projects left, since we assume there to be at least one feasible outcome in each election. This suffices for us to be able to apply the reduction.

Applying the DP algorithm described by Kellerer, Pferschy, and Pisinger [7] to the d-dimensional knapsack instance obtained through the reduction, we obtain the following algorithm. Suppose the projects in P are enumerated as $P = \{p_1, \dots, p_m\}$. We construct a $(d+1)$-dimensional table Z of size $(m+1) \times (b_1+1) \times \cdots \times (b_d+1)$. Intuitively, we do this by recursively computing in Z the highest approval score $Z[i, g_1, \dots, g_d]$ we can obtain using the projects $p_1, \dots, p_i$ for $1 \leqslant i \leqslant m$ such that their total costs respect the sub-budgets $g_1, \dots, g_d$.

The table Z is initialised by setting $Z[0, g_1, \dots, g_d] := 0$ for all $0 \leqslant g_1 \leqslant b_1, \dots, 0 \leqslant g_d \leqslant b_d$. If $c_j(p_i) > g_j$ for some $j \in R$, we set $Z[i, g_1, \dots, g_d]$ to

$$Z[i-1, g_1, \dots, g_d].$$

If $c_j(p_i) \leqslant g_j$ for all $j \in R$, we set $Z[i, g_1, \dots, g_d]$ to

$$\begin{aligned} \max\{&Z[i-1, g_1, \dots, g_d], \\ &V(p_i) + Z[i-1, g_1 - c_1(p_i), \dots, g_d - c_d(p_i)]\}, \end{aligned}$$

where $V(p_i)$ is the approval score of p_i. It is easily seen that this recursion is well-defined, and matches the earlier-mentioned intuitions.

After filling the entire table, we now go on to compute the outcome with the optimal approval score. We construct another $(d+1)$-dimensional table X, this time of size $m \times b_1 \times \cdots b_d$. For each $1 \leqslant i \leqslant m$ and $1 \leqslant g_1 \leqslant b_1, \dots, 1 \leqslant g_d \leqslant b_d$, we set $X[i, g_1, \dots, g_d]$ to

$$\begin{cases} 0 & \text{if } Z[i, g_1, \dots, g_d] = Z[i-1, g_1, \dots, g_d] \\ 1 & \text{otherwise.} \end{cases}$$

In other words, $X[i, g_1, \dots, g_d]$ is set to 1 iff the project p_i is not part of the outcome achieving approval score $Z[i, g_1, \dots, g_d]$ with budgets $g_1, \dots, g_d$.

We then iteratively construct the optimal solution S by looping through X in the following way, starting from $i := m$ and $g_j := b_j$ for all $j \in R$. If $X[i, g_1, \dots, g_d] = 1$ we place project p_i in S, and proceed with $i := i - 1$ and $g_j := g_j - c_j(p_i)$ for all $j \in R$. If $X[i, g_1, \dots, g_d] = 0$, we do not place p_i in S and instead proceed with $i := i - 1$ and $g_j := g_j$ for all $j \in R$. By construction of X, this process guarantees that S is a max-approval outcome.

Computing the approval scores V requires time $O(mn)$, while the tables Z and X both require time $O(mb^d_{\max})$, and the final construction of S^* requires time $O(m)$. Putting this all together, we see that the algorithm runs in time $O(m(n + b^d_{\max}))$. $\square$