Evaluation of Project Performance in Participatory Budgeting

Niclas Boehmer[1], **Piotr Faliszewski**[2], **Łukasz Janeczko**[2], **Dominik Peters**[3], **Grzegorz Pierczyński**[2,4], **Šimon Schierreich**[5], **Piotr Skowron**[4] and **Stanisław Szufa**[2,3]
[1]Harvard University
[2]AGH University, Kraków
[3]CNRS, LAMSADE, Université Paris Dauphine - PSL
[4]University of Warsaw
[5]Czech Technical University in Prague

Abstract

We study ways of evaluating the performance of losing projects in participatory budgeting (PB) elections by seeking actions that would make them win. We focus on lowering their costs, obtaining additional approvals, and removing approvals for competing projects: The larger a change is needed, the less successful is the given project. We seek efficient algorithms for computing our measures and we analyze them experimentally, focusing on GREEDYAV, PHRAGMÉN, and EQUAL-SHARES PB rules.

1 Introduction

One of the more spectacular successes of computational social choice was the recent adoption of the EQUAL-SHARES proportional rule [Peters and Skowron, 2020; Peters *et al.*, 2021] in several real-life participatory budgeting (PB) elections. In such elections, residents vote for projects to implement in their city; there is a fixed budget, each project has a cost, and, typically, each voter indicates which projects he or she approves. So it was a bit worrisome that in the first-ever application of EQUAL-SHARES, in Wieliczka, Poland (in the "Green Million" program), the rule acted in a seemingly unreasonable way: It chose a more expensive project that got fewer votes (Project 61, costing 51 732 PLN and receiving 229 votes) and rejected a more popular, less expensive one (Project 16, costing 50 000 PLN and receiving 376 votes).[1] However, no public outrage about this has been reported, and other cities have since adopted EQUAL-SHARES. Why?

Upon closer inspection, it turns out that choosing Project 61 over Project 16 in Wieliczka was, in fact, justified. To see why, consider a toy example with 10 voters and budget $B = 10$ PLN, where the project costs and the votes are as follows:

		voters									
proj.	cost	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
a	6	✓	✓	✓	✓	✓	✓	-	-	-	-
b	3	✓	✓	✓	✓	✓	-	-	-	-	-
c	4	-	-	-	-	-	-	✓	✓	✓	✓

In particular, project a costs 6 PLN and is approved by 6 voters, $v_1, \ldots, v_6$. One can check that EQUAL-SHARES selects projects a and c, which is arguably fair: It spends the whole budget and satisfies two disjoint groups of voters, the first 6 ones and the last 4, proportionally to their sizes. Note that project b is cheaper and more popular than project c, but it is popular among the voters who also like project a, whereas c is supported by the voters who would have been underrepresented if c were left out. The seemingly odd behavior of EQUAL-SHARES in Wieliczka was similar in nature and is a consequence of EQUAL-SHARES seeking proportional representation of the voters. Nonetheless, without appropriate explanations, it would have been quite easy to criticize EQUAL-SHARES over its behavior and, indeed, the organizers of the "Green Million" did provide explicit explanations as to why several projects, including Project 16, were not funded. Still, given the intricacy of the inner workings of EQUAL-SHARES and the complexity of real-world instances, presenting simple and convincing explanations for why a project was not funded is challenging.

Motivated by this real-world example, combined with the general importance of transparency in civic participation schemes, the main goal of this paper is to provide performance measures for losing projects. The guiding idea is that each of our measures explains what *should have happened* in some aspect of a given PB election for a particular project to win (if at all possible). By presenting these measures, organizers of PB elections can explain their results in a principled way, by quantifying in what way and by how much a project has fallen short of being funded. For instance, for another project from the "Green Million" election (Project 67), we identify that it would need around 15% more approvals to get funded, while its cost would have to be decreased by 70%. This indicates that the project's loss was mainly due to missing support and not due to excessive cost, as even a considerably cheaper version of the project approved by the same supporters would not have been funded. Thereby, our measures allow for making participatory budgeting elections more transparent and can even offer advice to project proposers for future campaigns. We consider three main reasons why a project might have lost:

1. A project might have lost because it was too expensive. Hence, we ask how much cheaper should it have been to get funded, assuming everything else stays fixed.
2. A project might have lost because it was not approved by sufficiently many voters. Hence, we ask how many extra

[1]See https://equalshares.net/elections/zielony-milion.

votes would have ensured its funding. However, this idea requires some care, as it is important not only *how many* approvals we add, but also *where* we add them. Thus, we provide several measures based on the idea of adding approvals and on different attitudes toward risk.

3. A project might have lost because of its competition with other projects. Hence, we ask how many voters that support the considered project should have refrained from approving any other ones to ensure its victory.

Naturally, these measures are interrelated, but each of them focuses on a different facet of the election, and only together they give a comprehensive answer as to why a project lost.

Technical Contribution. We are interested in two aspects of our measures. First, as real-life PB instances can include hundreds of projects and tens of thousands of voters, we seek fast algorithms for computing them in practice. Since some of our measures are NP-hard to compute, we find FPT algorithms for them that work well in realistic settings.

Second, we compute our measures for projects from real-life PB instances from Pabulib [Faliszewski *et al.*, 2023] and analyze their behavior. As a side result, we obtain some striking examples of nonmonotonicity of proportionality-oriented PB rules. We also demonstrate the usefulness of our measures by giving a detailed analysis of the Wieliczka election.

We focus on the classic GREEDYAV rule and on two proportional rules, EQUAL-SHARES and PHRAGMÉN. All missing proofs and additional experimental results are available in the full paper [Boehmer *et al.*, 2023b]. The code for our experiments is available at github.com/Project-PRAGMA/Project-Performance-IJCAI-2024.

2 Preliminaries

A *participatory budgeting* (PB) instance $E = (P, V, B)$ consists of a set of projects $P = \{p_1, \dots, p_m\}$, a set of voters $V = \{v_1, \dots, v_n\}$, and budget $B \in \mathbb{N}$. Each voter v has a set $A(v) \subseteq P$ of projects that he or she approves (referred to as his or her *approval set*), and each project $p \in P$ has a price for implementing it, denoted $\mathrm{cost}(p)$. We extend this notation so that for a project p, $A(p)$ is the set of voters who approve it, and we refer to $|A(p)|$ as the approval score of project p. A set $S \subseteq P$ of projects is feasible if its cost, denoted as $\mathrm{cost}(S) = \sum_{p \in S} \mathrm{cost}(p)$, is at most B. A PB rule is a function that given a PB instance outputs a feasible set S of selected projects (i.e., our rules are resolute and, so, give unique outcomes). We refer to projects in S as the *selected* (or, *funded*) ones, and to the remaining ones as *losing*.

We consider three PB rules, GREEDYAV, PHRAGMÉN [Brill *et al.*, 2017; Los *et al.*, 2022], and EQUAL-SHARES [Peters and Skowron, 2020; Peters *et al.*, 2021]. Each of them starts with an empty set of projects W, performs a sequence of rounds, extending W with a single project in each, and eventually outputs W as the final outcome. Upon an internal tie (i.e., if two or more projects fulfill a given condition), they break it using a given, prespecified order over the projects. Given a PB instance $E = (P, V, B)$, our rules work as follows:

GREEDYAV (AV). In each round, GREEDYAV considers a project p with the highest approval score that it has not considered yet. If $\mathrm{cost}(W) + \mathrm{cost}(p) \leq B$ (i.e., if it can afford to fund p) then it includes p in W. The rule terminates upon considering all the projects.

PHRAGMÉN (PH). The voters start with empty virtual bank accounts, but receive funds in a continuous manner, one unit of funds per one unit of time. As soon as there is a project $p \in P \setminus W$ such that the voters in $A(p)$ have $\mathrm{cost}(p)$ funds in total and $\mathrm{cost}(W) + \mathrm{cost}(p) \leq B$, the rule includes p in W and resets the bank accounts of the voters from $A(p)$ to zero (these voters *buy* the project). The process stops when for every project p with at least one approval it holds that $\mathrm{cost}(W) + \mathrm{cost}(p) > B$.

EQUAL-SHARES (EQ). This rule also uses voters' virtual bank accounts, but it initiates them to $B/|V|$ per voter and does not provide further funds. Each round proceeds as follows, where b_i is the current account balance of voter $v_i \in V$. The idea is to select a project that its supporters can afford, with each supporter covering as small a fraction of its cost as possible. Formally, a project $p \in P \setminus W$ is affordable if there is $q(p) \in [0, 1]$ such that:

$$\textstyle\sum_{v_i \in A(p)} \min\big(b_i, q(p) \cdot \mathrm{cost}(p)\big) = \mathrm{cost}(p).$$

For each voter $v_i \in A(p)$, we let $q_i(p)$ be the fraction of $\mathrm{cost}(p)$ that v_i needs to cover; it is $q(p)$ if $b_i \geq q(p) \cdot \mathrm{cost}(p)$ (i.e., if v_i can afford its full share) and it is $b_i/\mathrm{cost}(p)$ otherwise. The rule selects an affordable project p with the smallest value of $q(p)$, includes it in W, and charges each $v_i \in A(p)$ with $q_i(p) \cdot \mathrm{cost}(p)$ (if there are no affordable projects, then the rule terminates).[2]

A known issue of EQUAL-SHARES is that it is not exhaustive, i.e., upon termination there may be enough funds left for further projects. Hence, in practice one needs to apply a *completion method*. In most of our experiments we do the following: When EQUAL-SHARES terminates, we run PHRAGMÉN, but with voters' bank accounts initiated to their value at the end of EQUAL-SHARES. We call this rule EQ/PHRAGMÉN. In Wieliczka, a different completion method was used, which, unfortunately, is too computationally intensive for our full set of experiments [Boehmer *et al.*, 2023b, Appendix A].

Each of the above-described PB rules can be computed in polynomial time using a *round-based* algorithm, which executes each round following the definition. Many of our measures can be computed while running a round-based algorithm, by performing some additional steps in each round.

Definition 2.1. *Let f be a PB rule. Let $\mathrm{measure}_E(p)$ be a function that takes as input a PB instance E and project p, and let $t\colon \mathbb{N} \to \mathbb{N}$ be some function. We say that $\mathrm{measure}_E(p)$ can be computed* alongside $f(E)$ at a cost of $t(|E|)$ per round *if it is possible to compute its value using a standard round-based algorithm for $f(E)$, extended to perform at most $t(|E|)$ additional computational steps in each round.*

[2] In the language of Brill *et al.* [2022], this definition is based on *cost utilities*. We can also understand the rule as selecting in each round the project p supported by a group of voters with the largest weighted size, defined as $\sum_{v_i \in A(p)} q_i(p)/q(p)$ (intuitively, voters that would contribute their full share count as 1 in this sum, but those who would contribute less count as respective fractions).

measure	AV	PH	EQ
cost-red	along/$O(1)$	along/$O(1)$	along/$O(1)$
optimist-add	along/$O(1)$	along/$O(n \log n)$	along/$O(n \log n)$
50%-add	along/$O(1)$	sampling	sampling
pessimist-add	along/$O(1)$	NP-com./FPT	FPT
singleton-add	along/$O(1)$	along/$O(1)$	brute-force
rival-red	sampling	sampling	sampling

Table 1: Summary of the algorithms for computing our measures. By along/$O(1)$ and along/$O(n \log n)$ we mean that the measure can be computed alongside the rule, with $O(1)$ or $O(n \log n)$ additional cost per round (where n is the number of voters). By sampling, we mean algorithms based on simulating a given action a number of times. By FPT, we mean the algorithm from Theorem 3.6. By brute-force, we mean adding singleton voters one by one.

3 The Measures and How to Compute Them

In this section, we describe our measures and provide ways of computing them. While we also analyze their worst-case computational complexity, our focus is on obtaining practically usable algorithms that can be applied to PB elections from Pabulib [Faliszewski *et al.*, 2023]. We consider GREEDYAV (AV), PHRAGMÉN (PH), and EQUAL-SHARES (EQ) as this allows for a clean presentation; by combining algorithms for EQ and PH, one can obtain algorithms that work for EQ/PHRAGMÉN. We summarize our algorithms in Table 1.

The common feature of our measures is that they correspond to specific actions that either the voters or the project proposers could have taken in the election. In this respect, they are closely related to the margin-of-victory [Magrino *et al.*, 2011; Cary, 2011; Xia, 2012] and, more broadly, bribery notions [Faliszewski *et al.*, 2009; Faliszewski and Rothe, 2015; Yang, 2020]. In particular, we borrow ideas from Faliszewski *et al.* [2017] about bribery in multiwinner elections.

Except for rivalry reduction (see Section 3.4), our measures are well-defined in all but a few pathological cases (see Appendix B.1 in our full version [Boehmer *et al.*, 2023b]).

3.1 Cost-Reduction Measure

Our conceptually simplest measure is the one based on reducing the project's cost. The measure was also studied theoretically by Baumeister *et al.* [2021] in the context of manipulating election results. Formally, we define it as follows.

Definition 3.1. *Let f be a PB rule, let $E = (P, V, B)$ be a PB instance, and let $p \in P \setminus f(E)$ be a losing project. The cost-reduction measure of p in E, denoted* $\text{cost-red}_E^f(p)$*, is the largest value such that if we replace p's cost with it, then f selects p.*

That is, $\text{cost-red}_E(p)$ is the project's cost after the smallest possible reduction that gets p funded (we drop the superscript denoting the rule when it is clear from the context).

For AV, PH, and EQ, it is immediate that we can compute the cost-reduction measure in polynomial time using binary search, but it would require recomputing the rules multiple times. Following Baumeister *et al.* [2021] who considered this approach for AV, we compute it alongside our rules, for each round finding the largest cost at which our project can be selected right then, and outputting the globally largest one.

Proposition 3.2. *For* AV*,* PH*, and* EQ*,* $\text{cost-red}_E(p)$ *can be computed alongside the rule, at an $O(1)$ cost per round.*

3.2 Add-Approvals Measures

Next, we consider the number of additional approvals needed by a losing project to be funded. To see why it matters where we add the approvals, let us consider an example. We have projects a, b, c, d, budget $B = 10$, and ten votes as follows:

		voters									
proj.	cost	x_1	x_2	x_3	y_1	y_2	y_3	z_1	z_2	z_3	z_4
a	4	-	-	-	-	-	-	✓	✓	✓	✓
b	3	-	-	-	✓	✓	✓	-	-	-	-
c	2	-	✓	✓	-	-	-	-	-	-	-
d	2	✓	-	-	-	-	-	-	-	-	-

Under EQ, first voters z_1, z_2, z_3, z_4 buy project a, then voters y_1, y_2, y_3 buy project b and, finally, voters x_2, x_3 buy project c. If d got additional approvals from x_2 and x_3, then EQ would select it instead of c. However, if d got the extra approvals from z_1 and z_2 instead, then it would still lose; these voters spend all their funds on a before d is even considered. (For PH there also are cases where it matters who adds the extra approvals, whereas for AV only their number matters).

Our solution is to consider different attitudes toward risk when asking random voters to add approvals: An optimist hopes to get new approvals from exactly the right voters, a pessimist prefers certainty of being funded, and a middle-ground position is to ask for some fixed probability of success.

Definition 3.3. *Let f be a PB rule, let $E = (P, V, B)$ be a PB instance, and let $p \in P \setminus f(E)$ be a losing project. Then:*

1. $\text{optimist-add}_E^f(p)$ *is the smallest number ℓ such that it is possible to ensure that p is funded by choosing ℓ voters and extending their approval sets with p.*
2. $\text{pessimist-add}_E^f(p)$ *is the smallest number ℓ such that for each subset of ℓ voters who do not approve p, extending their approval sets with p ensures that p is funded.*
3. $50\%\text{-add}_E^f(p)$ *is the smallest number ℓ such that if ℓ voters selected uniformly at random (among those who originally do not approve p) extend their approval sets with p, then p is funded with probability at least* 50%.

The optimist measure was previously considered by Faliszewski *et al.* [2017] in multiwinner voting, whereas the 50%-threshold one is inspired by an analogous notion used by Boehmer *et al.* [2021; 2022] in the single-winner setting, and by Boehmer *et al.* [2023a] in the robustness analysis of PB outcomes. For each rule f, each PB instance E, and each losing project p, we naturally have the following:

$$\text{optimist-add}_E(p) \leq 50\%\text{-add}_E(p) \leq \text{pessimist-add}_E(p).$$

For GREEDYAV all three measures are equal and immediate to compute, so we mostly focus on the proportional rules. Computing the optimist measure is easy, as it suffices to consider each round independently and find the "richest" voters whose approval for p would lead to funding it.

Proposition 3.4. *For* PH *and* EQ*,* $\text{optimist-add}_E(p)$ *can be computed alongside the rule, at an $O(n \log n)$ cost per round, where n is the number of voters. For* AV *it can be computed alongside the rule at an $O(1)$ cost per round.*

On the negative side, deciding if the pessimist measure has at least a given value under PHRAGMÉN is coNP-complete (the result for EQUAL-SHARES remains elusive, but we suspect intractability). Our proof looks at the complement of the problem, where we ask if it is possible to add a certain number of approvals for p without getting it funded, and we give a reduction from a variant of SET COVER where we ask for an exact cover (i.e., a family of pairwise disjoint sets that union up to a given universe). The idea is to set up a PB instance where PHRAGMÉN interleaves between selecting candidates from the to-be-covered universe and dummy candidates, whose selection resets voters' budgets to a fixed state. If we add approvals for a designated candidate p to voters that form an exact cover, then this process goes as planned, but if we add approvals to two voters whose corresponding sets overlap, then p gets funded. This is our most involved theoretical result.

Theorem 3.5. *For* PH*, the problem of deciding if* $\text{pessimist-add}_E(p)$ *is at least a given value* ℓ *is* coNP*-complete, even if all projects have unit cost.*

Fortunately, for PHRAGMÉN and EQUAL-SHARES we can compute the pessimist measure using an FPT algorithm parameterized by the number of originally funded projects. Unlike many FPT algorithms, this one is indeed practical and we use it in our experiments (using Gurobi).

Theorem 3.6. *For* PH *and* EQ*, there is an algorithm that computes* $\text{pessimist-add}_E(p)$ *and runs in* FPT *time with respect to parameter* $|f(E)|$*, i.e., the number of rounds.*

*Proof sketch (*PHRAGMÉN*).* Let $E = (P, V, B)$ be a PB instance with losing project p (for ease of exposition, we assume p to be last in the tie-breaking order). We will show how to compute the largest number of approvals whose addition *does not* lead to funding p; $\text{pessimist-add}_E(p)$ is one larger. Let $k = |\text{PHRAGMÉN}(E)|$ be the number of rounds performed by the rule. We assume that p is approved by at least one voter.

For each round i, let m_i be the difference between $\text{cost}(p)$ and the total funds that approvers of p have in round i. Our goal is to find a largest group of voters who do not approve p and whose total funds in each round i are at most m_i.

For each voter $v_i \in V \setminus A(p)$, we define its *balance vector* $(b_1^i, \dots, b_k^i)$, which contains the balance of the voter's bank account right before each round. We partition the voters not approving p into *voter-types* $T = \{T_1, \dots, T_t\}$, where each type consists of the voters with identical balance vectors, and write $(b_1^{T_i}, \dots, b_k^{T_i})$ for the balance vector of voters from group T_i.

To solve the problem, we form an integer linear program (ILP). For every voter-type T_i we have a nonnegative integer variable x_{T_i} that represents the number of voters of this type that will additionally approve p. For each round i we form the following *round-constraint*: $\sum_{j \in \{1,\dots,t\}} x_{T_j} \cdot b_i^{T_j} \leq m_i$. The objective function is to maximize the sum of all the x_{T_i} variables. Our algorithm outputs the value of this sum plus 1.

There are at most $O(2^k)$ voter types. Indeed, each voter type corresponds to a k-dimensional 0/1 vector which has 1 in position i if a voter approves—and, hence, pays for—the candidate selected in round i. Thus, the number of variables in our ILP is $O(2^k)$; we solve it using the classic algorithm of Lenstra, Jr. [1983] in FPT time with respect to k. □

For the 50%-threshold measure, we resort to sampling. That is, given rule f, a PB instance E and a losing project p, we iterate over numbers ℓ of approvals to add and for each of them we repeat the following experiment t times (where t is a parameter): We add approvals for p to ℓ voters chosen uniformly at random (among those not approving p) and we compute f on the thus-modified instance. We terminate for the smallest value ℓ where p was funded at least $t/2$ times. We use sampling because the construction from Theorem 3.5 also shows that for PHRAGMÉN, the problem of evaluating the probability that a given project is funded after randomly adding a given number of approvals is #P-complete.

3.3 Add-Singletons Measure

Instead of asking existing voters to approve some project p, one can also recruit additional voters, who would approve p only. This gives rise to the following measure.

Definition 3.7. *Let* f *be a PB rule, let* $E = (P, V, B)$ *be a PB instance, and let* $p \in P \setminus f(E)$ *be a losing project. Then* $\text{singleton-add}_E^f(p)$ *is the smallest number* ℓ *such that if we extend* V *with* ℓ *voters who only approve* p*, then* f *selects* p*.*

For AV, this is equal to the measures from the previous section. For PHRAGMÉN, $\text{singleton-add}_E(p)$ is upper-bounded by $\text{optimist-add}_E(p)$: In each round the newly added voters always have at least as much money as the original ones (until p is selected). Under EQUAL-SHARES adding new voters changes the initial balances of the voters, which can change the overall execution of the rule and, hence, there is no clear relation between the two measures. In fact, under EQUAL-SHARES it is even possible that a project is funded after adding ℓ voters, but may fail to be funded after adding $\ell + 1$ of them [Lackner and Skowron, 2023, Proposition A.3]; see also the experiments in Section 4.1.

Computing the value of $\text{singleton-add}_E(p)$ is easy for PHRAGMÉN because we can compute how much additional funds each new voter would bring in each round. For EQUAL-SHARES, we keep adding voters one by one and recompute the rule each time (in our experiments, we were forced to add larger groups to speed up computation).

Proposition 3.8. *For* AV *and* PH*,* $\text{singleton-add}_E(p)$ *can be computed alongside the rule, at an* $O(1)$ *cost per round.*

3.4 Rivalry-Reduction Measure

Under proportional rules, a project may lose because its supporters also approve other projects, on which they spend their virtual money. Thus, another strategy that a project proposer could employ to increase a project's chances of success is to try to convince its supporters to not approve other projects.

Definition 3.9. *Let* f *be a PB rule, let* $E = (P, V, B)$ *be a PB instance, and let* $p \in P \setminus f(E)$ *be a losing project. Then* $\text{rival-red}_E^f(p)$ *is the smallest number* ℓ *such that if we select* ℓ *voters uniformly at random (among those who approve* p*) and change them to only approve* p*, then* p *is funded with probability at least* 50%*.*

This measure is not always defined: A project with too few voters will not win even if they do not support any other projects. For the sake of focus, we do not study the optimist

	optimist	pessimist	50%	singleton	rival	cost
optimist	–	0.87	0.98	0.98	0.88	0.76
pessimist	0.87	–	0.94	0.84	0.50	0.73
50%	0.98	0.94	–	0.95	0.78	0.79
singleton	0.98	0.84	0.95	–	0.93	0.74
rival	0.88	0.50	0.78	0.93	–	0.63
cost	0.76	0.73	0.79	0.74	0.63	–

Table 2: Pearson Correlation Coefficients between measures for EQ/PHRAGMÉN (values near 1 mean strong correlation).

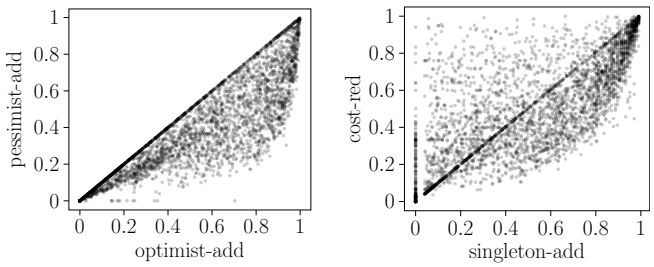


Figure 1: Correlation plots where each point is a project. The measures are normalized between 0 (maximal change) and 1 (no change).

and pessimist variants, except to note that even computing the optimist variant would be NP-complete (by basically adapting proofs on bribery and control in single-winner approval voting [Faliszewski *et al.*, 2009; Hemaspaandra *et al.*, 2007]):

Proposition 3.10. *For* AV, PH, *and* EQ, *the problem of deciding, given a PB instance E, a losing project p, and an integer ℓ, if it is possible to ensure p's victory by changing at most ℓ votes that originally approve p to only approve p is* NP*-complete, even when $B = 1$ and projects have unit costs.*

Note that this intractability result already holds in very restricted settings, thereby ruling out algorithmic results analogous to the ones presented in Section 3.2. To compute rival-red, we use an analogous sampling approach as in the case of 50%-add. One may also wonder why we consider *all* the voters who approve p and not only those who approve p *and* some further project(s). Our approach captures a campaign that reaches the supporters of the project randomly.

4 The Measures in Practice

Next, we analyze the behavior of our measures on real-world PB instances from Pabulib [Faliszewski *et al.*, 2023], focusing on EQ/PHRAGMÉN. Afterwards, we present a detailed study of the PB election from Wieliczka. See the full paper for more details and the results for PH and AV [Boehmer *et al.*, 2023b].

Data. We conduct our experiments on all 551 PB instances with approval votes from Pabulib [Faliszewski *et al.*, 2023] for which both PHRAGMÉN and EQ/PHRAGMÉN terminate within one second (on 1 thread of an Intel(R) Xeon(R) Gold 6338 CPU @ 2.00GHz core) that were available when writing the paper. In total, there are 3639 losing projects for EQ/PHRAGMÉN and 3513 for PHRAGMÉN.

Measures. To compute the measures, we use the algorithms described in Section 3. For the adding-voters and sampling algorithms, we increase the approval score of the designated project by 1% in each step (repeating each step 100 times for the sampling algorithms). To simplify comparisons between measures, we normalize them to lie between 0 (being far away from victory) and 1 (being close to victory). Specifically, for our four measures modifying a project's approval score, we divide its original approval score by its approval score plus the measure. For example, if a project with score 20 requires 80 additional approvals (according to the measure), then the normalized value is 0.2, since the project received 20% of the needed approvals. For cost-red, we divide cost-red(p) by cost(p), and for rival-red, we report the fraction of supporters who can continue to approve other projects.

Running Times. We ran our experiments on 10 threads of an Intel(R) Xeon(R) Gold 6338 CPU @ 2.00GHz core. The algorithms for optimist-add, pessimist-add, singleton-add, and cost-red are all very fast, finishing in below 20 seconds on 95% of instances. The sampling-based algorithms for rival-red and 50%-add are naturally slower, but still finish in 88% of cases in below 10 minutes. Altogether, while our implementations are certainly not fully optimized, running the different algorithms for a PB exercise in one's city is feasible, even for tens of thousands of voters and hundreds of projects.

4.1 Behavior and Correlation

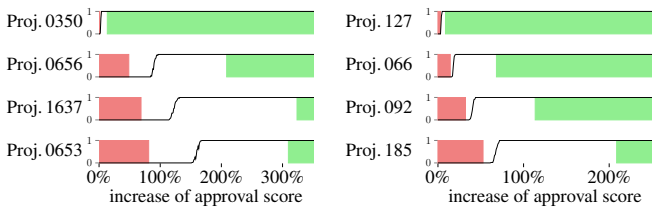
In Table 2, we show correlation between our measures for EQ/PHRAGMÉN, as given by the Pearson Correlation Coefficient (PCC); in Figure 1 we show two examples of correlation plots. Notably, many projects appear to be close to funding.

Add-Approvals and Singletons Measure. We first analyze the four measures related to adding approvals to existing or new ballots. As seen in Table 2, they all have a strong pairwise correlation. However, there are small differences motivating a partitioning of the measures into two groups: optimist-add, 50%-add, and singleton-add all have a pairwise correlation of at least 0.95, whereas pessimist-add has a lower correlation with the other three measures. In the first group, the correlation between optimist-add and singleton-add is particularly strong. For more than 90% of the projects, the difference between the two measures is less than 0.051 (exceptions include projects where one of the two measures is zero).

Comparing the optimistic and pessimistic views on adding approvals, while they have a strong correlation of 0.87, on the level of single projects, they can produce quite different results (see Figure 1 left). In fact, for around 10% of projects, almost twice as many approvals are needed to get the project funded under the pessimistic approach than under the optimistic one. Thus, under EQ/PHRAGMÉN, it really matters which voters add approvals for a project.

The 50%-add measure lies between optimist-add and pessimist-add and is strongly correlated with them. It tends to be slightly closer to the optimistic view (average difference 0.063) than to the pessimistic one (0.077).

In Figure 2, we show for two instances how the funding probabilities of projects evolve when adding approvals uniformly at random to existing voters. Projects very quickly transition from having a funding probability close to 0% to having one close to 100%, even for projects where there is a large



(a) Wierzbno Wygledow 2019 (b) Lodz 2022

Figure 2: Line plots showing how the funding probability of a project develops from 0 to 1 when increasing its approval score by adding approvals uniformly at random to existing voters. The red area goes until the optimist-add value and the green area extends from the pessimist-add value.

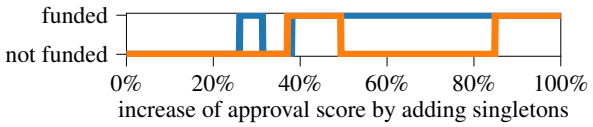
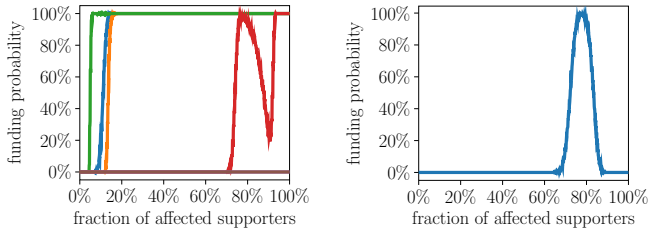


Figure 3: Behavior of two projects when adding voters who only support the project, taken from Warsaw 2023 (Praga-Polnoc, in blue) and Warsaw 2017 (Goclaw, in orange).



(a) Boernerowo 2018 (b) Kamionek 2019

Figure 4: Line plots showing how the funding probability of a project develops if we remove rivalry approvals from its supporters selected uniformly at random (each line corresponds to a single project; all non-funded projects are shown).

gap between optimist-add and pessimist-add. This phase-transition-like behavior appears for almost all projects. Thus, for practical purposes, reasonably optimistic and pessimistic views coincide (i.e., requiring that a project gets funded with at least 2% probability is very similar to requiring that it gets funded with at least 98% probability). Thus, 50%-add gives us a good estimate of how many approvals a project "practically misses" to get funded.

If one wants to reduce the number of measures, it is probably simplest to use singleton-add, as it is strongly correlated with both optimist-add and 50%-add, can be explained very easily, and its brute-force algorithm works for all rules. The only drawback of this measure for EQUAL-SHARES is that, as discussed in Section 3.3, adding singletons can make a project lose. In fact, this occasionally happens on our data in unexpected ways (see Figure 3 for two examples of strongly nonmonotonic behavior).

Rivalry-Reduction Measure. Our sampling-based algorithm for rival-red returns a value for 1605 out of the 3639 not-funded projects, which in particular implies that we can get these projects funded after removing all other approvals from the ballots of some (or all) of their supporters. This highlights the general power of lobbying one's supporters to only approve a single project in an election using a proportional rule. Regarding the correlation of rival-red with the other measures, Table 2 shows the PCC values on the 1605 projects where rival-red is defined. The correlation with the adding approval measures is strong (but not very strong). One interesting observation is that for projects where optimist-add is below 0.5 (i.e., for projects whose approval score needs to at least double to be able to win), removing rivalry approvals is almost never sufficient to get funded. Regarding the 2034 projects that cannot be funded after removing rivalry approvals, they have a very diverse performance with regard to the other measures. For example, removing rivalry approvals might not be sufficient even for projects that would be funded after getting only a few additional approvals.

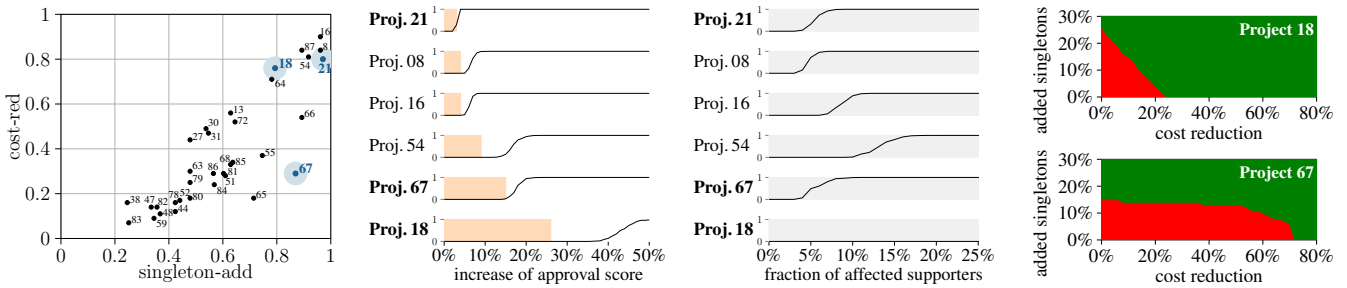
Interestingly, removing more rivalry approvals does not necessarily help a project. The reason for this is that by removing an approval of a rival, we can modify the execution of EQ/PHRAGMÉN in arbitrary rounds and thereby also help other projects to get funded. Interestingly, this nonmonotonic behavior of projects appears in many forms. Figure 4 shows two examples. That said, for rivalry reduction, for most projects there is a quick jump from an almost 0% to an almost 99% funding probability, as in the case of adding-approvals.

Cost-Reduction Measure. The cost reduction measure is less correlated to the other ones, which is expected since it is the only measure that modifies the costs. Yet, as all our measures help projects in some way, it is intuitive that there is a certain correlation (of around 0.75) to the other measures. To analyze the tradeoff between adding approvals and reducing costs in more detail, Figure 1 (right) shows the connection between cost-red and singleton-add. We see that for a majority of projects, adding approvals is more powerful than reducing the project's cost. However, there are also numerous projects where it is the other way around, so looking at both measures is more informative than considering only one of them.

4.2 Wieliczka's Green Million

We conclude by applying some of our measures to the PB election held in Wieliczka in 2023, where EQUAL-SHARES was used for the first time (https://equalshares.net/resources/zielony-milion/). This PB election focused on ecological issues, with 64 projects placed on the ballot. 6586 people voted, each project had cost up to 100 000 PLN, and each voter could approve any number of projects. The budget was one million PLN (around 225 000 EUR). Notably, Green Million used a more involved completion method of EQUAL-SHARES which increases the initial balances of the voter's bank accounts in small steps (see [Boehmer *et al.*, 2023b, Appendix A] for a description). This is computationally intensive, so we focus on the easier-to-compute measures: cost-red, singleton-add, rival-red, and 50%-add.

Figure 5 shows the results. In particular, Figure 5 (a) depicts the correlation between singleton-add and cost-red, Figure 5 (b) shows how the funding probabilities of projects change when adding approvals to existing voters at random,



(a) Correlation plots for cost-red and singleton-add. Larger font and pale background indicate the three projects that we focus on.

(b) Funding probability when adding approvals uniformly at random to voters. Orange areas go to the singleton-add value.

(c) Funding probability when removing rivalry approvals from some uniformly at random selected group of supporters.

(d) Tradeoff between fractional reduction of costs and increase of approval score by adding singletons.

Figure 5: Information on the Wieliczka 2023 Green Million election.

and Figure 5 (c) shows their behavior when removing rivalry approvals. Regarding project 16 which we featured in the introduction, our results confirm the intuition that the project was close to getting funded, as the project performs quite well according to all measures. For instance, it only needs 4% additional singleton votes or a cost reduction by 10%. However, we also see that other projects (e.g., project 21 as discussed below) are closer to getting funded according to some metrics, a phenomenon that is difficult to infer by just looking at vote counts and costs. Below, we analyze three other projects cherry-picked for interesting conclusions, arguing what project proposers could learn from our measures. The analysis reinforces the conclusion that projects close to being funded regularly appear and that our measures contribute different perspectives.

Project 21 (cost 100 000 PLN, 496 votes). According to our measures, this project was very close to winning. Indeed, it needs only about 3% additional singleton votes or additional approvals to be funded. It was also close to winning in terms of rivalry reduction, in the sense that relatively few of its supporters (below 10%) would need to refrain from supporting other projects in order for Project 21 to be funded. On the other hand, the cost-reduction measure shows that the project would have to be 20% cheaper to be selected, given the current votes. This indicates a strong project that probably should be resubmitted in the next edition of the program, with a slightly more aggressive support campaign. Reducing the cost of the project might be somewhat less effective.

Project 18 (cost 51 000 PLN, 163 votes). The project performs similarly under singleton-add and cost-red; both measures indicate that it was about 80% on the way to winning. With respect to both of these measures, only 5 other losing projects do better. However, Project 18's 50%-add performance is much worse, and the project continues to lose even if we remove rivalry approvals from all its supporters. This indicates that the main problem of Project 18 is insufficient votes: Even in the first round of EQUAL-SHARES, its supporters don't have enough money available to buy the project. Adding approvals to existing voters helps less with this funding gap, as these voters might spend their budget in earlier rounds on more popular projects.

Project 67 (cost 16 500 PLN, 140 votes). Project 67 needs around 15% more approvals according to singleton-add and 50%-add, but its cost would need to be reduced by 70% to get funded. Thus Project 67 is not so far off in terms of the number of supporters it has, but under EQUAL-SHARES, those supporters spend almost all of their money on more popular projects before Project 67 is considered by the rule. This interpretation is confirmed by its performance with respect to rival-red, where its performance matches projects like Project 21, which is very close to being funded according to all measures. Hence, Project 67's main issue is competition with other projects supported by the same voters.

Combined Strategy. One might also ask whether a combined strategy of slightly lowering the cost of a project and getting a few more singleton voters could be effective for our projects. We illustrate this approach in Figure 5 (d). The x-axis shows the percentage by which the cost is reduced, and the y-axis shows the increase of the approval score of the project (by adding singletons). A point is colored green if the project would be funded if both actions were performed, and red if it would continue to lose. We see that for Project 21 we can exchange additional votes for cost reduction in a linear way, but for Project 67 the behavior is highly nonlinear.

5 Future Work

We have begun the work of designing useful performance measures for projects in PB elections. In the future, it would be useful to collect feedback from participants about what measures and visualizations lead to an increased perceived transparency and legitimacy of the election outcome. There is also room for developing improved measures. For example, for increasing the score of project p, one could try to prioritize adding approvals to voters who are most likely to vote for p (e.g., because they are similar to current supporters of p). Our rival-red measure could also be refined by allowing for the deletion of singular (instead of all) competing approvals from supporter's ballots. For all our probabilistic measures, considering funding thresholds different from 0%, 50%, and 100% present additional natural extensions.

Ethical Statement

While the goal of our measures is to increase the transparency and perceived legitimacy of participatory budgeting outcomes, they might also be seen as contributing to encouraging "manipulative" behavior. In particular, presenting and advertising the measures in a participatory budgeting election, one might (accidentally) raise awareness for the possibility to manipulate the outcome. For instance, project proposers might be tempted to underreport the costs of their projects if they believe that this considerably improves their chances of getting funded or to bribe (specific) voters to change their votes. Similarly, voters might start to consider the possibility of misreporting their preferences. However, the immediate benefit that our measures provide for such a behavior is limited, as measures can only be computed after the election already happened and all anonymized votes have been released. Moreover, as voters are anonymous, the measures do not lead to any actionable, specific manipulative actions. As such the influence would rather be implicit and the influence on votes and project costs would be limited to the next iteration of the election (which will most likely unfold very differently, given that some popular projects have now been implemented).

Acknowledgements

This work was done while Niclas Boehmer was affiliated with TU Berlin, where he was supported by the DFG project ComSoc-MPMS (NI 369/22). This work was co-funded by the European Union under the project Robotics and advanced industrial production (reg. no. CZ.02.01.01/00/22_008/0004590). Šimon Schierreich also acknowledges the support of the Grant Agency of the Czech Technical University in Prague, grant No. SGS23/205/OHK3/3T/18. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 101002854). Piotr Skowron was supported by the European Research Council (ERC; project PRO-DEMOCRATIC, grant agreement No 101076570). Łukasz Janeczko's research presented in this paper is supported in part from the funds assigned by Polish Ministry of Science and Technology to AGH University. Additionally, Grzegorz Pierczyński was supported by Poland's National Science Center grant no. 2022/45/N/ST6/00271.

References

[Baumeister *et al.*, 2021] D. Baumeister, L. Boes, and J. Hillebrand. Complexity of manipulative interference in participatory budgeting. In *Proceedings of ADT-2021*, pages 424–439. Springer, 2021.

[Boehmer *et al.*, 2021] N. Boehmer, R. Bredereck, P. Faliszewski, and R. Niedermeier. Winner robustness via swap- and shift-bribery: Parameterized counting complexity and experiments. In *Proceedings of IJCAI-2021*, pages 52–58, 2021.

[Boehmer *et al.*, 2022] N. Boehmer, R. Bredereck, P. Faliszewski, and R. Niedermeier. A quantitative and qualitative analysis of the robustness of (real-world) election winners. In *Proceedings of EAAMO-2022*, pages 7:1–7:10, 2022.

[Boehmer *et al.*, 2023a] N. Boehmer, P. Faliszewski, L. Janeczko, and A. Kaczmarczyk. Robustness of participatory budgeting outcomes: Complexity and experiments. In *Proceedings of SAGT-2023*, 2023.

[Boehmer *et al.*, 2023b] N. Boehmer, P. Faliszewski, L. Janeczko, D. Peters, G. Pierczynski, S. Schierreich, P. Skowron, and S. Szufa. Evaluation of project performance in participatory budgeting. *arXiv:2312.14723*, 2023.

[Brill *et al.*, 2017] M. Brill, R. Freeman, S. Janson, and M. Lackner. Phragmén's voting methods and justified representation. In *Proceedings of AAAI-2017*, pages 406–413, 2017.

[Brill *et al.*, 2022] M. Brill, S. Forster, M. Lackner, J. Maly, and J. Peters. Proportionality in approval-based participatory budgeting. In *Proceedings of AAAI-2023*, pages 398–404, 2022.

[Cary, 2011] D. Cary. Estimating the margin of victory for instant-runoff voting. 2011. Presented at 2011 Electronic Voting Technology Workshop/Workshop on Trushworthy Elections.

[Faliszewski and Rothe, 2015] P. Faliszewski and J. Rothe. Control and bribery in voting. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 7. Cambridge University Press, 2015.

[Faliszewski *et al.*, 2009] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How hard is bribery in elections? *Journal of Artificial Intelligence Research*, 35:485–532, 2009.

[Faliszewski *et al.*, 2017] P. Faliszewski, P. Skowron, and N. Talmon. Bribery as a measure of candidate success: Complexity results for approval-based multiwinner rules. In *Proceedings of AAMAS-17*, pages 6–14, 2017.

[Faliszewski *et al.*, 2023] P. Faliszewski, J. Flis, D. Peters, G. Pierczyński, P. Skowron, D. Stolicki, S. Szufa, and N. Talmon. Participatory budgeting: Data, tools and analysis. In *Proceedings of IJCAI-2023*, pages 2667–2674, 8 2023.

[Hemaspaandra *et al.*, 2007] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007.

[Lackner and Skowron, 2023] M. Lackner and P. Skowron. *Multi-Winner Voting with Approval Preferences*. Springer Briefs in Intelligent Systems. Springer, 2023.

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

[Los *et al.*, 2022] M. Los, Z. Christoff, and D. Grossi. Proportional budget allocations: Towards a systematization. In *Proceedings of IJCAI-2022*, pages 398–404, 2022.

[Magrino *et al.*, 2011] T. Magrino, R. Rivest, E. Shen, and D. Wagner. Computing the margin of victory in IRV elections. Presented at 2011 Electronic Voting Technology Workshop/Workshop on Trushworthy Elections, August 2011.

[Peters and Skowron, 2020] D. Peters and P. Skowron. Proportionality and the limits of welfarism. In *Proceedings of EC-20*, pages 793–794, 2020.

[Peters *et al.*, 2021] D. Peters, G. Pierczynski, and P. Skowron. Proportional participatory budgeting with additive utilities. In *Proceedings of NeurIPS-2021*, pages 12726–12737, 2021.

[Xia, 2012] L. Xia. Computing the margin of victory for various voting rules. In *Proceedings of EC-12*, pages 982–999, June 2012.

[Yang, 2020] Y. Yang. On the complexity of destructive bribery in approval-based multi-winner voting. In *Proceedings of AAMAS-2020*, pages 1584–1592, 2020.