Participatory Budgeting with Multiple Degrees of Projects and Ranged Approval Votes

Gogulapati Sreedurga
Indian Institute of Science
gogulapatis@iisc.ac.in

Abstract

In an indivisible participatory budgeting (PB) framework, we have a limited budget that is to be distributed among a set of projects, by aggregating the preferences of voters for the projects. All the prior work on indivisible PB assumes that each project has only one possible cost. In this work, we let each project have a set of permissible costs, each reflecting a possible degree of sophistication of the project. Each voter approves a range of costs for each project, by giving an upper and lower bound on the cost that she thinks the project deserves. The outcome of a PB rule selects a subset of projects and also specifies their corresponding costs. We study different utility notions and prove that the existing positive results when every project has exactly one permissible cost can also be extended to our framework where a project has several permissible costs. We also analyze the fixed parameter tractability of the problem. Finally, we propose some important and intuitive axioms and analyze their satisfiability by different PB rules. We conclude by making some crucial remarks.

1 Introduction

Participatory budgeting (PB) is a democratic voting paradigm that aggregates the opinions of citizens while deciding on how to fund the public projects [Cabannes, 2004; Shah, 2007; Sintomer *et al.*, 2008; Wampler, 2010; Röcke, 2014].

Divisible PB assumes that the costs of the projects are totally flexible and any amount can be allocated to each of them [Airiau *et al.*, 2019; Sreedurga *et al.*, 2022b]. The existing work on indivisible PB, on the other hand, assumes that every project has a fixed cost that is to be allocated to the project if it is selected. However, many times in real-world, each project can be implemented upto different levels of sophistication. For example, a building can be built with wood or stone, depending on the amount allocated to it. That is, each project has a set of possible and permissible costs. Each cost in this set corresponds to a degree of sophistication of the project. In addition to choosing the set of projects that are to be funded, the mechanism designer must also choose a permissible cost

for each of the selected projects. This idea of having multiple permissible costs for each project was first proposed in a survey by Aziz and Shah [Aziz and Shah, 2021], in which the authors called each of these costs a *degree of completion*. However, the idea still remained to be studied and our paper bridges this gap.

Preference elicitation methods typically studied in any voting framework include approval votes, ordinal votes, and cardinal votes. These methods also continue to be the most studied preference elicitation approaches in PB [Aziz *et al.*, 2018; Talmon and Faliszewski, 2019; Rey *et al.*, 2020; Jain *et al.*, 2020; Fairstein *et al.*, 2022; Sreedurga *et al.*, 2022a; Benade *et al.*, 2021; Shapiro and Talmon, 2017; Aziz and Lee, 2021; Sreedurga and Narahari, 2022; Peters *et al.*, 2020]. However, PB is a setting with several attributes like costs attached to the projects. The preferences and utilities of the voters are thus much more complex. This propels the need to devise preference models specific to PB, as pointed out by Aziz and Shah [Aziz and Shah, 2021]. One step in this direction is the introduction of a special case of approval votes, called knapsack votes, by Goel et al. [Goel *et al.*, 2019], where each voter reports her most favorite budget division. This idea however is criticised for its assumption that any project not in this division yields no utility to the voter.

Our paper is another step in the direction of preference modeling for PB. We introduce the approach of *ranged approval votes*, which strictly generalizes the approval votes. Each voter reports a lower bound and an upper bound on the cost that she thinks each project deserves. All the bounds are initially set to 0 by default. Voting proceeds in two steps. In the first step, the voter starts by approving the projects she likes. For these approved projects, only the upper bounds will automatically change to the highest permissible cost. In the second step, the voter may *optionally* change bounds for some of these approved projects, if she wishes to have a say on the amount they deserve. Note that this is cognitively not much more demanding than the approval votes since we do not force the voters to report the bounds. Notably, all our computational results can also be extended to more general utility functions with minor tweaks. Nevertheless, we present the results for ranged approval votes due to their cognitive simplicity and natural relevance in the model.

It needs to be mentioned that a work by Goel et al. [2019] views divisible PB as a model of indivisible PB where every

PROPERTIES\RULES	$R_{\|S\|}$	$R_{\mathrm{c}(S)}$	$R_{\widehat{\mathrm{c}(S)}}$	$R_{\|\ \|}$
Degree-efficient	×	×	✓	×
Shrink-resitant	✓	✓	✓	✓
Discount-proof	✓	×	×	×
Range-abiding	✓	×	×	✓
Range-converging	✓	✓	✓	✓
Range-unanimous	✓	×	×	✓
Upper bound-sensitive	✓	✓	×	✓
Lower bound-sensitive	×	×	×	✓

Table 1: Results for budgeting axioms in Section 4

unit of money in the cost corresponds to one possible degree of sophistication, and proposes a greedy algorithm. Their work however imposes knapsack constraint on each vote as well as assumes all lower bounds to be zero. Their model hence forms a very restricted special case of ours.

Our contributions. The primary goal of this paper is to systematically study the PB model where each project has a set of permissible costs. Such a study has been conducted by Talmon and Faliszewski [Talmon and Faliszewski, 2019] for approval-based PB, which is a special case of our model with every project having only one permissible cost. We generalize the PB rules and all the positive results in [Talmon and Faliszewski, 2019] to our model. Namely, we propose polynomial-time, pseudo-polynomial time, and PTAS algorithms. Followed by this, we present some results on parameter tractability using a recently introduced parameter scalable limit [Sreedurga *et al.*, 2022a] and another novel parameter we introduce in the further sections. It needs to be highlighted that, as a part of our study, we also introduce and investigate some novel utility notions that are specific to our model and are not extensions of the literature. Finally, we propose some budgeting axioms for our model and examine their satisfiability with respect to our PB rules. All the axiomatic results are summarized in Table 1. We conclude by discussing the impact of our computational and axiomatic results and pointing out future directions.

Organization of the paper. We start by introducing the mathematical model in Section 2. We define different utility notions (some extended from the existing ones in the literature and some new) and the corresponding PB rules in Section 2.1. In Section 3, we analyze the computational complexity of our PB rules and suggest ways to cope up with intractabilities. In Section 4, we define budgeting axioms for our model and examine their satisfiability.

2 Model

In our model, a budget b is given. We denote the set of n voters by $N = \{1, \ldots, n\}$ and the set of m projects by P. Each project $j \in P$ has t_j possible degrees of sophistication captured by the set $\mathrm{D}(P_j) = \{P_j^0, P_j^1, \ldots, P_j^{t_j}\}$. The cost of each degree P_j^t is indicated by c_j^t. We assume that c_j^0 is zero for all $j \in P$ and it corresponds to not funding the project j.

Each voter $i \in N$ reports for every project j, a lower bound $l_i(j)$ and an upper bound $h_i(j)$ such that $l_i(j), h_i(j) \in \{c_j^0, \ldots, c_j^{t_j}\}$ and $l_i(j) \leq h_i(j)$. Let $\mathcal{L}$ and $\mathcal{H}$ respectively

denote the collection all the lower bounds and upper bounds reported by all the voters.

Let $\mathcal{D}$ denote the set of all the possible degrees of all projects, or in other words, $\mathcal{D} = \bigcup_{j \in P} \mathrm{D}(P_j)$. We denote the cost of a set $S \subseteq \mathcal{D}$, $\sum_{P_j^t \in S} c_j^t$, by $\mathrm{c}(S)$. Given a subset $S \subseteq \mathcal{D}$, we use $S(j)$ to denote the chosen degree(s) of project j in S. In other words, $S(j) = S \cap \mathrm{D}(P_j)$. We use the shorthand notation $c^S(j)$ to denote $\mathrm{c}(S(j))$. We say a subset $S \subseteq \mathcal{D}$ is ***valid*** if $\mathrm{c}(S) \leq b$ and $|S(j)| = 1$ for every $j \in P$. Let $\mathcal{V}$ denote the collection of all the valid subsets.

A PB instance with multiple degrees of sophistication is denoted by $I = \langle N, \mathcal{D}, \mathrm{c}, b, \mathcal{L}, \mathcal{H} \rangle$. Given an instance, the objective of a PB rule R is to output a valid subset $S \in \mathcal{V}$.

2.1 The PB Rules

The PB rules we study are utilitarian rules. This implies that, given a function u that measures utility a voter derives from a subset of projects, the corresponding PB rule outputs a valid set of projects that maximizes the sum of utilities of all the voters, i.e., it outputs some $S \in \mathcal{V}$ such that $\sum_{i \in N} u_i(S)$ is maximized. We say a subset S of projects is selected under a PB rule R if it maximizes the total utility of the voters. We say a project $P_j^t \in \mathcal{D}$ ***wins*** under a PB rule R if it belongs to some set that can be selected under R. Let $R(I)$ be the collection of all the projects that win under the PB rule R, given an instance I.

Now, we only need to define different utility functions. Given the lower and upper bounds reported by the voters, we define the utility of a voter from a valid set $S \in \mathcal{V}$ in four ways. For each of these utility functions, we also give a shorthand notation for the utilitarian PB rule associated to it.

1. **Cardinal utility ($R_{\|S\|}$ rule)**: Each voter i derives a utility of 1 from a project j if the cost of the chosen degree falls within the bounds specified by the voter. Thus, $u_i(S) = |j \in P : l_i(j) \leq c^S(j) \leq h_i(j)|, c^S(j) \neq 0$.

2. **Cost utility ($R_{\mathrm{c}(S)}$ rule)**: A voter i derives a utility of $c^S(j)$ from a project j if its value falls within the bounds specified by the voter.
$$u_i(S) = \sum_{j : l_i(j) \leq c^S(j) \leq h_i(j)} c^S(j).$$

3. **Cost capped utility ($R_{\widehat{\mathrm{c}(S)}}$ rule)**: Each voter i derives a utility of $c^S(j)$ from a project j if the value falls within the bounds reported by her, a utility of $h_i(j)$ if $c^S(j) > h_i(j)$, and a utility of 0 if $c^S(j) < l_i(j)$. That is, with slight abuse of notation, $u_i(S) = \sum_{j \in P} u_i(S, j)$, where $u_i(S, j)$ is defined as:
$$u_i(S, j) = \begin{cases} 0 & c^S(j) < l_i(j) \\ c^S(j) & l_i(j) \leq c^S(j) \leq h_i(j) \\ h_i(j) & \text{otherwise} \end{cases}$$

4. **Distance disutility ($R_{\|\ \|}$ rule)**: From every project j, each voter derives a disutility of 0 if the value falls within the bounds reported by her, a disutility of $c^S(j) - h_i(j)$ if $c^S(j) > h_i(j)$, and a disutility of $l_i(j) - c^S(j)$ if $c^S(j) < l_i(j)$. That is, with slight abuse of notation, $d_i(S) = \sum_{j \in P} d_i(S, j)$, where $d_i(S, j)$ is defined as

follows:

$$d_i(S,j) = \begin{cases} l_i(j) - c^S(j) & c^S(j) < l_i(j) \\ 0 & l_i(j) \leq c^S(j) \leq h_i(j) \\ c^S(j) - h_i(j) & \text{otherwise} \end{cases}$$

The corresponding PB rule $R_{\|\,\|}$ minimizes the total disutility.

The first two notions of utilities are the natural extensions of the notions considered in [Talmon and Faliszewski, 2019]. The former reflects that the voter is happy as long as the cost allocated to the project is acceptable to her, whereas the latter reflects that as the project gets more money, the voter gets happier if the cost is acceptable to her.

The second utility notion clearly assumes that if the project gets more money than what the voter thinks it deserves, voter derives zero utility. However, this need not always be the case in real-world. For example, say a voter will be happy to have an entertainment park in the neighborhood but feels that the park deserves any amount between 1000 and 5000 units. Now, if the park project is allocated 7000 units, the voter could still be happy that there is a park in the neighborhood but derive no more utility than 5000 (since it is the maximum value she thinks the park deserves). The third utility notion tackles such scenarios by modifying the second utility notion to cap the utility at $h_i(j)$ instead of dropping it to 0.

The first three utility notions assume that allocating any cost outside the range reported by the voter yields the same utility to her. However, in many situations, closer the cost of the project is to the acceptable range of the voter, higher is the satisfaction voter derives from it. For example, say a voter feels that a certain project is worth at least 1000 units. An outcome that allocates 900 units to it is likely to be more preferred by the voter over something that allocates 50 units to it. The fourth rule handles such situations by considering the distance between the cost allocated and the closest acceptable cost as the disutility. Farther the cost is from the acceptable range, higher is the amount misspent.

3 Computational Complexity

Here, we analyze the computational complexity of our PB rules. We strengthen the existing positive results in the literature [Talmon and Faliszewski, 2019] and also present a few new results on fixed parameter tractability. All the exact algorithms we present are based on dynamic programming. The approximation schemes we present depend on both dynamic programming as well as a clever rounding scheme.

3.1 The Rule $R_{|S|}$

Recall that $R_{|S|}$ outputs a valid set that maximizes the sum of the utilities of voters, where the utility of a voter is defined as $u_i(S) = |j \in P : l_i(j) \leq c^S(j) \leq h_i(j)|$. Talmon and Faliszewski [2019] prove that for the case with only one permissible cost for each project, a subset maximizing the utility can be computed in polynomial time. We strengthen this result and prove that even when there are multiple permissible costs for each project, a subset maximizing the total utility can be computed in polynomial time.

Theorem 1. *For any instance I, a subset $S \in \mathcal{V}$ that is selected under $R_{|S|}$ can be computed in polynomial time.*

Proof. We present an algorithm that uses dynamic programming. Let $\mathrm{s}(P_j^t)$ denote the number of voters i such that $l_i(j) \leq c_j^t \leq h_i(j)$. Construct a dynamic programming table such that $A(x,y)$ corresponds to the cost of cheapest valid subset of $\bigcup_{j=1}^{x} \mathrm{D}(P_j)$ such that the total score of projects in the set is exactly y.

Let $F(y) \subseteq \mathrm{D}(P_1)$ such that $\mathrm{s}(P_1^t) = y$ for every $P_1^t \in \mathrm{D}(P_1)$. We compute the first row as: $A(1,y) = \min(c_j^t : P_j^t \in F(y))$. All the remaining rows are computed recursively as follows: $A(x,y) = \min\Big(A(x-1,y)\,,\,\min_{t=1}^{t_x}\big(A(x-1,y-\mathrm{s}(P_x^t)) + c_x^t\big)\Big)$. In addition to computing this table, we store the sets corresponding to each entry in A in a separate table B as follows: if $A(x,y)$ achieves the minimum value at $A(x-1,y)$, we copy $B(x-1,y)$ to $B(x,y)$ and append P_j^0. If $A(x,y)$ achieves the minimum value at $A(x-1,y-\mathrm{s}(P_x^t)) + c_x^t$ for some $t \in [1,t_x]$, we set $B(x,y) = B(x-1,y) \cup \{P_x^t\}$. Finally, we output the set at $B(x,y)$ such that $A(x,y) \leq b$ and y is maximized.

Correctness. Recurrence ensures that for any entry in B, we select at most one project from $\mathrm{D}(P_j)$ for every j. While selecting the outcome from B, we ensured that its cost is within the budget. Hence, the output of the algorithm will be a valid subset. Optimality follows from the way A is defined.

Running time. Each row in A corresponds to making a decision about one project from P, and hence the number of rows is m. Each column corresponds to a possible total score. Since we select only one project from each $\mathrm{D}(P_j)$ and maximum score of any degree of a project is n by definition, maximum total score is mn. Thus, we have mn columns. Computing each entry in row x takes $O(t_x)$ time. Thus, the running time is $O(m^2 n t^*)$, where $t^* = \max_{j \in P} t_j$. □

3.2 The Rule $R_{c(S)}$

Recall that $R_{c(S)}$ outputs a valid set that maximizes the sum of the utilities of voters, where the utility of a voter is:

$$u_i(S) = \sum_{j:l_i(j) \leq c^S(j) \leq h_i(j)} c^S(j).$$

Talmon and Faliszewski [2019] prove that for the case where each project has only one permissible cost, it is NP-hard to determine if there exists a feasible subset that guarantees a total utility of at least a given value. Since their model can be modeled as a special case of ours, the hardness easily follows.

Proposition 1. *For an instance I and a value s, it is NP-hard to check if $R_{c(S)}$ outputs a set with at least a total utility s.*

Proof. Given a PB instance with approval votes (each project j has some cost $c(j)$ and every voter reports a subset A_i of projects that she likes), we construct an instance for $R_{c(S)}$ as follows: For every project $j \in P$, create exactly two degrees such that $c_j^0 = 0$ and $c_j^1 = c(j)$. For each voter i, set $l_i(j) = 0$ for every project j and set $h_i(j) = c_j^1$ if and only if $j \in A_i$. Clearly, both the instances are equivalent and we skip the proof of correctness. □

To cope up with the intractability, Talmon and Faliszewski [2019] prove that if every project has only one permissible

cost, the problem has a pseudo-polynomial time algorithm and FPTAS. We extend both these results to our model with multiple permissible costs.

Proposition 2. *For any instance I, a subset $S \in \mathcal{V}$ selected under $R_{c(S)}$ can be computed in pseudo-polynomial time.*

Proof. We multiply every $\mathrm{s}(P_j^t)$ in the proof of Thm. 1 with c_j^t. DP tables are also constructed as explained in Thm. 1. The maximum score achievable by each P_j^t is nc_j^t. The total score is upper bounded by $nc(S)$ since at most a single project from each D is chosen into each $B(x, y)$. This is bounded by nb since $S \in \mathcal{V}$. Thus, the table size is $O(mnb)$ and the time is $\boldsymbol{O(mnt^*b)}$, where $t^* = \max_{j \in P} t_j$. $\square$

Theorem 2. *There is an FPTAS for $R_{c(S)}$.*

Proof. The idea is inspired from one of the existing FPTAS algorithms of the knapsack problem [Ibarra and Kim, 1975]. We round the scores of all the projects and use the DP table explained in Thm. 1 on the modified instance.

Given an instance I, let M be the maximum score of a degree of project, i.e., $M = \max_{j \in P, t \in [1, t_j]} \mathrm{s}(P_j^t)$. We can easily ensure that $M \leq OPT$ by eliminating all the project degrees that cannot be part of any set with cost within b. Take any $\epsilon \in (0, 1)$. Now, create new scores of all the projects as follows: $\alpha(P_j^t) = \left\lfloor \frac{\mathrm{s}(P_j^t)m}{\epsilon M} \right\rfloor$. We construct the DP tables similar to those in Thm. 1, considering new scores. By the definition of $\alpha(P_j^t)$, for every $j \in P$ and $t \in [1, t_j]$:

$$\alpha(P_j^t) \leq \frac{\mathrm{s}(P_j^t)\, m}{\epsilon M} \qquad (1)$$

$$\mathrm{s}(P_j^t) \leq \frac{\epsilon M(\alpha(P_j^t) + 1)}{m} \qquad (2)$$

Let S be the outcome of this DP algorithm. Say O denotes the optimal solution for I under $R_{c(S)}$. Please note that the DP ensures that S is a valid set. Since S is the optimal solution for the modified scores,

$$\sum_{j=1}^{m} \alpha\left(P_j^{O(j)}\right) \leq \sum_{j=1}^{m} \alpha\left(P_j^{S(j)}\right) \qquad (3)$$

Now, we prove that the approximation factor of $(1 - \epsilon)$ holds.

$$OPT = \sum_{j=1}^{m} \mathrm{s}\left(P_j^{O(j)}\right) \leq \sum_{j=1}^{m} \frac{\epsilon M(\alpha\left(P_j^{O(j)}\right) + 1)}{m}$$
$$\text{(From (2))}$$

$$\leq \sum_{j=1}^{m} \frac{\epsilon M \alpha\left(P_j^{S(j)}\right)}{m} + \epsilon M$$
$$\text{(From (3))}$$

$$\leq \frac{\epsilon M}{m} \sum_{j=1}^{m} \frac{\mathrm{s}\left(P_j^{S(j)}\right) m}{\epsilon M} + \epsilon M$$
$$\text{(From (1))}$$

$$\sum_{j=1}^{m} \mathrm{s}\left(P_j^{S(j)}\right) \geq (1 - \epsilon)OPT \quad (\because M \leq OPT)$$

Running Time: The table has m rows. The modified score of each degree of project is upper bounded by $\frac{\mathrm{s}(P_j^t)m}{\epsilon M}$, which is further bounded by $\frac{m}{\epsilon}$ due to the definition of M. Since the output is a valid set, it has at most m projects and the maximum possible total score is upper bounded by $\frac{m^2}{\epsilon}$, which is the number of columns. Computing each entry in row x takes $O(t_x)$ time. Thus, the running time is $\boldsymbol{O(\frac{m^3 t^*}{\epsilon})}$, where $t^* = \max_{j \in P} t_j$. $\square$

Fixed Parameter Tractability

Recently, Sreedurga et al. [2022a] introduced a new parameter called scalable limit for PB. The authors observed that this value is often small in PB elections in real life (e.g., datasets at https://pbstanford.org/ where budget and all the costs are multiples of some really high value). Motivated by this, we prove that $R_{c(S)}$ is in FPT w.r.t. scalable limit. Before that, we define the scalable limit in our model as follows:

Definition 1. *For any instance I, we refer to* $\frac{\max_{P_j^t \in \mathcal{D}} c_j^t}{GCD(c_1^1, \ldots, c_m^{t_m}, b)}$ *as the **scalable limit** (δ) of the instance.*

Intuitively, we scale down all the costs and budget as much as possible, while ensuring that all these values continue to be integers. The scalable limit then refers to the cost of the costliest degree in $\mathcal{D}$ in this scaled down instance.

Proposition 3. *For any instance I, computing a subset $S \in \mathcal{V}$ that is selected under $R_{c(S)}$ is in FPT w.r.t. scalable limit δ.*

Proof. We again prove this by constructing a DP table. Let $k = \frac{1}{GCD(c_1^1, \ldots, c_m^{t_m}, b)}$. First, we get a new instance I' by scaling down all the costs and budget as follows: $c_j'^t = kc_j^t$ and $b' = kb$. Then, construct the DP table similar to Theorem 1.

Running Time: The table has m rows. The modified cost of each degree of project is upper bounded by δ and hence the maximum score possible from each degree of a project is upperbounded by $n\delta$. Since the output is a valid set, there can be at most m projects in it and hence the maximum possible total score is upper bounded by $mn\delta$, which is the number of columns. Computing each entry in row x takes $O(t_x)$ time and the running time is $\boldsymbol{O(m^2 n\delta t^*)}$. $\square$

Proposition 4. *For any instance I, computing a subset $S \in \mathcal{V}$ that is selected under $R_{c(S)}$ is in FPT w.r.t. m.*

The above proposition follows from the fact the total number of valid subsets is exponential in m (upper bounded by $(t^*)^m$, where $t^* = \max_{j \in P} t_j$) and computing the total utility of any set under $R_{c(S)}$ can be done in polynomial time.

3.3 The Rule $R_{\widehat{c(S)}}$

This rule is similar to the previous rule $R_{c(S)}$, with the only difference being the way a voter is assumed to view the projects allocated higher cost than the approved limit. The utility is defined as $u_i(S) = \sum_{j \in P} u_i(S, j)$, where $u_i(S, j)$ is taken as defined in Section 2.1. The proof of the following result is deferred to the appendix [Sreedurga, 2023].

Proposition 5. *Given an instance I, the following statements hold:*

1. *For any value s, it is NP-hard to determine if $R_{\widehat{c(S)}}$ outputs a set that has a total utility of at least s.*
2. *A subset $S \in \mathcal{V}$ that is selected under $R_{\widehat{c(S)}}$ can be computed in pseudo-polynomial time.*
3. *There is an FPTAS for $R_{\widehat{c(S)}}$.*
4. *Computing a subset $S \in \mathcal{V}$ that is selected under $R_{\widehat{c(S)}}$ is in FPT w.r.t. scalable limit δ.*
5. *Computing a subset $S \in \mathcal{V}$ that is selected under $R_{c(S)}$ is in FPT w.r.t. m.*

3.4 The Rule $R_{\| \|}$

Recall that $R_{\| \|}$ outputs the feasible set that minimizes $d_i(S) = \sum_{j \in P} d_i(S, j)$, where $d_i(S, j)$ is defined as in Section 2.1. It is worth bearing in mind that the results from Section 3.2 and Section 3.3 cannot be transferred since we have a minimization problem and the fact that we cannot upper bound the disutilities by b. We start the section by proving that this rule too is NP-hard, using a slightly different but simple reduction from the same problem (used in Proposition 1) solved by Talmon and Faliszewski [2019]. Then, we give a parameterized approximation algorithm, more specifically a parameterized FPTAS, which guarantees the approximation ratio when the parameter is small [Feldmann *et al.*, 2020].

Proposition 6. *For an instance I and a value s, it is NP-hard to determine if $R_{\| \|}$ outputs a set that has a total disutility of at most s.*

Proof. For a PB instance with approval votes (each project j has some cost $c(j)$ and every voter reports a subset A_i of projects that she likes) and a positive value x, we construct an instance for $R_{c(S)}$ as follows: For every project $j \in P$, create exactly two degrees such that $c_j^0 = 0$ and $c_j^1 = c(j)$. For each voter i, set $h_i(j) = c(j)$ for every project j and set $l_i(j) = c_j^1$ if and only if $j \in A_i$. Let $Z = \sum_{i \in N} c(A_i)$. Set $s = Z - x$.

Correctness. To prove the forward direction, say the approval-based PB instance is a YES instance. That implies, there exists a set S of projects such that $\sum_{i \in N} c(S \cap A_i) \geq x$. Now calculate the disutility of S under $R_{\| \|}$. Then, $d_i(S, j) = c(j)$ for every $j \in A_i \setminus S$ and 0 otherwise. Therefore, $d_i(S) = -c(A_i) - c(A_i \cap S)$. Then, total utility from $S = Z - \sum_{i \in N} c(A_i \cap S)$. Since the former is at least x, this total disutility is at most Z-x. The backward direction can be argued similarly. $\square$

Parameterized Approximation Algorithm (FPTAS)
The very high disutilities that are hard to be bound motivate us to propose a *parameterized FPTAS*, which has been of a great interest recently [Feldmann *et al.*, 2020]. We consider a parameter that we call *variance coefficient*, γ, which, on being given an instance I, intuitively shows how divergent the disutilities of different degrees of projects in the instance are. It is captured by measuring the highest disutility a single project can have, relative to sum of the least possible disutilities from all the projects. We explain this formally below.

Given an instance I, for each degree of a project $P_j^t \in \mathcal{D}$, we use q_j^t to denote the disutility the project will contribute to any set that contains it. We call this *disutility contribution* of

P_j^t. Suppose for example, $P_j^t \in S$. That implies, $c^S(j) = c_j^t$ ($\because S$ is valid). Therefore,
$$q_j^t = \sum_{i:c_j^t < l_i(j)} (l_i(j) - c_j^t) + \sum_{i:h_i(j) < c_j^t} (c_j^t - h_i(j)).$$
It can be observed that, $\sum_{i \in N} d_i(S) = \sum_{P_j^t \in S} q_j^t$.

Let $q_m = \max_{P_j^t \in \mathcal{D}} q_j^t$. That is, q_m is the maximum of disutility contributions of all the degrees of all the projects. We use q_σ to denote the sum of minimum disutility contributions from each $D(P_j)$, i.e., $q_\sigma = \sum_{j \in P} \min_{t \in D(P_j)} q_j^t$. Our parameter γ is the ratio q_m/q_σ. Intuitively, this parameter means that we want the disutility contribution from a single project not to be extremely higher than the sum of the least disutility contributions from all the m projects.

Theorem 3. *For any instance I and $\epsilon \in [0, 1]$, a subset $S \in \mathcal{V}$ such that $\sum_{i \in N} d_i(S) \leq (1 + \epsilon)OPT$ can be computed in $O(\frac{m^3 \gamma t^*}{\epsilon})$ time, where OPT is the optimal possible total disutility under $R_{\| \|}$, γ is the variance coefficient, and $t^* = \max_{j \in P} t_j$.*

Proof. The idea is similar to that in Thm. 2. We define new disutilities for each project by rounding their disutility contributions. We then construct a DP table using which we select a set. For each project P_j^t, we define $\alpha(P_j^t)$ as follows:
$$\alpha(P_j^t) = \lfloor q_j^t m / \epsilon q_\sigma \rfloor.$$

We construct the DP tables A and B similar to those in Thm. 1, with a slight change the each column now represents the total disutility, and $s(P_j^t)$ is replaced by $\alpha(P_j^t)$ defined in the above para. We output the set at $B(x, y)$ such that $A(x, y) \leq b$ and y is *minimized*. Let S be the resultant outcome. Say O denotes the optimal solution for I under $R_{\| \|}$. Please note that the DP ensures that S is a valid set. Since S is the optimal solution w.r.t. new disutilities (i.e., w.r.t. α's),
$$\sum_{j=1}^{m} \alpha\left(P_j^{S(j)}\right) \leq \sum_{j=1}^{m} \alpha\left(P_j^{O(j)}\right) \tag{4}$$
By the definition of α, we have
$$q_j^t \geq \frac{\epsilon q_\sigma \alpha(P_j^t)}{m} \tag{5}$$
$$\alpha(P_j^t) \geq \frac{q_j^t m}{\epsilon q_\sigma} - 1 \tag{6}$$
Now, we prove that the approximation factor of $(1 + \epsilon)$ holds.
$$OPT = \sum_{j=1}^{m} q_j^{O(j)} \geq \sum_{j=1}^{m} \frac{\epsilon q_\sigma \alpha\left(P_j^{O(j)}\right)}{m} \quad \text{(From (5))}$$
$$\geq \sum_{j=1}^{m} \frac{\epsilon q_\sigma \alpha\left(P_j^{S(j)}\right)}{m} \quad \text{(From (4))}$$
$$\geq \left(\frac{\epsilon q_\sigma}{m} \sum_{j=1}^{m} \frac{m q_j^{S(j)}}{\epsilon q_\sigma}\right) - \epsilon q_\sigma \quad \text{(From (6))}$$
$$\sum_{j=1}^{m} q_j^{S(j)} \leq OPT + \epsilon q_\sigma \tag{7}$$

q_σ is the sum of minimum possible disutility contribution from each $D(P_j)$. Any valid set should have one degree (degree 0 corresponds to not funding the project) from each $D(P_j)$. Since the optimal solution has to be a valid set, the optimal disutility must be at least q_σ, i.e., $OPT \geq q_\sigma$. Applying this in Eqn. 7, results in $\sum_{j=1}^{m} q_j^{S(j)} \leq (1 + \epsilon)OPT$.

Running time. Now, we need to check the running time of constructing the DP table. The table has m rows. The disutilities calculated for each degree of project is upper bounded by $\frac{q_j^t m}{\epsilon q_\sigma}$, which is further bounded by $\frac{m\gamma}{\epsilon}$ (by the definition of γ, $\because q_j^t \leq q_m$). Since the output is a valid set, there can be at most m projects in the output and hence the maximum possible total disutility is upper bounded by $\frac{m^2\gamma}{\epsilon}$, which is the number of columns. Computing each entry in row x takes $O(t_x)$ time. Thus, the running time is $\boldsymbol{O\left(\frac{m^3\gamma t^*}{\epsilon}\right)}$, where $t^* = \max_{j \in P} t_j$. □

Fixed parameter tractability. Here, we prove that $R_{\parallel \parallel}$ is also in FPT with respect to the parameters δ or m.

Proposition 7. *For any instance I, computing a subset $S \in \mathcal{V}$ that is selected under $R_{\parallel \parallel}$ is in* FPT *w.r.t. scalable limit δ.*

The proof of the above proposition is similar to the proof of Proposition 3. We scale down the costs of the projects and the budget as explained in that proof. Followed by this, the DP table and the output will be computed as explained in Proposition 7. The running time analysis is exactly the same as explained in Proposition 3

Proposition 8. *For any instance I, computing a subset $S \in \mathcal{V}$ that is selected under $R_{\parallel \parallel}$ is in* FPT *w.r.t. m.*

The above result follows from the fact that when m is constant, we can exhaustively search all the valid sets for the set with optimal disutility.

4 Budgeting Axioms

An enormous amount of work has been done on the axiomatic study of PB with approval votes [Aziz *et al.*, 2018; Aziz and Lee, 2019; Talmon and Faliszewski, 2019; Rey *et al.*, 2020; Baumeister *et al.*, 2020; Sreedurga *et al.*, 2022a]. However, our PB model in which each project has a set of permissible costs is unique technically as well as realistically. This uniqueness demands the development of novel axioms applicable explicitly in such a setting. We introduce different axioms and investigate which of the PB rules introduced in Section 2.1 satisfy our axioms and which do not. We defer some proofs in this section to appendix [Sreedurga, 2023].

The first axiom implies that if the voters are narrowing down their interval towards a degree that was winning, then the degree still continues to win.

Definition 2 (Shrink-resistant). *A PB rule R is said to be shrink-resistant if for any instance I, a voter i, any $j \in P$, it holds that a set S selected under R continues to be selected even if $l_i(j)$ and $h_i(j)$ are shifted closer to $c^S(j)$.*

Proposition 9. *All the four rules $R_{|S|}$, $R_{c(S)}$, $R_{\widehat{c(S)}}$, and $R_{\parallel \parallel}$ are shrink-resistant.*

Given an instance I, for each project $j \in P$, we define $\tau_j = \{c_j^t : P_j^t \in \mathcal{D}, \forall i \in N \ l_i(j) \leq c_j^t \leq h_i(j)\}$. Let $\overline{\tau_j} = \max(\tau_j)$ and $\underline{\tau_j} = \min(\tau_j)$. The second axiom requires that if some range of costs is found to be acceptable unanimously by all the voters, then the cost allocated to the project must not go beyond this range.

Definition 3 (Range-abiding). *A PB rule R is said to be range-abiding if for any instance I, a project $j \in P$, and a set S selected under R, it holds that*
$$\tau_j \neq \emptyset \implies c^S(j) \leq \overline{\tau_j}.$$

Proposition 10. *The rules $R_{|S|}$ and $R_{\parallel \parallel}$ are range-abiding, whereas $R_{c(S)}$ and $R_{\widehat{c(S)}}$ are not.*

Proof. First, we prove that $R_{c(S)}$ and $R_{\widehat{c(S)}}$ are not range-abiding. Recall that b denotes the budget. Say we have a single project j with exactly two permissible costs, respectively equal to $\lfloor \frac{b-1}{n} \rfloor$ and b. Say all the voters set $l_i(j) = c_j^1$. One voter i sets $h_i(j) = c_j^2$, whereas all the remaining voters report c_j^1 as the upper bound. Clearly, in a set S containing P_j^1, the utility from project j is at most $b - 1$. Whereas, if S selects P_j^2, the utility from j is equal to b (but all due to the utility of a single voter). Thus P_j^2 is selected. Note that, in this example, $\tau_j = \lfloor \frac{b-1}{n} \rfloor$ and clearly $c_j^2 > \tau_j$.

Now, assume that $R_{|S|}$ and $R_{\parallel \parallel}$ are not range-abiding. That is, in a set S selected under the rule, $c^S(j) > \overline{\tau_j}$. Consider the set $S' = S \setminus \{S(j)\} \cup \{P_j^t : \overline{\tau_j} = c_j^t\}$. Clearly, S' is valid since $c^S(j) > \overline{\tau_j}$ and S is valid. Also, $u_i(S', j) > u_i(S, j)$ since the degree chosen in S' is within the bounds reported by every voter (it gives the optimal utility of n for $R_{|S|}$ and the optimal disutility of 0 for $R_{\parallel \parallel}$). Thus, S' is a strictly better set and S must not be selected. This forms a contradiction. □

The next axiom requires that increasing the budget should result in the winning degree of some project moving closer to the range of costs for it found to be acceptable unanimously by all the voters.

Definition 4 (Range-converging). *A PB rule R is said to be range-converging if for any instance I, a set S selected for I under R, and a set $S' \neq S$ selected under R on increasing the budget, it holds that whenever there is at least one project k with $\tau_k \neq \emptyset$, there also exists some project $j \in P$ such that:*
$$c^S(j) \notin \tau_j \implies |c^S(j) - \underline{\tau_j}| > |c^{S'}(j) - \underline{\tau_j}|.$$

Proposition 11. *All the four rules $R_{|S|}$, $R_{c(S)}$, $R_{\widehat{c(S)}}$, and $R_{\parallel \parallel}$ are range-converging.*

The next axiom requires that if some range of costs is found to be acceptable unanimously by all the voters, the project must be allocated the maximum amount in this range.

Definition 5 (Range-unanimous). *A PB rule R is said to be range-unanimous if for any instance I, whenever $\sum_{j \in P} \overline{\tau_j} \leq b$, the set $\{P_j^t : j \in P, c_j^t = \overline{\tau_j}\}$ is selected under R.*

Note that the above holds by default if $\overline{\tau_j}$ is not defined for some j. Range-unanimity and range-abidingness do not

imply each other, though they seem closely related. For example, take the rule that picks for each project j, degree with minimum cost in τ_j (whenever this set costs lesser than b). This is range-abiding but not range-unanimous. In an instance where the maximum costs in τ_j together cost more than the budget, range-unanimity is satisfied by default by any rule. However, a rule that selects a degree for project j whose cost is greater than maximum cost in τ_j, does not satisfy range-abidingness. Thus, range-unanimity does not imply range-abidingness.

Proposition 12. *The rules $R_{|S|}$ and $R_{\|\ \|}$ are range-unanimous, whereas $R_{c(S)}$ and $R_{\widehat{c(S)}}$ are not.*

The following axiom essentially implies that if two valid sets differ only on the degree of one project, then the set with higher degree needs to be preferred.

Definition 6 (Degree-efficient). *A PB rule R is said to be* degree-efficient *if for any instance I, any project $j \in P$, any set S selected under R, and the degree $x \in S(j)$, it holds that*
$$k > x \implies c(S) - c^S(j) + c_j^k > b.$$

Proposition 13. *The $R_{\widehat{c(S)}}$ is degree-efficient, whereas $R_{|S|}$, $R_{c(S)}$, and $R_{\|\ \|}$ are not.*

Proof. Let R be $R_{|S|}$, $R_{c(S)}$, or $R_{\|\ \|}$. Consider the example with single project j such that for every i: (i) $c_j^{t_j} < b$ and $h_i(j) < c_j^{t_j}$ (ii) there exists $t < t_j$ such that $l_i(j) \le c_j^t \le h_i(j)$. The set $\{P_j^t : c_j^t = \overline{\tau_j}\}$ is selected under R since we know that $\tau_j \ne \emptyset$ and $\overline{\tau_j} < b$. Assume that the axiom is satisfied. Then, by the definiton of t_j, $c_j^{t_j} > b$ contradicting (i). The proof for $R_{\widehat{c(S)}}$ is deferred to appendix [Sreedurga, 2023]. $\square$

The next two axioms insist that the valid set closer to the bounds reported by all the voters must be preferred over a valid set farther from them.

Definition 7 (Lower bound-sensitive). *A PB rule R is said to be* lower bound-sensitive *if for any instance I, any project $j \in P$, and any two valid set S, S' such that for every voter i we have $c^S(j) < c^{S'}(j) < l_i(j)$, S is not selected under R.*

Proposition 14. *The $R_{\|\ \|}$ is lower bound-sensitive, whereas $R_{|S|}$, $R_{c(S)}$, and $R_{\widehat{c(S)}}$ are not.*

Proof. The proof for $R_{\|\ \|}$ is deferred to appendix [Sreedurga, 2023]. Let R be $R_{|S|}$, $R_{c(S)}$, or $R_{\widehat{c(S)}}$. For lower-bound sensitivity, consider a counter example as follows: (i) there are two projects with $D(P_1) = \{P_1^0, P_1^1, P_1^2, P_1^3\}$ and $D(P_2) = \{P_2^0, P_2^1\}$ (ii) $c_1^1 = 1$, $c_1^2 = 2$, $c_1^3 = b - 3$, and $c_2^1 = b - 2$ (iii) for every voter $i \in N$, $l_i(1) = h_i(1) = c_1^3$ and $l_i(2) = h_i(2) = c_2^1$. Clearly, $S = \{P_1^1, P_2^1\}$ is a set that is selected under R. Set $S' = \{P_1^2, P_2^1\}$ and $j = 1$. $\square$

Definition 8 (Upper bound-sensitive). *A PB rule R is said to be* upper bound-sensitive *if for any instance I, any project $j \in P$, and any two valid set S, S' such that for every voter i we have $c^S(j) > c^{S'}(j) > h_i(j)$, S is not selected under R.*

Proposition 15. *The rules $R_{|S|}, R_{c(S)}$, and $R_{\|\ \|}$ are upper bound-sensitive, whereas $R_{\widehat{c(S)}}$ is not.*

Next, we extend discount montonicity [Talmon and Faliszewski, 2019] to ensure that a winning degree of project shouldn't be omitted if it becomes less expensive.

Definition 9 (Discount-proof). *A PB rule R is said to be* discount-proof *if for any instance I, any project $j \in P$, and a set S that is selected under R, S continues to be selected if $c^S(j)$ is decreased by 1.*

Proposition 16. *The $R_{|S|}$ is discount-proof, whereas $R_{c(S)}$, $R_{\widehat{c(S)}}$, and $R_{\|\ \|}$ are not.*

5 Summary

Many times, there are multiple ways of executing a public project and hence several, but limited number of, choices for the amount to be allocated to this project. Unfortunately, the existing preference elicitation methods and aggregation rules for participatory budgeting do not take this factor into account. Our work is an attempt to bridge this gap. We generalized two utility notions defined for PB with approval votes to our model. We also proposed two other utility notions unique to our model. We analyzed all the corresponding utilitarian PB rules computationally and axiomatically.

Our computational part strengthens all the existing positive results, and also introduces several new parameterized tractability results (FPT, parameterized FPTAS) taking into account the parameters recently introduced in the PB literature. It is worth highlighting that all our computational results in Section 3 can be generalized by replacing the utilities with cardinal utility for every degree of each project. However, we present all the results for ranged approval votes due to their practical relevance, simplicity, and deep association with axiomatic analysis.

Followed by this, we introduce several axioms for our model with ranged approval votes and investigate which of these are satisfied by our PB rules. Note that, though none of the proposed PB rules satisfies all the axioms, every rule satisfies some axioms. Axiomatic analysis reflects the properties of each rule, using which the PB organizer can pick a rule based on the context. Also, it is worth bearing in mind that the novel disutility notion and PB rule $R_{\|\ \|}$ we proposed for our model satisfies as many axioms as any simple approval-based PB rule satisfies. One of the key takeaways of this paper is hence a conclusion that $R_{\|\ \|}$ is a very good choice when each voter approves a range of costs.

Acknowledgements

References

[Airiau *et al.*, 2019] Stéphane Airiau, Haris Aziz, Ioannis Caragiannis, Justin Kruger, Jérôme Lang, and Dominik Peters. Portioning using ordinal preferences: Fairness and efficiency. In *IJCAI*, 2019.

[Aziz and Lee, 2019] Haris Aziz and Barton E Lee. Proportionally representative participatory budgeting with ordinal preferences. *arXiv preprint arXiv:1911.00864*, 2019.

[Aziz and Lee, 2021] Haris Aziz and Barton E Lee. Proportionally representative participatory budgeting with ordinal preferences. In *AAAI*, volume 35, pages 5110–5118, 2021.

[Aziz and Shah, 2021] Haris Aziz and Nisarg Shah. Participatory budgeting: Models and approaches. In *Pathways Between Social Science and Computational Social Science*, pages 215–236. Springer, 2021.

[Aziz et al., 2018] Haris Aziz, Barton E Lee, and Nimrod Talmon. Proportionally representative participatory budgeting: Axioms and algorithms. In *AAMAS*, pages 23–31, 2018.

[Baumeister et al., 2020] Dorothea Baumeister, Linus Boes, and Tessa Seeger. Irresolute approval-based budgeting. In *AAMAS*, pages 1774–1776, 2020.

[Benade et al., 2021] Gerdus Benade, Swaprava Nath, Ariel D Procaccia, and Nisarg Shah. Preference elicitation for participatory budgeting. *Management Science*, 67(5):2813–2827, 2021.

[Cabannes, 2004] Yves Cabannes. Participatory budgeting: a significant contribution to participatory democracy. *Environment and urbanization*, 16(1):27–46, 2004.

[Fairstein et al., 2022] Roy Fairstein, Meir Reshef, Dan Vilenchik, and Kobi Gal. Welfare vs. representation in participatory budgeting. *arXiv preprint arXiv:2201.07546*, 2022.

[Feldmann et al., 2020] Andreas Emil Feldmann, Euiwoong Lee, and Pasin Manurangsi. A survey on approximation in parameterized complexity: Hardness and algorithms. *Algorithms*, 13(6):146, 2020.

[Goel et al., 2019] Ashish Goel, Anilesh K Krishnaswamy, Sukolsak Sakshuwong, and Tanja Aitamurto. Knapsack voting for participatory budgeting. *ACM TEAC*, 7(2):1–27, 2019.

[Ibarra and Kim, 1975] Oscar H Ibarra and Chul E Kim. Fast approximation algorithms for the knapsack and sum of subset problems. *Journal of the ACM (JACM)*, 22(4):463–468, 1975.

[Jain et al., 2020] Pallavi Jain, Krzysztof Sornat, and Nimrod Talmon. Participatory budgeting with project interactions. In *IJCAI*, pages 386–392, 2020.

[Peters et al., 2020] Dominik Peters, Grzegorz Pierczyński, and Piotr Skowron. Proportional participatory budgeting with cardinal utilities. *arXiv preprint arXiv:2008.13276*, 2020.

[Rey et al., 2020] Simon Rey, Ulle Endriss, and Ronald de Haan. Designing participatory budgeting mechanisms grounded in judgment aggregation. In *KR*, volume 17, pages 692–702, 2020.

[Röcke, 2014] Anja Röcke. *Framing citizen participation: participatory budgeting in France, Germany and the United Kingdom*. Springer, 2014.

[Shah, 2007] Anwar Shah. *Participatory budgeting*. World Bank Publications, 2007.

[Shapiro and Talmon, 2017] Ehud Shapiro and Nimrod Talmon. A participatory democratic budgeting algorithm. *arXiv preprint arXiv:1709.05839*, 2017.

[Sintomer et al., 2008] Yves Sintomer, Carsten Herzberg, and Anja Röcke. Participatory budgeting in europe: Potentials and challenges. *International journal of urban and regional research*, 32(1):164–178, 2008.

[Sreedurga and Narahari, 2022] Gogulapati Sreedurga and Yadati Narahari. Indivisible participatory budgeting under weak rankings. *arXiv preprint arXiv:2207.07981*, 2022.

[Sreedurga et al., 2022a] Gogulapati Sreedurga, Mayank Ratan Bhardwaj, and Y Narahari. Maxmin participatory budgeting. *arXiv preprint arXiv:2204.13923*, 2022.

[Sreedurga et al., 2022b] Gogulapati Sreedurga, Soumyarup Sadhukhan, Souvik Roy, and Yadati Narahari. Characterization of group-fair social choice rules under single-peaked preferences. *arXiv preprint arXiv:2207.07984*, 2022.

[Sreedurga, 2023] Gogulapati Sreedurga. Participatory budgeting with multiple degrees of projects and ranged approval votes. *arXiv preprint arXiv:2305.10972*, 2023.

[Talmon and Faliszewski, 2019] Nimrod Talmon and Piotr Faliszewski. A framework for approval-based budgeting methods. In *AAAI*, volume 33, pages 2181–2188, 2019.

[Wampler, 2010] Brian Wampler. *Participatory budgeting in Brazil: Contestation, cooperation, and accountability*. Penn State Press, 2010.