Essays on Participatory Budgeting

Candidate number: 1062958
word count: 12834 (Overleaf word count)

Michaelmas Term, 2022

Contents

Abstract

Participatory budgeting generally involves a group of voters collectively deciding what options to select to make up some common shared outcome. This can apply to federal elections, choosing committees for representation, deciding how to allocate funds towards public projects, deciding what movies to play at a cinema, etc. Participatory budgeting studies how we can use the results of the polls to pick which projects to fund in a fair and efficient manner. There is a large variety of situations in which these ideas can be used, as mentioned before, and there is also a large variety of ways even to poll voters.

The contribution of this thesis is fourfold. We investigate Pareto optimality in a very general setting by looking at the complexity of computing Pareto optimal solutions and verifying Pareto optimality, and briefly touch on the idea of negative utilities. We also consider selecting committees given voters' positive, negative, or neutral opinions on candidates, and discuss notions of fairness and the complexities involving them. We extend work on using ranked preferences to choose committees to the more general setting where the size of the committee is not fixed, but rather each candidate has a cost and there is a total budget, and we consider two aggregation rules and investigate how variations in the costs of items and the overall budget affects the outcome, so that project proposers can better understand what kind of margins they were working with.

1 Introduction

1.1 Motivation

Participatory budgeting generally involves a group of voters collectively deciding what options to select to make up some common shared outcome. This can apply to federal elections, choosing committees for representation, deciding how to allocate funds towards public projects, deciding what movies to play at a cinema, etc.

For example, suppose a city has a certain budget for making the city a nice place to live in, and it has a bunch of different public projects it could fund, like building parks, fixing roads, putting on festivals, etc. It can poll its citizens to get their opinions about these different projects, then aggregate these opinions to decide how to allocate these funds. Not only does this help the overseers understand the preferences of the citizens, it also allows the population to be more directly involved in the running of the city.

Aziz and Shah [6] note that several cities have directed significant funding for public projects to be allocated by participatory budgeting, so it is important how voter opinions are aggregated and used to decide funding allocation. In particular, we do not want to end up with undesirable outcomes, like only projects located in the most densely populated areas being funded, or certain significant groups of the population being underrepresented.

Participatory budgeting studies how we can use the results of the polls to pick which projects to fund in a fair and efficient manner. There is a large variety of situations in which these ideas can be used, as mentioned before, but there is also a large variety of ways even to poll voters. For example, voters could give a thumbs up or thumbs down to each idea, they could rank ideas, they could assign real valued utilities to the ideas, they could propose their favourite outcomes, the list goes on. Some of these polling methods get a more fine-grained view on voters' preferences, though they may require significantly more time and effort on the voters' part.

Once the voters' preferences have been collected through some manner of polling, there's still the question of how to aggregate the results. We can establish methods for aggregating these results, and analyze the fairness of the results, but as Peters and Skowron [12] show, there are even conflicting notions of fairness. We can therefore look at a variety of aggregation rules and notions of fairness, and compare and contrast them, seeing how they perform by different metrics.

1.2 Overview of contributions

We start off in section 4 by looking at a very general setting, and investigating Pareto optimality. In particular, we look at the complexity of computing Pareto optimal solutions and verifying Pareto optimality in the most general setting, and in some simple restrictions, and briefly touch on the idea of negative utilities. In section 5, where we consider selecting committees given voters' positive, negative, or neutral opinions on candidates, we point out several corrections in the work done to date, then propose a fairness axiom of our own, compare it with previously seen related axioms of fairness, check compatibility with the Pareto optimality, and prove hardness of verification of satisfaction of our axiom. In section 6, we extend work on using ranked preferences to choose committees to the more general setting where the size of the committee is not fixed, but rather each candidate has a cost and there is a total budget. In section 7, we consider two aggregation rules and investigate how variations in the costs of items and the overall budget affects the outcome, so that project proposers can better understand what kind of margins they were working with.

2 Preliminaries

2.1 General definitions

Definition $\mathbb{N} = \{0, 1, 2, ...\}$

Definition $[t] := \{1, 2, ..., t\}$ for $t \in \mathbb{N}$.

In a participatory budgeting scenario we have

- a set of items/projects/candidates $A = [m]$,
- a set of feasible sets/bundles/outcomes $F \subseteq \mathcal{P}(A)$ which we will usually define via a cost function $c : A \to \mathbb{N}$ and a budget $b \in \mathbb{N}$ where $B \in F$ iff $\sum_{j \in B} c(j) \leq b$ (when we have such a cost function, we may extend the definition of c so that for $B \subseteq A$, $c(B) = \sum_{j \in B} c(j)$),
- a set of voters $V = [n]$,
- a partial order over F which denotes how much a voter likes a bundle, where we will by default assume that this partial order is given by a utility function $u : V \times \mathcal{P}(A) \to \mathbb{R}$ which is additive, so $u(i, V) = \sum_{j \in B} u(i, \{j\})$, making it useful to define $u(i, j) := u(i, \{j\})$.

Definition We will consider several algorithms for computing outcomes given a participatory budgeting scenario. We may refer to them in several different ways, such as an aggregation rule, budgeting method, algorithm for computing an outcome, etc.

Definition An outcome B is Pareto optimal or efficient iff for every outcome C there is at least one voter who strictly prefers B to C or every voter is indifferent between B and C. If there is an outcome C for which no voter strictly prefers B to C and at least one voter strictly prefers C to B, then we call C a Pareto improvement over B (such a C exists iff B is not Pareto optimal).

Definition An aggregation rule is strategyproof iff once the utilities of all other voters has been fixed, the outcome resulting from voter i reporting their utilities truthfully is at least as good (from i's perspective) as any outcome that could be produced by this algorithm with the other voters' reported utilities fixed.

Definition A general class of strategyproof algorithms that return Pareto optimal outcomes are serial dictatorships.

In a serial dictatorship, the first voter restricts the set of feasible outcomes $F_0 = F$ to those outcomes which maximize their utility, F_1. The second voter then does the same, restricting F_1 to F_2, their most preferred outcomes in F_1, and so on for the rest of the voters in a fixed order. Any remaining outcome is then returned.

Suppose there are two feasible outcomes B, C such that C is a Pareto improvement over B. If C is eliminated at any point in the algorithm by a voter i, then B will also be eliminated by voter i (if it has not already been eliminated) as voter i likes B no more than C. If C is not eliminated by any voter, then B will be eliminated by a voter i who strictly prefers C to B. In either case B will not be returned, so whichever outcomes is returned will be Pareto optimal.

Now consider two runs of a serial dictatorship where the only difference in the input is that in the second run voter i may not report their utility accurately. Note that the first place in which this makes a difference is when voter i is restricting the feasible outcomes. In the first run, F_{i-1} is restricted to F_i, voter i's favourite outcomes in F_{i-1}. In the second run, F_{i-1} is restricted to F_i'. Note that the final outcome of the algorithm will be an element of F_i or F_i' for the respective runs, and no outcome in F_i' is preferred to any outcome in F_i, so the resulting outcome of the second run will not be preferred by voter i to the outcome of the first run.

2.2 Dichotomous background

Definition We call participatory budgeting scenarios where feasible outcomes are all of those with size exactly k ($F = \{B \subseteq A : |B| = k\}$ and $u(i, j) \in \{0, 1\}$ for all $i \in V$ and $j \in A$, Approval-Based Multi-Winner scenarios. In these scenarios we will typically represent voters' utilities by having for each voter i an "approval set" A_i of the items they assign a utility of 1 to.

The feasible outcomes being those of size k is what we might expect if we are filling multiple identical positions, like appointing a committee, hiring employees, or electing a parliament, and a nice property of approval-based voting is the simplicity of voting, as a voter just needs to decide whether or not they approve of each candidate, as opposed to other setups where they might have to rank the candidates or give real-valued utilities for the candidates.

When considering what kind of outcomes are desirable, we ideally want a balance between being fair by representing groups and finding efficient solutions which maximize the overall happiness.

Definition The Approval Voting rule returns the k candidates with the most approvals (breaking ties arbitrarily). Note that this will maximize the sum of the utilities attained by all the voters.

A significant problem with Approval Voting though is illustrated by the following scenario.

Example Suppose that $A = [100]$, $k = 50$, $V = [50]$, $A_i = [50]$ for $i \in [26]$ and $A_i = A \setminus [50]$ for $i \in V \setminus [26]$. Each candidate in [50] has 26 approvals, whereas each candidate in $A \setminus [50]$ has 24 approvals, so Approval Voting would choose [50] as the outcome. This may seem unfair though as there are 50 candidates to choose and 50 voters, but every single voter in $V \setminus [26]$ has none of their approved candidates selected and so are completely unrepresented, even though we might intuitively think each of the 50 voters should be represented by at least one of the 50 candidates.

This type of scenario is the motivation behind Aziz et al. [2] introducing the following JR, EJR, and core notions of fairness of outcomes of a voting scenario.

Definition A committee B provides justified representation (JR) if for every subset of voters $V' \subseteq V$ such that $|V'| \geq \frac{n}{k}$,

$$\bigcap_{i \in V'} A_i \neq \emptyset \implies \exists i' \in V' \ s.t. \ A_{i'} \cap B \neq \emptyset.$$

Intuitively, if a set of voters that agrees on a candidate is large enough to deserve representation by a candidate, then at least one of these voters has representation. Note that if i is the voter with representation, then if we consider the set $V' \setminus \{i\}$, it may still be large enough to deserve representation, so another voter in $V' \setminus \{i\}$ would be represented, and so on until the set is not big enough for representation.

Remark Aziz et al. [2] show that it can be checked in polynomial-time whether a committee B satisfies JR.

Example If we reconsider the example from before when $A = [100]$, $k = 50$, $V = [50]$, $A_i = [50]$ for $i \in [26]$ and $A_i = A \setminus [50]$ for $i \in V \setminus [26]$, we now have that a candidate from $[50]$ and a candidate from $A \setminus [50]$ must be in the outcome for it to provide JR, as for $V' = [26]$, $|V'| \geq \frac{50}{50}$ and $\bigcap_{i \in V'} A_i = [50] \neq \emptyset$, and similarly for $V' = V \setminus [26]$. Note though that once we have these two candidates, as every voter has a candidate they approve of in the outcome, JR is satisfied. Therefore the outcome $[49] \cup \{51\}$ satisfies JR. We now have that every voter has at least 1 representative, but this may not seem satisfactory, as the 24 voters in $V \setminus [26]$ have only one representative whereas the 26 voters in $[26]$ have 49 representatives.

Sánchez-Fernández et al. [15] proposed the following strengthening of JR.

Definition A committee B satisfies proportional justified representation (PJR) if for every subset of voters $V' \subseteq V$ and all $\ell \in [k]$, if $|V'| \geq \ell \cdot \frac{n}{k}$ then

$$\left|\bigcap_{i \in V'} A_i\right| \geq \ell \implies \left|\bigcup_{i \in V'} A_i \cap B\right| \geq \ell.$$

Intuitively, if a set of voters that agrees on ℓ candidates is large enough to deserve representation by ℓ candidates, then there are ℓ chosen candidates that are each approved by at least one of these voters.

Remark Aziz et al. [3] show that it is co-NP-complete to check whether a committee B satisfies PJR.

Example Considering the example from before one last time, when $A = [100]$, $k = 50$, $V = [50]$, $A_i = [50]$ for $i \in [26]$ and $A_i = A \setminus [50]$ for $i \in V \setminus [26]$, we now have that 26 candidates from $[50]$ and 24 candidates from $A \setminus [50]$ must be in the outcome for it to provide PJR, as for $V' = [26]$, $|V'| \geq 26 \cdot \frac{50}{50}$ and $\left|\bigcap_{i \in V'} A_i\right| = 50 \geq 26$, and similarly for $V' = V \setminus [26]$.

While the above example highlights some positive results, we now give another example in which voters do not have such unanimous groups which may give us motivation for a further strengthening.

Example Consider when $A = [100]$, $k = 50$, $V = [50]$, $A_i = \{i\} \cup ([100] \setminus [50])$. The committee $B = [50]$ satisfies PJR as each voter has a candidate they uniquely like in B, so any group of $\ell \cdot \frac{n}{k} = \ell$ voters has representation by ℓ candidates. Note though that all of the voters would be strictly better off with $[100] \setminus [50]$ as they all approve of all of these candidates.

This example motivates the following stronger definition by Aziz et al. [2]

Definition A committee B satisfies extended justified representation (EJR) if for every subset of voters $V' \subseteq V$ and all $\ell \in [k]$, if $|V'| \geq \ell \cdot \frac{n}{k}$ then

$$\left|\bigcap_{i \in V'} A_i\right| \geq \ell \implies \exists i' \in V' \ s.t. \ |A_{i'} \cap B| \geq \ell.$$

Intuitively, if a set of voters V' that agrees on ℓ candidates is large enough to deserve representation by ℓ candidates, then one of these voters i approves of at least ℓ of the chosen candidates. While this may not seem particularly strong, note that i is represented by ℓ candidates, and if we consider the set $V' \setminus \{i\}$, it may still be large enough to deserve representation by ℓ candidates, so another voter in $V' \setminus \{i\}$ is represented by ℓ candidates, and so on until the set is not big enough for a single candidate.

Example Considering the previous example, when $A = [100]$, $k = 50$, $V = [50]$, $A_i = \{i\} \cup ([100] \setminus [50])$. Taking $V' = V$, EJR states that at least one of the voters must like every candidate, so at least 49 candidates from $[100] \setminus [50]$ will be chosen. In particular, at least one of the voters will be as happy as possible, and all the others will be very close to as happy as possible if not as happy as possible.

Remark Aziz et al. [2] show that it is co-NP-complete to check whether a committee B satisfies EJR.

We now introduce a final strengthening of our fairness conditions, the core, though Peters et al. [11] state it is unknown if there always exists an outcome in the core.

Definition A committee B is in the <u>core</u> if for every subset of voters $V' \subseteq V$ and all $\ell \in [k]$, if $|V'| \geq \ell \cdot \frac{n}{k}$ then for all $S \subseteq A$ with $|S| \leq \ell$, there exists a voter $i \in V'$ such that

$$|A_i \cap B| \geq |A_i \cap S|.$$

Remark For an outcome B, B is in the core $\implies$ B satisfies EJR $\implies$ B satisfies PJR $\implies$ B satisfies JR. These implications follow directly from the definitions.

Peters and Skowron [12] give the following intuitive and efficient algorithm and show that it always finds an outcome that satisfies EJR (and so also PJR and JR).

Equal Shares Each voter i is given $b_i = \frac{k}{n}$ dollars.

We say that a candidate j is ρ-affordable for $\rho \geq 0$ if ρ is the least value such that

$$\sum_{i \in V} \min \{b_i, \mathbb{1}[j \in A_i] \cdot \rho\} = 1,$$

and we say it is affordable if it is ρ-affordable for any ρ, or equivalently

$$\sum_{i \in V} b_i \cdot \mathbb{1}[j \in A_i] \geq 1.$$

Starting with an empty outcome $B = \emptyset$, we take an item j which is ρ-affordable for the least ρ (breaking ties arbitrarily), add j to B, and subtract $\min \{b_i, \mathbb{1}[j \in A_i] \cdot \rho\}$ from each b_i (ρ can be seen as the ratio dollars spent/utility gained). We repeat this until no item is ρ-affordable for any ρ, then extend B arbitrarily to a set of size k and terminate (any extension will satisfy EJR).

Note that finding a ρ for which an item j is ρ-affordable (or whether no such ρ exists) is solvable in $O(n)$ time as for fixed $b_1, ..., b_n$,

$$f_j(\rho) = \sum_{i \in V} \min \{b_i, \mathbb{1}[j \in A_i] \cdot \rho\}$$

is a continuous non-decreasing piecewise linear function with breakpoints exactly at

$$\{b_i : i \in V, \mathbb{1}[j \in A_i]\}.$$

2.3 Properties of aggregation rules

To implement participatory budgeting in the real world, there is a good chance that the people in charge, who may not have any background in the subject, will have to be convinced that the algorithm/rule being used is a reasonable one. For this reason, it can be desirable to not only have guarantees on outcomes having certain properties, like the fairness properties previously discussed, but also have easily accessible information about how small changes in the input could change the output, and intuitive relations between different outcomes for similar inputs.

Talmon and Faliszewski [16] consider a few different utility functions for which we can see some intuitive applications. The utility functions considered in this paper all made use of approval sets as described in Section 2.2, though the feasible outcomes they consider are defined using a cost and budget rather than having a fixed size.

The first function discussed was $u(v, B) = \mathbb{1}[A_v \cap B \neq \emptyset]$, which can be seen as voter v being happy iff one of their items is selected. This could be used in a pizza party where A is the set of pizza types, c gives the cost of a pizza type, and a voter is happy iff there is a type of pizza they like. The second function discussed was $u(v, B) = |A_v \cap B|$, which is voter v's happiness being proportional to the number of their items being selected. This could be used by a group deciding which movies to purchase, and a voter's happiness is proportional to the number of movies they can now watch that they would enjoy. The third function discussed was $u(v, B) = \sum_{a \in A_v \cap B} c(a)$, which is voter v's happiness being proportional to the amount of money spent on the items they approve of. This could also be used by a group deciding which movies to purchase, but now if the group does not purchase a movie that a voter likes, they will buy themselves a personal copy.

For each of these utility functions, Talmon and Faliszewski [16] then consider three different rules for choosing bundles: one which maximizes the sum of utilities, one which iteratively constructs the bundle by greedily taking the next item that is still affordable and maximizes the marginal utility, and one which iteratively constructs the bundle by greedily taking the next item that is still affordable and maximizes the marginal utility divided by the marginal cost. For each of these nine rules, they then discussed computational complexity as well as whether or not they satisfy a short list of potentially desirable axioms.

Definition A budgeting method satisfies Inclusion Maximality if for all inputs A, c, b, V, u, the bundle it selects, B, is not a strict subset of any other feasible bundle.

This can be a desirable property as more things is often better, though if some items are disliked this may not be the case.

Definition A budgeting method satisfies Limit Monotonicity if for all inputs A, c, b, V, u, either there is an item j for which $c(j) = b + 1$, or the bundle it selects, B, is contained in the bundle it selects for $A, c, b + 1, V, u$.

This can be a desirable property when the budget might not be fully fixed beforehand, as if the budget increasing causes voters' choices to be unselected, this could be seen as a manipulation of the results.

Definition A budgeting method satisfies Discount Monotonicity if for all inputs A, c, b, V, u, for all items j in the bundle it selects, if the cost of j were decreased, then it would still be selected.

This is a desirable property as it can motivate the people proposing projects to make them more affordable.

Definition A budgeting method satisfies Splitting Monotonicity if for all inputs A, c, b, V, u, for all items j in the bundle it selects, the bundle that would be selected for $A \cup A' \setminus \{j\}, c', b, V, u'$, where $c'(A') = c(j)$, and c' and c agree otherwise, and all voters that approved of j approve of all items in A', contains at least one element of A'.

Intuitively, this states that if an item that would have been selected is decomposed into parts, then at least one of those parts will still be selected.

Definition A budgeting method satisfies Merging Monotonicity if for all inputs A, c, b, V, u, for all sets of items A' contained in the bundle it selects, such that each voter approves of all of A' or none of A', the bundle that would be selected for $A \cup \{j\} \setminus A', c', b, V, u'$, where $c(A') = c'(j)$, and c' and c agree otherwise, and all voters that approved of all of A' approve of j, contains j.

Intuitively, this states that if a set of items A' is such that each person likes all of A' or none of A', and A' would have been entirely selected, and A' is combined into one item, then it will still be selected.

Remark The previous two axioms are collectively useful for allowing proposers of projects to not have to worry about how many parts their proposal comes in.

3 The Knapsack problem

We will now formally define the knapsack problem for our use in this paper, a slight variation on a problem in Karp's famous list of 21 NP-complete problems [9].

Definition An instance of the knapsack problem has

- a set of items/projects/candidates $A = [m]$,
- a set of feasible sets/bundles/outcomes $F \subseteq \mathcal{P}(A)$ which is defined via a cost function $c : A \to \mathbb{N}$ and a budget $b \in \mathbb{N}$ where $B \in F$ iff $c(B) \leq b$,
- an injective utility function $u : \mathcal{P}(A) \to T$, where T is a totally ordered set,

and as a goal, wants to find a bundle with the greatest utility.

Remark Typically, T will be $\mathbb{R}, \mathbb{Q}, \mathbb{Z}$, or $\mathbb{N}$, but we allow for more flexibility so that we can optimize lexicographically over vectors of utilities.

Remark Other common variations on this problem are to consider the decision problem of whether a total utility of v is attainable, the optimization problem of finding the largest total utility attainable, and the problem of finding an affordable bundle with total utility of at least v if such a bundle exists. Finding optimal outcomes is typically what we desire in participatory budgeting, so that is what we will focus on, but note that given an optimal outcome we can quickly obtain answers to the other 3 problems, so it is at least as hard as the other problems.

As noted before, Karp [9] proved that when $T = \mathbb{Z}^+$, the decision problem of whether a total utility of v is attainable is NP-complete, but if the costs, budget, or utilities are given in unary, then it can be solved in polynomial time, as we will show shortly.

Remark Note that an instance of the knapsack problem is equivalent to an instance of our standard participatory budgeting scenario with exactly one voter and utilities restricted to $\mathbb{Z}^+$.

Theorem 3.1 *When $T = \mathbb{Z}_{\geq 0}$, there is an algorithm to find an optimal outcome of the knapsack problem that runs in $\Theta(m \cdot u(A))$ time.*

Proof Vazirani [18] gives us the following dynamic program.

Let $s[j, v]$ be a smallest cost bundle with items in $[j]$ and total utility exactly v, or some exception value *None* if no such bundle exists. We define $None \cup B := None$ and $c(None) = \infty$. We have

$$s[j,v] = \begin{cases} \{\} & \text{if } v = 0 \\ None & \text{else if } j = 0 \\ s[j-1, v-u(j)] \cup \{j\} & \text{else if } u(j) \leq v \text{ and} \\ & c\left(s[j-1, v-u(j)] \cup \{j\}\right) < c(s[j-1, v]) \\ s[j-1, v] & \text{else.} \end{cases}$$

We can then fill out a 2-dimensional array containing s-values by sequentially calculating $s[0,0], s[0,1], ..., s[0,u(A)], s[1,0], s[1,1], ..., s[m,u(A)]$, and then return the first bundle in the sequence $s[m,u(A)], s[m,u(A)-1], ..., s[m,0]$ whose cost is at most b.

Note that calculating the cost of a bundle in our third case takes $O(m)$ time, but if we store the costs in the array along with the bundles we can remove this extra factor. ∎

Definition A generalization of the knapsack problem is the multi-dimensional knapsack problem [10]. An instance of the knapsack problem has

- a set of items/projects/candidates $A = [m]$,
- a set of feasible sets/bundles/outcomes $F \subseteq \mathcal{P}(A)$ which is defined via cost functions $c_1, ..., c_d : A \to \mathbb{N}$ and budgets $b_1, ..., b_d \in \mathbb{N}$ where $B \in F$ iff $c_i(B) \leq b_i$ for all $i \in [d]$,
- an injective utility function $u : \mathcal{P}(A) \to T$, where T is a totally ordered set,

and as a goal, wants to find a bundle with the greatest utility.

Theorem 3.2 *There is an algorithm to find an optimal outcome of the multi-dimensional knapsack problem that runs in time in*

$$O\left(\max\{d,t\} \cdot m \cdot \prod_{i=1}^{d} (\min\{b_i, c_i(A)\} + 1)\right),$$

where t is the maximum time it takes to either compare two elements of T or compute $u(B \cup \{j\})$ given $u(B)$ for any B and j.

Proof This algorithm is based on a dynamic programming approach illustrated by Andonov et al. [1] Let $r[j, b'_1, ..., b'_d]$ be an optimal outcome with items in $[j]$ and budgets $b'_1, ..., b'_d$. We have

$r[j, b'_1, ..., b'_d] =$

$$\begin{cases} \emptyset & \text{if } j = 0 \\ r[j-1, b'_1, ..., b'_d] & \text{else if } c_i(j) > b'_i \text{ for any } i \in [d] \text{ or} \\ & u(r[j-1, b'_1, ..., b'_d]) > \\ & u(r[j-1, b'_1 - c_1(j), ..., b'_d - c_d(j)] \cup \{j\}) \\ r[j-1, b'_1 - c_1(j), ..., b'_d - c_d(j)] \cup \{j\} & \text{else.} \end{cases}$$

We can fill out a $(d+1)$-dimensional array storing r-values with indices from 0 to m, 0 to $\min\{b_1, c_1(A)\}$, ..., 0 to $\min\{b_d, c_d(A)\}$ in lexicographical order, then query this array on $r\left[m, \min\{b_1, c_1(A)\}, ..., \min\{b_d, c_d(A)\}\right]$ to get a desired outcome.

Note that if we store the utilities in the array along with the bundles, we can calculate query the previously calculated entries in $O(d)$, the utilities in the if-statement in $O(t)$, compare the utilities in the if-statement in $O(t)$, and calculate the new utilities in $O(t)$, for a total of $O(\max\{d, t\})$ time, giving the desired total runtime. ∎

Remark This algorithm can, and often is, used with dimension 1.

4 Pareto optimality in our standard setting

4.1 Introduction

In any situation with multiple parties and multiple possible outcomes, when considering the efficiency of outcomes, Pareto optimality is a desirable property. Given an outcome that is not Pareto optimal, no voter has any reason to object to switching outcomes to a Pareto improvement, and at least one voter would actively desire this switch. For this reason, being able to find Pareto optimal outcomes, verify Pareto optimality, and find Pareto improvements, are important tools to have available when possible.

Unfortunately, Aziz and Monnot [5] show that in our standard participatory budgeting setup, even with uniform cost items and utilities in $\{0, 1\}$, it is NP-hard to verify Pareto optimality (and finding Pareto improvements is at least that hard). We will elaborate on their results in Section 6.1, but this is certainly not a promising start.

A further negative result is that in our standard participatory budgeting setup with utilities in $\mathbb{Z}_{\geq 0}$, even if we only have one voter, computing a Pareto optimal outcome is NP-hard. This is directly from the NP-hardness of the knapsack problem, discussed in Section 3.

We therefore investigate our standard participatory budgeting setup with a few different restrictions to get some positive results.

4.2 Computing a Pareto optimal outcome when there is a constant number of distinct costs

Theorem 4.1 *In our usual participatory budgeting setup but with M a bound on the number of distinct costs, there exists an efficient strategyproof algorithm that returns a Pareto optimal outcome.*

Proof Consider when the number of distinct costs is bounded by some M.

We give a serial dictatorship algorithm for finding an outcome. Recall from section 2.1 that this is a strategyproof way to obtain a Pareto optimal outcome. We included an implementation of this in Python in Appendix A in case the more informal description was not sufficiently convincing, but the more informal description is as follows.

For $c_1 > \cdots > c_M$ the distinct costs, and m_p the number of items with cost c_p, we first make a list of all maximal tuples (with respect to each entry) of nonnegative integers such that each tuple $(t_1, ..., t_M)$ is such that $\sum_{p=1}^{M} c_p * t_p \leq b$ and $t_p \leq m_p$

for each $p \in [M]$.

For each tuple $(t_1, ..., t_M)$, the corresponding set of feasible solutions is all sets of items for which there are at most t_p items of cost c_p for each $p \in [M]$. Note that every feasible solution corresponds to at least one of these tuples. The feasible solutions for a given tuple can then be split up by cost, so for a given tuple $(t_1, ..., t_M)$ and a given cost c_p we have the feasible solutions are all sets of size no more than t_p of items which each cost c_p, then taking the Cartesian product of the feasible sets for each cost give the feasible sets for the tuple.

Note that for a given tuple $(t_1, ..., t_M)$, voter 1 can restrict the feasible outcomes to those which are optimal for them by restricting the feasible outcomes for each cost in this tuple to optimal outcomes. Once they have done this, they can delete any tuples which do not attain their optimal utility, leaving only feasible outcomes which are optimal for them. The remaining voters can then do the same in order, until we arrive at a set of feasible outcomes resulting from a serial dictatorship.

Note that the number of tuples is no more than

$$\prod_{p=1}^{M} (m_p + 1) \leq \left(\frac{\sum_{p=1}^{M} (m_p + 1)}{M} \right)^M = \left(\frac{m + M}{M} \right)^M,$$

which is polynomial for a fixed M. The feasible outcomes for a fixed tuple and cost can then be feasibly restricted by each voter to those which are optimal for them, as we do in the code, so overall this can be done efficiently. ∎

4.3 Computing a Pareto optimal outcome when costs or budget is polynomially bounded

Theorem 4.2 *In our usual participatory budgeting setup, there exists a strategyproof algorithm that returns a Pareto optimal outcome that runs in*

$$O\left(n \cdot m \cdot \min\left\{b, c(A)\right\}\right).$$

Proof Using the 1-dimensional version of the algorithm given in Theorem 3.2, where we let the utility function return the vector $(u(1, B), ..., u(n, B))$ and have the ordering on these vectors be lexicographical, we can get an outcome in $O\left(n \cdot m \cdot \min\left\{b, c(A)\right\}\right)$ time that would maximize the lexicographical utility (be chosen by a serial dictatorship) subject to the budget. ∎

4.4 Turning bads into goods

Negative utilities have been far less studied than positive utilities, so we make a few remarks about commonalities and differences with positive utilities, before continuing on with our results.

Consider the situation where we have a set of "bads" $A = [m]$, a value function $v : A \to \mathbb{N}$ and a quota $q \in \mathbb{N}$ such that $B \in F$ iff $\sum_{j \in B} v(j) \geq q$, $V = [n]$, and we have an additive utility function $u : V \times F \to \mathbb{R}_{\leq 0}$.

This is equivalent to the situation with $A = [m]$, a cost function $c = v$ and a budget $b = -q + \sum_{j \in A} v(a)$ such that $B \in F$ iff $c(B) \leq b$, $V = [n]$, and $u' : V \times F \to \mathbb{R}_{\leq 0}$ where $u'(i, S) = u(i, A) - u(i, S)$. What we have done here is essentially "by default" added all of the items, and now our budget is how much value we can lose while still meeting the quota.

Note that for each voter i, $u(i, A)$ is a constant, and $(-u(i, S)) : V \times \mathcal{P}(A) \to \mathbb{R}_{\geq 0}$ is an additive function. We can therefore see u' as indicating that everyone's utility is an additive function of the bads selected plus some baseline utility value. This essentially reduces the problem with bads (non-liked items) to the more standard problem with goods (non-disliked items).

4.5 Mixing goods and bads

As discussed previously, we can turn a problem about bads into a problem about goods, so given a situation with items such that each item is either disliked by nobody or liked by nobody, a budget, and costs in $\mathbb{Z}$, we can similarly turn it into a situation where all the costs are non-negative by adding the negative cost items to the outcome "by default" and having a cost to remove them. We are then left with a scenario which is essentially a problem about goods (there may be some items which no one likes which have a non-negative cost which we would trivially never add to an outcome).

More interesting is when some items are liked by some people and disliked by others. Even with no budget and no costs, what is or is not fair isn't clear.

Example Consider when $A = [2]$, $V = [3]$, $u(1, 1) = u(2, 2) = u(3, 2) = 1$ and all other utilities are -1. Is it preferable to have utilities (0, 0, 0) or $(-1, 1, 1)$?
If (0, 0, 0) is preferable, what about similar scenarios where we can choose between (0, ..., 0) vs $(-1, 1, ..., 1)$? If $(-1, 1, 1)$ is preferable, what about $(\underbrace{-1, ..., -1}_{n \text{ copies}}, \underbrace{1, ..., 1}_{n+1 \text{ copies}})$?

4.6 co-NP-completeness of verifying Pareto optimality when utilities are in $\{-1, 0, 1\}$ and there are no costs

As discussed at the start of the section, Aziz and Monnot [5] show that in our standard participatory budgeting setup, even with uniform cost items and utilities in $\{0, 1\}$, it is NP-hard to verify Pareto optimality. We show that if we add -1 to the allowed utilities, it is co-NP-complete to verify Pareto optimality of the empty selection even if we restrict the budget to be unlimited and each item to have at most 4 voters with non-zero utility for it.

Theorem 4.3 *In our usual participatory budgeting setup, but restricted to utilities in* $\{-1, 0, 1\}$*, unlimited budget (*$F = \mathcal{P}(A)$*), and for all items* $j \in A$*, we have* $|\{i \in V : u(i, j) \neq 0\}| \leq 4$*, it is co-NP-complete to verify Pareto optimality over the empty selection.*

Proof Note that a Pareto improvement over the empty selection is a witness to the empty selection not being Pareto optimal, so our problem is in co-NP.

We show hardness by a reduction from Set Cover which Karp [9] showed to be NP-complete. An instance of Set Cover has $S_1, ..., S_m \subseteq [n]$ and $k \in [m]$, and asks if there is a set $Q \subseteq \{S_1, ..., S_m\}$ with $|Q| \leq k$ such that $\bigcup_{S \in Q} S = [n]$.

Suppose we have an instance of Set Cover, with $S_1, ..., S_m \subseteq [n]$ and $k \in [m]$. If $n = 0$ we can return True and if $k \leq 2$ this can be solved by checking all $\binom{m}{k}$ possibilities in quadratic time. Otherwise, let $A = \{s_1, ..., s_m, a_1, ..., a_k\}$,
$V = \{e_1, ..., e_n, c, h, \ell_1, ..., \ell_k\}$, and u be as following.

$$
\begin{aligned}
u(e_i, s_j) &= \mathbb{1}\,[i \in S_j] \\
u(e_i, a_j) &= -\mathbb{1}\,[j = 1] \\
u(c, s_j) &= -1 \\
u(c, a_j) &= 1 \\
u(h, x) &= 1 \quad \text{for any } x \in A \\
u(\ell_i, s_j) &= 0 \\
u(\ell_i, a_i) &= -1 \\
u(\ell_i, a_{(i \mod k)+1}) &= 1 \\
u(\ell_i, a_j) &= 0 \quad \text{for any } j \notin [k] \setminus \{i, (i \mod k) + 1\}
\end{aligned}
$$

We claim that there is a Pareto-improvement over the empty selection of this problem iff there was a set cover of size at most k, which is sufficient to complete the proof.

Suppose $B \subseteq A$ is a Pareto-improvement over the empty selection. If B contains any item $s_{j'}$, then it must also contain an item a_j so that voter c is not worse off. Therefore as B is not the empty selection it must contain an item a_j. Note that such an item a_j contributes -1 utility to voter ℓ_j, so if we look at the last k items as a k-cycle, the next item must also be in B so that that voter does not have negative utility. Continuing like this, we have that $\{a_1, ..., a_k\} \subseteq B$. Note that

$$\begin{aligned} u(e_i, \{a_1, ..., a_k\}) &= -1 \\ u(c, \{a_1, ..., a_k\}) &= k \\ u(h, \{a_1, ..., a_k\}) &= k \\ u(\ell_i, \{a_1, ..., a_k\}) &= 0 \end{aligned}$$

so $u(e_i, B \cap \{s_1, ..., s_m\}) \geq 1$ for all $i \in [n]$ (voters must have net non-negative utility for B) and $|B \cap \{s_1, ..., s_m\}| \leq k$ (voter c gets -1 utility for each item in $B \cap \{s_1, ..., s_m\}$). As

$$1 \leq u(e_i, B \cap \{s_1, ..., s_m\}) = \sum_{s_j \in B \cap \{s_1, ..., s_m\}} u(e_i, s_j) = \sum_{s_j \in B \cap \{s_1, ..., s_m\}} \mathbb{1}\left[i \in S_j\right],$$

we have that $i \in \bigcup_{s_j \in B \cap \{s_1, ..., s_m\}} S_j$ for all $i \in [n]$, and since $|B \cap \{s_1, ..., s_m\}| \leq k$ we have that a set cover of size $\leq k$ exists.

Now instead suppose there is a set cover $Q = \left\{S_{j_1}, ..., S_{j_{k'}}\right\}$ with $|Q| \leq k$. Let $C = \{s_j : S_j \in Q\}$. Let $B = C \cup \{a_1, ..., a_k\}$. For $i \in [n]$,

$$u(e_i, B) = \sum_{s_j \in C} \mathbb{1}\left[i \in S_j\right] + u(e_i, \{a_1, ..., a_k\}) = |\{s_j \in C : i \in S_j\}| - 1 \geq 0,$$

$$u(c,B) = \sum_{s_j \in C} -1 + u(c, \{a_1, ..., a_k\}) = -|C| + k \geq 0,$$

$$u(h,B) = \sum_{s_j \in C} 1 + u(h, \{a_1, ..., a_k\}) = |C| + k > 0,$$

and for $i \in [k]$,

$$u(\ell_i, B) = \sum_{s_j \in C} 0 + u(\ell_i, \{a_1, ..., a_k\}) = 0 + 0,$$

so there is a Pareto improvement over the empty selection. ∎

5 Proportionality in Committee Selection with Negative Feelings

5.1 Section overview

In this section we will cover a generalization of Approval-Based Multi-Winner scenarios given by Talmon and Page [17], where instead of approving or disapproving, voters can approve, disapprove, or be neutral about each candidate. We will briefly discuss their notions of fairness, go over several of corrections to their paper, then consider a further notion of fairness, reasons for this choice, and the complexity of computing fair outcomes and verifying fairness of outcomes.

5.2 Trichotomous background

Similarly to the dichotomous setting, Talmon and Page [17] consider scenarios where feasible outcomes are all of those with size exactly k, but now they have each voter i divide the items into items they approve of (A_i^+), items they disapprove of (A_i^-), and items they are indifferent about (A_i^0). This allows for voters to express a somewhat more detailed opinion, as there can be a significant difference between being indifferent about a particular project being implemented and being actively opposed to it.

In the dichotomous setting, the fairness axioms were of the form "If there is a large enough group of voters with similar enough opinions, then they deserve at least this much representation." Talmon and Page [17] consider two classes of fairness axioms.

The first is fairly liberal with what is allowed for "similar enough opinions", in that they only require $\left|\left(\bigcup_{i\in V'} A_i^+\right) \setminus \left(\bigcup_{i\in V'} A_i^-\right)\right|$ to be sufficiently large, and then they consider a variety of options for what such a group would deserve, including making sure they have sufficient representation within the candidates they approve of (and sometimes also those for which they are neutral) and making sure candidates they disapprove of are not selected.

The second class of axioms is more strict in what it means to have "similar enough opinions", in that they require $\left|\bigcap_{i\in V'} A_i^+\right|$ to be large enough. This is very similar to the criteria we have seen in the dichotomous setting.

5.3 Corrections

We first go over several of points that seem to be incorrect in the paper. [1]

The first point relates to the following definition Talmon and Page [17] give.

Definition A committee B satisfies <u>weakest axiom (WA)</u> if for every subset of voters $V' \subseteq V$ and all $\ell \in [k]$, if $|V'| \geq \ell \cdot \frac{n}{k}$ then

$$\left|\left(\bigcup_{i\in V'} A_i^+\right) \setminus \left(\bigcup_{i\in V'} A_i^-\right)\right| \geq \ell \implies \left|\left(\left(\bigcup_{i\in V'} A_i^+\right) \cup \left(\bigcup_{i\in V'} A_i^0\right)\right) \cap B\right| \geq \ell.$$

In this paper they claim that WA is not always satisfiable, but we claim that WA is no more demanding than PJR, and [15] shows that PJR is always satisfiable.

Proposition 5.1 *WA is always satisfiable.*

Proof Suppose we have a trichotomous scenario. Consider the dichotomous sce-

[1]This paper is an arXiv preprint.

nario with $A_i = A_i^+ \cup A_i^0$. A PJR committee committee exists by [15], so let B be such a committee. Let $V' \subseteq V$ and $\ell \in [k]$ be arbitrary such that $|V'| \geq \ell \cdot \frac{n}{k}$. Suppose $\left|\left(\bigcup_{i\in V'} A_i^+\right) \setminus \left(\bigcup_{i\in V'} A_i^-\right)\right| \geq \ell$.

$$\left(\bigcup_{i\in V'} A_i^+\right) \setminus \left(\bigcup_{i\in V'} A_i^-\right) \subseteq A \setminus \left(\bigcup_{i\in V'} A_i^-\right) = \bigcap_{i\in V'} \overline{A_i^-} = \bigcap_{i\in V'} \left(A_i^+ \cup A_i^0\right),$$

so $\left|\bigcap_{i\in V'} \left(A_i^+ \cup A_i^0\right)\right| \geq \ell$. Therefore $\left|\bigcap_{i\in V'} A_i\right| \geq \ell$, so since B satisfies PJR, $\left|\left(\bigcup_{i\in V'} A_i\right) \cap B\right| \geq \ell$. Therefore $\left|\left(\left(\bigcup_{i\in V'} A_i^+\right) \cup \left(\bigcup_{i\in V'} A_i^0\right)\right) \cap B\right| \geq \ell$, so WA is satisfied. ∎

The second point relates to the following definition Talmon and Page [17] give, though we restate it slightly to avoid potentially ambiguous negations.

Definition A committee B satisfies New Cohesiveness Representation (NCR) if for every subset of voters $V' \subseteq V$ and all $\ell \in [k]$, if $|V'| \geq \ell \cdot \frac{n}{k}$ then

$$\left|\bigcap_{i\in V'} A_i^+\right| \geq \ell \implies \left|\bigcap_{i\in V'} A_i^+ \cap B\right| \geq \ell.$$

In this paper they claim that NCR is always satisfiable, but we give a counterexample here.

Example Let $A = [3]$, $k = 2$, $V = [3]$, $A_i^+ = [3] \setminus \{i\}$. Note that for $V' = \{1, 2\}$, $|V'| \geq 1 \cdot \frac{3}{2}$ and $\left|\bigcap_{i\in V'} A_i^+\right| \geq 1$, so for NCR to be satisfied it must be the case that $\left|\bigcap_{i\in V'} A_i^+ \cap B\right| \geq 1$ where $\bigcap_{i\in V'} A_i^+ = \{3\}$. In particular it must be the case that $3 \in B$. Note that symmetrically by taking $V' = \{2, 3\}$ and $V' = \{1, 3\}$, we get that $1, 2 \in B$ which is impossible as it must be the case that $|B| \leq k = 2$.

5.4 A further axiom

As Talmon and Page [17] point out while exploring their axioms, the first class of axioms for the most part is not always satisfiable, and the one that is always satisfiable is no stronger than PJR.

We also noticed that the second class of axioms did not make use of either the A_i^0 or the A_i^-, which means that for all intents and purposes for each voter i it only takes into account whether or not items are in A_i^+, which effectively reduces things to the dichotomous setting. We can therefore note their proof of satisfiability of NCR works if we replace NCR with the similar axiom EJR in the dichotomous setting.

We therefore tried to come up with a balance between using the trichotomous setting while still having a strong axiom which is always satisfiable. In doing so we noticed that if we consider each voter getting 1 unit of utility for every liked item selected and -1 unit of utility for every disliked item selected, then this is a fairly symmetric problem where we are partitioning the items into two sets, a "selected" set or committee B and an "unselected" set $C = \overline{B}$.

In particular, we can see this symmetry by comparing utility functions

$$\begin{aligned}
u^+(i, B) &= \left|A_i^+ \cap B\right| - \left|A_i^- \cap B\right| \\
u^-(i, B) &= \left|A_i^- \cap \overline{B}\right| - \left|A_i^+ \cap \overline{B}\right| \\
u^\pm(i, B) &= \left|A_i^+ \cap B\right| + \left|A_i^- \cap \overline{B}\right|
\end{aligned}$$

In particular, note that

$$u^+(i, B) + \left|A_i^-\right| = u^-(i, B) + \left|A_i^+\right| = u^\pm(i, B),$$

so these utility functions are the same up to an additive constant for each voter, so maximizing one of these functions for a particular voter is equivalent to maximizing all of them.

Definition For B a committee and $C = \overline{B}$, B satisfies <u>symmetric trichotomous justified representation (STJR)</u> if each of the following holds.

- For all sets of voters $V' \subseteq V$ for all $\ell \in \mathbb{N}$, if $|V'| \geq \ell \cdot \frac{n}{k}$, then for some $i \in V$,

$$\left|A_i^+ \cap B\right| \geq \min\left(\ell, \left|\bigcap_{i \in V'} A_i^+\right| + \ell - k\right)$$

- For all sets of voters $V' \subseteq V$ for all $\ell \in \mathbb{N}$, if $|V'| \geq \ell \cdot \frac{n}{m-k}$, then for some $i \in V$,

$$\left|A_i^- \cap C\right| \geq \min\left(\ell, \left|\bigcap_{i \in V'} A_i^-\right| + \ell - (m-k)\right).$$

Remark To get some intuition for this axiom, we first note that if we remove the $+\ell - k$ in the first part of the axiom from the second argument to min, this becomes EJR where $A_i = A_i^+$, and a symmetric thing holds for the other part.

Example In this EJR format though it is not always satisfiable. Consider $A = [4]$, $k = 2$, $V = [2]$, $A_1^+ = \{1\}$, $A_2^- = \{1\}$. Voter 1 makes up $1 \cdot \frac{n}{k}$ of the voters and has $\left|A_1^+\right| = 1$, so 1 would have to be selected. On the other hand, voter 2 makes up $1 \cdot \frac{n}{m-k}$ of the voters and has $\left|A_2^-\right| = 1$, so 1 would have to be not selected.

Remark What this example illustrates is we have a problem when groups are able to decide which specific set of items is selected/unselected, as this could directly conflict with another group's desires. We instead make it so that if a group agrees on k items to be selected (or $m-k$ items to not be selected), then they are proportionally represented in the selected items (or unselected items). Note that if $\left|\bigcap_{i \in V'} A_i^+\right| \geq k$ (or the equivalent in the other case), then this reduces to the EJR axiom version, but for every item they are short of k, they lose one fewer guaranteed item in representation.

Theorem 5.2 *There is a polynomial-time algorithm for computing an outcome which satisfies STJR.*

Proof

Begin algorithm description

If $k \leq \frac{m}{2}$, use Equal Shares (without the arbitrary extension at the end) on the dichotomous scenario with candidates in A, $A_i = A_i^+$, and with committee of size k to get a committee B'. If $|B'| < k$, use Equal Shares (with the arbitrary extension at the end) on the dichotomous scenario with candidates in $A \setminus B'$, $A_i = A_i^- \cap (A \setminus B')$ and with committee of size $m - k$ to get a committee C. Set $B = \overline{C}$ and return these.

If $k > \frac{m}{2}$ then start with C' instead.

End algorithm description

We first note that this alternate definition of EJR is equivalent to the original (though more closely resembles STJR).

Definition A committee B satisfies <u>extended justified representation (EJR)</u> if for every subset of voters $V' \subseteq V$ and all $\ell \in [k]$, if $|V'| \geq \ell \cdot \frac{n}{k}$ then for some $i \in V$,

$$\left|A_i^+ \cap B\right| \geq \min\left(\ell, \left|\bigcap_{i \in V'} A_i^+\right|\right).$$

Note that Peters and Skowron [12] showed that Equal Shares satisfies EJR, so (in the case where $k \leq \frac{m}{2}$) B' will satisfy EJR in the corresponding dichotomous setting (so B will too). As we can see by comparing our alternate definition of EJR to the first part of STJR (and noting that $\ell \leq k$), B satisfying EJR in the corresponding dichotomous setting guarantees that the first part of STJR is satisfied.

Now suppose there is a set of voters V' with $|V'| \geq \ell \cdot \frac{n}{m-k}$. Let $B'' \subseteq B'$ be the items that voters in V' did not spend any money on in the first Equal Shares algorithm.

Note that

$$
\begin{aligned}
|B''| &\leq \frac{k}{n} \cdot (|V \setminus V'|) \\
&\leq \frac{k}{n} \cdot \left(n - \ell \cdot \frac{n}{m-k}\right) \\
&= k \cdot \left(1 - \frac{\ell}{m-k}\right) \\
&= \frac{k}{m-k} \cdot (m - k - \ell) \\
&\leq m - k - \ell,
\end{aligned}
$$

so in particular

$$
\begin{aligned}
\left|\bigcap_{i \in V'} A_i^- \cap \overline{B'}\right| &= \left|\bigcap_{i \in V'} A_i^-\right| - \left|\bigcap_{i \in V'} A_i^- \cap B'\right| \\
&\geq \left|\bigcap_{i \in V'} A_i^-\right| - |B''| \\
&\geq \left|\bigcap_{i \in V'} A_i^-\right| - (m - k - \ell)\,.
\end{aligned}
$$

Therefore when Equal Shares is being run the second time, we will have $|V'| \geq \ell \cdot \frac{n}{m-k}$ so for some $i \in V$,

$$
\left|A_i^- \cap C\right| \geq \min\left(\ell, \left|\bigcap_{i \in V'} A_i^- \cap \overline{B'}\right|\right) \geq \min\left(\ell, \left|\bigcap_{i \in V'} A_i^-\right| + \ell - (m-k)\right),
$$

as Equal Shares satisfies EJR. Therefore the second part of STJR will also be satisfied. ∎

Remark Unfortunately this axiom and Pareto optimality cannot always be simultaneously satisfied, where we use any of the previously described utility functions (which one we use is irrelevant as we previously noted that they are equivalent in this regard).

Example As such an example of incompatibility, consider the scenario where

$V = [3]$, $A = [12]$, $A_i^+ = \{j : j \equiv i \mod 3 \text{ and } j \leq 9\}$, $A_i^- = [9] \setminus A_i^+$, and $k = 3$. Note that $|A_i^+| = 3$, so the first part of STJR decrees that for each $i \in V$, $|A_i^+ \cap B| \geq 1$. In particular, $B \subset [9]$ and contains one item from each equivalence group mod 3. This gives each voter a total utility of -1. Note though that if we took $B = \{10, 11, 12\}$, each voter would have a strictly higher utility of 0.

Theorem 5.3 *It is co-NP-complete to check whether a committee B satisfies STJR.*

Proof To prove this we use a modification of the proof of hardness of verification of EJR given by Aziz et al. [2]

Note that checking if B satisfies STJR is in co-NP as if it does not satisfy STJR, then there is a set of voters V' for which STJR does not hold, and given V' it is easy to verify that the axiom does not hold.

To prove verification of STJR is co-NP-hard, we will reduce the Balanced Complete Bipartite Subgraph problem, which Garey and Johnson [8] have as [GT23] in their list of NP-complete problems, to our problem.

An instance of the Balanced Complete Bipartite Subgraph problem is given by an integer ℓ and a bipartite graph (L, R, E) where E is the set of edges going between the two disjoint sets of vertices L and R, and it asks "Does there exist $L' \subseteq L$ and $R' \subseteq R$ with $|L'| = |R'| = \ell$ such that for all $u \in L'$ and $v \in R'$, $\{u, v\} \in E$?"

Suppose we have an instance of the Balanced Complete Bipartite Subgraph problem, $(L, R, E), \ell$, with $|R| = s$. Note that if $\ell < 3$ then the problem can be solved in polynomial time with a brute force approach of checking all $\binom{|L|}{\ell} \cdot \binom{|R|}{\ell} \leq |L|^2|R|^2$ pairs of L', R' with $|L'| = |R'| = \ell$, and if $\ell \geq 3$ but $s < 3$ then the answer is clearly "no", so we assume $\ell, s \geq 3$.

We construct an instance of our problem as follows. We have

$A = C_0 \cup C_1 \cup C_1' \cup C_2 \cup C_3$ where C_0, C_1, C_1', C_2, C_3 are all disjoint and

$$C_0 = L, |C_1| = |C_1'| = \ell - 1, |C_2| = s\ell + \ell - 3s, |C_3| = \ell - 2,$$

and $V = V_0 \cup V_1 \cup V_2$ where V_0, V_1, V_2 are all disjoint and

$$V_0 = R, |V_1| = \ell(s-1), |V_2| = s\ell + \ell - 3s = (s+1)(\ell - 3) + 3 \geq 3.$$

We then have $A_i^+ = \{j : \{i,j\} \in E\} \cup C_1 \cup C_3$ for each $i \in V_0$, $A_i^+ = C_0 \cup C_1' \cup C_3$ for each $i \in V_1$, and there is a bijection $f : V_2 \to C_2$ where $A_i^+ = \{f(i)\}$ for each $i \in V_2$, and $A_i^- = \emptyset$ for all $i \in V$.

Finally we have $k = 2\ell - 2$ and $B = C_1 \cup C_1'$.

Note that for $n = |V|$, $n = s + \ell(s-1) + s\ell + \ell - 3s = 2s(\ell - 1)$, so $\frac{n}{k} = s$.

Suppose there exist $L' \subseteq L$ and $R' \subseteq R$ with $|L'| = |R'| = \ell$ such that for all $u \in L'$ and $v \in R'$, $\{u, v\} \in E$. Fix such L', R'. For $C^* = L'$, $V^* = R' \cup V_1$, [2] we have that $|V^*| = \ell s$ and $\bigcap_{i \in V^*} A_i^+ = L' \cup C_3$ with $|L' \cup C_3| = k$, but each voter in V^* only likes $\ell - 1$ of the candidates in B, so it is not the case that

$$\left|A_i^+ \cap B\right| \geq \min\left(\ell, \left|\bigcap_{i \in V'} A_i^+\right| + \ell - k\right),$$

for any $i \in V^*$, so STJR is not satisfied.

Now suppose instead that B does not satisfy STJR. Note that the second part of the axiom is trivially satisfied as all the A_i^- are empty, so it must be the first part of the axiom that is not satisfied. Letting V^*, ℓ' be the set of voters and the natural

[2] In the arXiv version of [2] there was a typo where the equivalent statement to $V^* = R' \cup V_1$ was written as $V^* = R \cup V_1$, which has been pointed out to one of the authors.

number for which it is not satisfied, we have

$$\left|A_i^+ \cap B\right| < \min\left(\ell', \left|\bigcap_{i \in V^*} A_i^+\right| + \ell' - k\right)$$

for all $i \in V^*$. Note that $V_2 \cap V^* = \emptyset$ as otherwise $\bigcap_{i \in V^*} A_i^+$ would be empty since $|V^*| > 1$ and all voters in V_2 share no liked items with anyone. Note then that it must be the case that

$$\min\left(\ell', \left|\bigcap_{i \in V^*} A_i^+\right| + \ell' - k\right) \geq \ell$$

since every voter in $V_0 \cup V_1$ has $\ell - 1$ of their liked candidates in B. We also have that $V^* \subseteq V_0 \cup V_1$, so $|V^*| \leq s + s\ell - \ell$. As $\ell' \cdot s \leq |V^*|$, we have $\ell' \cdot s \leq s + s\ell - \ell$, and

$$\begin{aligned}
&\ell' \cdot s \leq s + s\ell - \ell \\
\implies\ & \ell' \leq 1 + \ell - \frac{\ell}{s} \\
\implies\ & \ell' \leq \left\lfloor 1 + \ell - \frac{\ell}{s} \right\rfloor \\
\implies\ & \ell' \leq \ell
\end{aligned}$$

so $\ell' \leq \ell$. We also have

$$\ell' \geq \min\left(\ell', \left|\bigcap_{i \in V^*} A_i^+\right| + \ell' - k\right) \geq \ell,$$

so $\ell' = \ell$. From this we get that $|V^*| \geq \ell s$, so $|V^* \cap V_0| \geq \ell$ and $|V^* \cap V_1| \geq 1$. Since V^* contains voters from both V_0 and V_1, $\bigcap_{i \in V^*} A_i^+ \subseteq C_0 \cup C_3$. We therefore have that

$$\left|\bigcap_{i\in V^*} A_i^+\right| + \ell' - k \geq \ell$$

$$\implies \left|C_0 \cap \bigcap_{i\in V^*} A_i^+\right| + \left|C_3 \cap \bigcap_{i\in V^*} A_i^+\right| \geq k$$

$$\implies \left|C_0 \cap \bigcap_{i\in V^*\cap V_0} A_i^+\right| + \ell - 2 \geq 2\ell - 2$$

$$\implies \left|\bigcap_{i\in V^*\cap V_0} \{j : \{i,j\} \in E\}\right| \geq \ell,$$

so for $R'' = V^* \cap V_0 \subseteq R$ and $L'' = \bigcap_{i\in V^*\cap V_0} \{j : \{i,j\} \in E\} \subseteq L$, $|L''|, |R''| \geq \ell$ and for all $j \in L''$ and $i \in R''$, $\{u, v\} \in E$. Therefore for any $L' \subseteq L''$ and $R' \subseteq R''$ with $|L'| = |R'| = \ell$, we can see that the answer is yes for the Balanced Complete Bipartite Subgraph problem. ∎

6 Computing and testing Pareto optimal committees with ranked preferences

6.1 Uniform-cost background

Aziz and Monnot [5] investigate Pareto optimality in scenarios where $F = \{B \subseteq A : |B| = k\}$ for a variety of different classes of utility functions.

To define these classes of utility functions they have a preference profile $\succsim = (\succsim_1, ..., \succsim_n)$ where each preference relation $\succsim_i$ is a complete and transitive relation over A. For simplicity of notation they also write $j \succ_i j' \iff (j \succsim_i j'$ and $j' \not\succsim_i j)$ and $j \sim j' \iff (j \succsim j'$ and $j' \succsim j)$.

Definition For each voter i, the relation $\succsim_i$ induces a set of non-empty equivalence classes $E_i^1, ..., E_i^{k_i}$ where $j \sim_i j'$ for any $j, j' \in E_i^p$ for any p, and $j \succ_i j'$ for any $j \in E_i^p$ and $j' \in E_i^q$ for any $p < q$.

Aziz and Monnot did not as it was not strictly necessary, but we assume that $k_i \geq 2$ for all voters i as otherwise such a voter is completely indifferent about everything. This simplifies a few details later.

Definition We define the topwidth to be $tw(\succsim) := \max_{i \in [n]} |E_i^1|$.

Definition We say that the set of voters have dichotomous preferences if $k_i = 2$ for all i.

Definition We say that the set of voters have linear preferences if $k_i = m$ for all i.

Definition For $B \subseteq A$, $\max_{\succsim_i}(B) := E_i^p \cap B$ for p the least integer such that $E_i^p \cap B \neq \emptyset$.

Definition For $B \subseteq A$, $\min_{\succsim_i}(B) := E_i^p \cap B$ for p the greatest integer such that $E_i^p \cap B \neq \emptyset$.

Definition Given a preference relation $\succsim_i$ for a voter i, they then extend it to a partial order on the sets of size k, by defining for $B, C \in F$,

- responsive set extension (RS), $B \succsim_i^{RS} C$ iff there is a bijection $f : B \to C$ such that $j \succsim_i f(j)$ for all $j \in B$,
- best set extension ($\mathcal{B}$), $B \succsim_i^{\mathcal{B}} C$ iff $j \succsim_i j'$ for $j \in \max_{\succsim_i}(B)$ and $j' \in \max_{\succsim_i}(C)$,
- worst set extension ($\mathcal{W}$), $B \succsim_i^{\mathcal{W}} C$ iff $j \succsim_i j'$ for $j \in \min_{\succsim_i}(B)$ and $j' \in \min_{\succsim_i}(C)$,
- downward lexicographical extension (DL), $B \succsim_i^{DL} C$ iff $|B \cap E_i^p| = |C \cap E_i^p|$ for all p or $|B \cap E_i^p| > |C \cap E_i^p|$ for the least p for which $|B \cap E_i^p| \neq |C \cap E_i^p|$
- upward lexicographical extension (UL), $B \succsim_i^{UL} C$ iff $|B \cap E_i^p| = |C \cap E_i^p|$ for all p or $|B \cap E_i^p| < |C \cap E_i^p|$ for the greatest p for which $|B \cap E_i^p| \neq |C \cap E_i^p|$

Remark Aziz and Monnot [5] note that for all $B, C \in F$,

- $B \succsim_i^{RS} C \implies B \succsim_i^{DL} C \implies B \succsim_i^{\mathcal{B}} C$

- $B \succsim_i^{RS} C \implies B \succsim_i^{UL} C \implies B \succsim_i^{\mathcal{W}} C$
- $B \succ_i^{RS} C \implies B \succ_i^{DL} C$
- $B \succ_i^{RS} C \implies B \succ_i^{UL} C$

In particular, if a committee B is DL-efficient or UL-efficient then it is also RS-efficient. This follows by a simple proof by contraposition, where if we assume B is not RS-efficient then there is an RS Pareto improvement C over B, but C would then also be both a DL and UL Pareto improvement over B.

Example They give the following example to illustrate these relationships:

For $A = [4], k = 2, V = [2]$ and

$$1 \succ_1 2 \succ_1 3 \succ_1 4,$$

$$4 \succ_2 3 \succ_2 2 \succ_2 1,$$

- the unique $\mathcal{B}$-efficient committee is $\{a, d\}$,
- the $\mathcal{W}$-efficient committees are $\{a, b\}, \{b, c\}, \{c, d\}$, [3]
- the DL-efficient committees are $F \setminus \{\{b, c\}\}$,
- the UL-efficient committees are $F \setminus \{\{a, d\}\}$,
- and the RS-efficient committees are all of F.

The primary results given by Aziz and Monnot [5] are the complexities of computing Pareto optimal committees and verifying Pareto optimality for each of the different set extensions. In terms of computing Pareto optimal committees, they show that for $\mathcal{B}$ when there are linear preferences, and $RS, DL, UL, \mathcal{W}$, one can be found in polynomial time, but finding one for $\mathcal{B}$ in general is NP-hard. Verifying Pareto optimality for $\mathcal{W}$ is in P, but verifying Pareto optimality for $RS, DL, UL, \mathcal{B}$ are all

[3] Their example mistakenly claimed $\{b, c\}$ was the only $\mathcal{W}$-efficient committee.

co-NP-complete, even if we restrict to linear preferences or dichotomous preferences. We only get a polynomial-time algorithm for verifying Pareto optimality with dichotomous preferences for $RS, DL, UL, \mathcal{B}$ if we restrict to $tw(\succsim) \leq 2$ for the first three and $tw(\succsim) = 1$ for $\mathcal{B}$. If $tw(\succsim)$ is allowed to be even 1 bigger then they return to being co-NP-complete.

The algorithms Aziz and Monnot [5] give for computing Pareto optimal outcomes are all serial dictatorships, which they acknowledge are not particularly fair between the voters, but do have the desirable properties of being strategyproof and returning Pareto optimal outcomes.

6.2 Extension to non-uniform costs

We extend some of the results of Aziz and Monnot [5] to when the cost of each item is no longer uniform, so

$$F = \{B \subseteq A : c(B) \leq b\}.$$

Note that if we try to compare bundles of different sizes, then RS will never return a comparison and $\mathcal{W}$ and UL become degenerate as an empty selection is always optimal. To remedy this problem we do the following. We have each $\succsim_i$ be a complete and transitive order over $[m+1]$ (where $A = [m]$), where "item" $m+1$ is always ranked as worst (it can be tied for worst). When comparing sets B and C, we compare them as we would before, but where they are "padded" with copies of "item" $m+1$ until they are both of size $\max\{|D| : D \in F\}$.

Remark Note that for $B, C \in F$ with $B \subsetneq C$, it will never be the case that $B \succ_i^\epsilon C$ for any extension ϵ, so there is no need to consider outcomes in F that are not maximal with respect to set inclusion.

Remark The relations Aziz and Monnot [5] noted previously still hold in the non-uniform cost case. In particular, for all $B, C \in F$,

- $B \succsim_i^{RS} C \implies B \succsim_i^{DL} C \implies B \succsim_i^{\mathcal{B}} C$
- $B \succsim_i^{RS} C \implies B \succsim_i^{UL} C \implies B \succsim_i^{\mathcal{W}} C$
- $B \succ_i^{RS} C \implies B \succ_i^{DL} C$
- $B \succ_i^{RS} C \implies B \succ_i^{UL} C$

and if a committee B is DL-efficient or UL-efficient then it is also RS-efficient.

Remark Note that a uniform cost scenario from before can be represented in our setup by giving all items a cost of 1 and having a budget of k. Therefore any hardness results concerning verification of Pareto optimality or computing Pareto optimal solutions will still hold as we are generalizing the problem.

Theorem 6.1 *There exists an efficient algorithm that returns an outcome that is Pareto optimal under the RS extension.*

Aziz and Monnot [5] give an algorithm which is also strategyproof. To do this they used the fact that all sets of items of the same size cost the same amount, so given a set of items, a size, and a budget it is easy to determine what a given voter's favourite outcomes are. We do not do that here as for example if we have $1 \succ_i 2 \succ_i 3 \succ_i 4$ for some voter i and $c(1) = 3, c(2) = c(3) = 2, c(4) = 1$, with size 2 and budget 4, it is unclear if voter i would prefer $\{1, 4\}$ or $\{2, 3\}$, whereas in the uniform cost case their favourite outcome would be $\{1, 2\}$ (not affordable in the non-uniform example).

Proof Let

$$r(i,j) := \begin{cases} m+1 & \text{if } j \in E_i^{k_i} \\ p & \text{if } j \in E_i^p \text{ for some } p \neq k_i. \end{cases}$$

Calculate $s(j) := \sum_{i \in V} (m + 1 - r(i,j))$ for each item $j \in A$. Take B such that $c(B) \leq b$ and $\sum_{j \in B} s(j)$ is maximized. Note that this is an instance of the knapsack problem with utilities in $\mathbb{Z}_{\geq 0}$, and since $s(j) \leq n \cdot m$, we have $u(A) \leq n \cdot m^2$, so this can be done in polynomial time by Theorem 3.1. We claim that such a B is Pareto optimal under the RS extension.

Suppose we have outcomes C and D such that C is a Pareto improvement over D. Let C' and D' be these outcomes once they have been padded to be comparable. There is a bijection $f : C' \to D'$ with $j \succsim_i f(j)$ for all $i \in V$ and all $j \in C'$, and $j \succ_i f(j)$ for some $i \in V$ and some $j \in C'$ (this is the definition of C being a Pareto improvement over D). Define $r(i, m+1) := m+1$ for all $i \in V$, and $s(m+1) = \sum_{i \in V} (m + 1 - r(i, m+1)) = 0$. Note that $r(i,j) \leq r(i, f(j))$ for all $i \in V$ and all $j \in C'$, and $r(i,j) < r(i, f(j))$ for some $i \in V$ and some $j \in C'$. We therefore have

$$\begin{aligned}
\sum_{j \in B} s(j) &= \sum_{j \in B'} s(j) \\
&= \sum_{j \in B'} \sum_{i \in V} (m + 1 - r(i,j)) \\
&> \sum_{j \in C'} \sum_{i \in V} (m + 1 - r(i,j)) \\
&= \sum_{j \in C'} s(j) = \sum_{j \in C} s(j).
\end{aligned}$$

In particular, of all the affordable outcomes, any one for which the sum of the s values is greatest is Pareto optimal, which is how we chose our outcome. ∎

Theorem 6.2 *There exist strategyproof algorithms that return committees that are Pareto optimal under DL and UL set extensions that run in time* $O(n \cdot m \cdot (\min\{b, c(A)\} + 1))$.

Proof We can define additive utility functions $u^{DL}, u^{UL} : V \times \mathcal{P}(A) \to \mathbb{N}$ such that for $B, C \in F$, $u^{DL}(i, B) \geq u^{DL}(i, C) \iff B \succsim_i^{DL} C$ and $u^{UL}(i, B) \geq u^{UL}(i, C) \iff B \succsim_i^{UL} C$ by taking

$$u^{DL}(i,j) = \begin{cases} 0 & \text{if } j \in E_i^{k_i} \\ u^{DL}\left(i, \bigcup_{q=p+1}^{k_i} E_i^q\right) + 1 & \text{if } j \in E_i^p \text{ for } p < k_i \end{cases}$$

and

$$u'(i,j) = \begin{cases} 0 & \text{if } j \in E_i^1 \\ u'\left(i, \bigcup_{q=1}^{p-1} E_i^q\right) - 1 & \text{if } j \in E_i^p \text{ for } p > 1 \end{cases}$$

$$u^{UL}(i,j) = u'(i,j) - \min\{u'(i,j) : j \in A\}$$

Suppose we have two outcomes B and C, and let B' and C' be them padded. Note that $u^{DL}(i, m+1) = u^{UL}(i, m+1) = 0$ for all $i \in V$, so both of these utility functions give the same utility to an outcome as to its padded version. We will give an argument as to why u^{DL} preserves preferences, and then note that a very similar argument holds for u^{UL}.

Note that $u^{DL}(i,j)$ has the same value for all j within the same E_i^p, so if $|B' \cap E_i^p| = |C' \cap E_i^p|$ for all p, then $u^{DL}(i, B') = u^{DL}(i, C')$. If it is not the case that $|B' \cap E_i^p| = |C' \cap E_i^p|$ for all p, then consider the first p on which they differ. If $|B' \cap E_i^p| > |C' \cap E_i^p|$ then

$$\begin{aligned}
u^{DL}(i, B') &= u^{DL}\left(i, B' \cap \bigcup_{q=1}^{p-1} E_i^q\right) + u^{DL}\left(i, B' \cap E_i^p\right) + u^{DL}\left(i, B' \cap \bigcup_{q=p+1}^{k_i} E_i^q\right) \\
&= u^{DL}\left(i, C' \cap \bigcup_{q=1}^{p-1} E_i^q\right) + u^{DL}\left(i, B' \cap E_i^p\right) + u^{DL}\left(i, B' \cap \bigcup_{q=p+1}^{k_i} E_i^q\right) \\
&\geq u^{DL}\left(i, C' \cap \bigcup_{q=1}^{p-1} E_i^q\right) + u^{DL}\left(i, B' \cap E_i^p\right) \\
&\geq u^{DL}\left(i, C' \cap \bigcup_{q=1}^{p-1} E_i^q\right) + u^{DL}\left(i, C' \cap E_i^p\right) + u^{DL}\left(i, \bigcup_{q=p+1}^{k_i} E_i^q\right) + 1 \\
&\geq u^{DL}\left(i, C' \cap \bigcup_{q=1}^{p-1} E_i^q\right) + u^{DL}\left(i, C' \cap E_i^p\right) + u^{DL}\left(i, C' \cap \bigcup_{q=p+1}^{k_i} E_i^q\right) + 1 \\
&> u^{DL}\left(i, C' \cap \bigcup_{q=1}^{p-1} E_i^q\right) + u^{DL}\left(i, C' \cap E_i^p\right) + u^{DL}\left(i, C' \cap \bigcup_{q=p+1}^{k_i} E_i^q\right) \\
&= u^{DL}(i, C'),
\end{aligned}$$

and similarly if $|C' \cap E_i^p| > |B' \cap E_i^p|$ then $u^{DL}(i, C') > u^{DL}(i, B')$. Therefore for

$B, C \in F$, we have $u^{DL}(i,B) \geq u^{DL}(i,C) \iff B \succsim_i^{DL} C$. A similar argument shows that for $B, C \in F$, we have $u^{UL}(i,B) \geq u^{UL}(i,C) \iff B \succsim_i^{UL} C$.

Using the one-dimensional Theorem 3.2 where we let the utility function return the vector $(u(1,B),...,u(n,B))$ for u either u^{DL} or u^{UL} depending on which set extension we care about, and have the order be lexicographical, using the same costs and budget, we can get an outcome in $O\left(n \cdot m \cdot \left(\min\left\{b_i, c_i(A)\right\} + 1\right)\right)$ time that would maximize the lexicographical utility (be chosen by a serial dictatorship) subject to the budget. ∎

Remark Note that if a committee is *DL* or *UL* efficient, then it is also *RS* efficient, so the above algorithm also works for *RS*, though as previously discussed, strategyproofness is not well-defined.

Remark In the dichotomous setting, *RS*, *DL*, and *UL* are equivalent (whichever of $B, C \in F$ has a larger intersection with E_i^1 is preferred).

Theorem 6.3 *For dichotomous preferences, a Pareto improvement (if one exists) over a committee with respect to the responsive set extension (or DL or UL set extension) can be computed in polynomial time when* $tw(\succsim) \leq 2$.

Proof We extend the proof of Theorem 3 given by Aziz and Monnot [5] to our non-uniform cost case.

Suppose the preferences are dichotomous and $tw(\succsim) \leq 2$. Let $B \in F$ be arbitrary.

Note that if $E_i^1 \subseteq B$ for all $i \in V$, then B is *RS*-efficient, so we will assume that there is at least one voter i such that $E_i^1 \backslash B \neq \emptyset$.

We will construct C, a Pareto improvement with respect to the *RS* extension over B if such a Pareto improvement exists.

Let

$$V' = \left\{i \in V : E_i^1 \subseteq B\right\}, W' = \bigcup_{i \in V'} E_i^1 \subseteq B,$$

so V' are all voters who have all their liked items chosen in B, and W' are all the items they like, so if such a C exists, it would have to be the case that $W' \subseteq C$,

$$V'' = \left\{i \in V \setminus V' : E_i^1 \cap (B \setminus W') \neq \emptyset\right\}, A' = \bigcup_{i \in V''} E_i^1,$$

so V'' are all the voters who like exactly one item in $B \setminus W'$ and exactly one item in $A \setminus B$ (the latter as otherwise they would be in V'), meaning if such a C exists, every voter in V'' would have at least one of their liked items in C, and A' is all of these voters' liked items. Note that all voters in $V \setminus V' \setminus V''$ have none of their liked items selected.

Consider now a graph with vertices A' and edges between all $[j_1, j_2]$ such that there is an $i \in V''$ with $E_i^1 = \{j_1, j_2\}$.

We claim that such a C exists iff there is a vertex cover C' such that either $c(C') \leq b - c(W')$ and there is an edge with both vertices covered, or for j the cheapest item in $\left(\bigcup_{i \in V} E_i^1\right) \setminus (W' \cup A')$, it holds that $c(C') \leq b - c(W' \cup \{j\})$.

Suppose there is a vertex cover C' such that $c(C') \leq b - c(W')$ and there is an edge with both vertices covered. We have $c(W' \cup C') \leq b$ so let $C = W' \cup C'$. All voters in V' still have all their liked items selected as $W' \subseteq C$, all voters in V'' still have at least one of the items they like selected as C' contains at least one element of E_i^1 for each $i \in V''$, and at least one voter in V'' has both of the items they like selected (as compared to the one they had before) as C' contains both elements of E_i^1 for some $i \in V''$. Therefore C is a Pareto improvement over B.

Now suppose instead there is a vertex cover C' such that for j the cheapest item in $\left(\bigcup_{i \in V} E_i^1\right) \setminus (W' \cup A')$, it holds that $c(C') \leq b - c(W' \cup \{j\})$. We have $c(W' \cup C' \cup \{j\}) \leq b$ so let $C = W' \cup C' \cup \{j\}$. Note that even without j, all voters in V' still have all their liked items selected as $W' \subseteq C$ and all voters in V'' still have at least one of the items they like selected as C' contains at least one element of E_i^1 for each $i \in V''$, so every voter is no worse off. We then have that since $j \in \left(\bigcup_{i\in V} E_i^1\right) \setminus (W' \cup A')$ any voter i for which $j \in E_i^1$ must be such that $i \in V \setminus V' \setminus V''$, so they have more of the items they like selected (they had none before), so C is a Pareto improvement over B.

Conversely, suppose there is a Pareto improvement C over B. Let $C' = C \cap A'$ (note $C' \cap W' = \emptyset$). As C is a Pareto improvement, all voters in V'' have at least one of their items selected, so C' is a vertex cover. Consider now a voter i who has strictly more of their liked items selected in C as compared to B. If $i \in V''$, then both of their liked items were selected, so there is an edge with both vertices covered by C', and $c(C' \cup W') \leq c(C) \leq b$. If $i \notin V''$ (and clearly $i \notin V'$ as those voters had all of their liked items selected in B), then for any $j \in E_i^1 \cap C$, we have that $j \notin W' \cup A'$, so since $W' \cup C' \cup \{j\} \subseteq C$, there is an item $j \in \left(\bigcup_{i\in V} E_i^1\right) \setminus (W' \cup A')$ with $c(C' \cup W' \cup \{j\}) \leq c(C) \leq b$, so $c(C') \leq b - c(W' \cup \{j\})$ (so in particular it would also hold for the cheapest such j).

Therefore our claim holds. Now it just remains to show that these graph theory problems can be solved in polynomial time.

Plummer and Lovász [14] state that finding a vertex cover in a bipartite graph with a bounded cost can be done in polynomial time.

To see if there is a vertex cover C' such that $c(C') \leq b - c(W')$ and there is an edge with both vertices covered, we can check for each edge $[j_1, j_2]$ in the graph (of which there are at most $\binom{m}{2}$) if there is a vertex cover C'' in the subgraph induced by $A' \setminus \{j_1, j_2\}$ such that $c(C'') \leq b - c(W' \cup \{j_1, j_2\})$.

We can also check if there is a vertex cover C' such that for j the cheapest item in $\left(\bigcup_{i\in V} E_i^1\right) \setminus (W' \cup A')$, it holds that $c(C') \leq b - c(W' \cup \{j\})$. ∎

Theorem 6.4 *For dichotomous preferences, a Pareto improvement (if one exists) over a committee with respect to $\mathcal{B}$ can be computed in $O(n + k)$ when $tw(\succsim) = 1$.*

Intuitively, in this setup every voter likes exactly 1 item, so they are happy iff their item is selected.

Proof Suppose we wish to check the $\mathcal{B}$-efficiency of an outcome B.

Letting $B' = B \cap \bigcup_{i \in V} E_i^1$, if there is any item $j \in \bigcup_{i \in V} E_i^1 \setminus B$ with cost no more than $b - c(B')$, then $C = B' \cup \{j\}$ forms a Pareto improvement as all the voters that were happy are still happy, and another voter who was not happy is now happy. If there are no such affordable items, then there is no Pareto improvement as to make some voter happier we would have to remove an item that another voter likes.

B' and $\bigcup_{i \in V} E_i^1 \setminus B$ can be constructed in $O(n)$ time, computing the remaining budget takes $O(k)$ time, and checking $\bigcup_{i \in V} E_i^1 \setminus B$ for affordable items takes $O(n)$ time. $\blacksquare$

Theorem 6.5 *Under linear preferences, there is a strategyproof algorithm to compute a $\mathcal{B}$-efficient committee that runs in $O(nm)$ time.*

Proof Starting with $\emptyset$, have each voter in turn add their favourite affordable item (the cost of a previously added item is 0) and subtract the cost from the remaining budget until all the voters have had a turn.

A voter finding their favourite affordable item take $O(m)$ time, and $O(n)$ voters do this. $\blacksquare$

Theorem 6.6 *There exists a polynomial-time algorithm that checks whether an outcome is $\mathcal{W}$-efficient and computes a Pareto improvement over it if possible.*

With $\mathcal{W}$, for an outcome B and $k = \max\{|C| : C \in F\}$, a voter is as unhappy as their least favourite item selected in B', where B' is B padded to size k. Note that if $|B| < k$ then every voter will be as unhappy as possible.

Proof Let $k = \max\{|C| : C \in F\}$.

Given an outcome B, if $|B| < k$, let $p_i = k_i$ for each voter. Otherwise, let p_i be the largest index such that $B \cap E_i^{p_i} \neq \emptyset$ for each voter i. Let

$$S = \bigcap_{i \in V} \left(\bigcup_{q=1}^{p_i} E_i^q \right) \setminus \{m+1\}.$$

S represents the set of all items that can be included in an outcome of size k while not making any voter less happy than they were in B, so any Pareto improvement C on B would have to be such that $C \subseteq S$ and $|C| = k$.

For each voter i, check if the k cheapest items in $S \setminus E_i^{p_i}$ are together affordable. If so for any voter, they form a Pareto improvement as that voter will be happier than with B, and no voter will be less happy than with B. If not, then there is no Pareto-improvement, as if there were a Pareto improvement C over B, then for i the voter who was strictly happier with C than B, we would have $C \subseteq S \setminus E_i^{p_i}$. ∎

Theorem 6.7 *There exists a polynomial-time and strategyproof algorithm that returns a $\mathcal{W}$-efficient committee.*

Proof Let $k = \max\{|C| : C \in F\}$.

Let $B_0 = A$.

Let p_1 be the least integer such that the k cheapest items in $B_0 \cap \bigcup_{q=1}^{p_1} E_1^q$ are together affordable.

Let $B_1 = B_0 \cap \bigcup_{q=1}^{p_1} E_1^q$.

Let p_2 be the least integer such that the k cheapest items in $B_1 \cap \bigcup_{q=1}^{p_2} E_2^q$ are together affordable.

Let $B_2 = B_1 \cap \bigcup_{q=1}^{p_2} E_2^q$.

Continue in this manner until we have B_n, then return the k cheapest items in B_n.

Note that this is a serial dictatorship, so it is Pareto optimal and strategyproof. ∎

7 Explainability

7.1 Introduction

In this section we investigate the Sequential Phragmén and Equal Shares budgeting methods. We chose these budgeting methods as they are fairly straightforward and have some desirable fairness guarantees, as we will describe later. They are also both generalizations on previous rules, so the results we find here are more widely applicable.

We look at how variations in the price of an item or the total budget affect whether that item is selected, to complement the search for ways to make budgeting methods more understandable. By doing this, we can inform proposers whose projects were not chosen how much cheaper it would have needed to be to get chosen or how much more overall budget they should have lobbied for to get their project funding, and we can inform proposers whose projects *were* chosen how much more expensive their project could have been while still getting chosen or how safe they were with respect to the overall budget being cut.

7.2 Sequential Phragmén

Given our standard participatory budgeting setup but with costs and budget in $\mathbb{R}_{\geq 0}$ and utilities in $\{0, 1\}$ (represented by approval sets), Sequential Phragmén is an algorithm which returns an outcome. The first version of this rule was given by Phragmém [13] for Approval-Based Multi-Agent scenarios. Brill et al. [7] then showed that Sequential Phragmén satisfies proportional justified representation (and so also justified representation), though unfortunately not extended justified representation. Aziz et al. [4] then extended Sequential Phragmén to the non-uniform cost case, but while doing so made the algorithm less straightforward. Finally, Peters et al. [11] gave an extension to non-uniform costs which was more in line with the original algorithm, and that is the one we will be considering.

Sequential Phragmén We start with an empty outcome and our total budget. We also give each voter i a personal budget which starts with \$0, then increases continuously by \$1 every second. As soon as an item j is affordable with the combined budgets of all voters who approve of j, j is added to the outcome, $c(j)$ is subtracted from the total budget, and the voters who approve of j have their budget reset to \$0 (in this paper, we by default break ties based on the natural ordering of $\{1, 2, ..., m\}$, where 1 wins any tie it is in). This continues until the item to be added has higher cost than the remaining total budget, or all the items that are not unanimously disapproved of have been added.

While the algorithms we use to implement Sequential Phragmén achieve the described outcome, clearly they do not actually implement a continuous process as described. Note though that it is easy to see how much time it would take a particular item to have enough funds, given its supporters' current budgets and total income, so it is also easy to figure out which item would be funded next and in how many seconds, and then each voter's budget can be updated accordingly.

Remark A common variant on this rule would be to change the termination condition to be "This continues until all the remaining items that are not unanimously disapproved of have higher cost than the remaining total budget."

7.2.1 Finding affordable pricing

To find at what costs item 1 will be selected by Sequential Phragmén given all other pieces of a scenario, we actually go about answering the somewhat more detailed question of "For which costs will it be selected first? second? ... mth?"

Note that if $c(1) = 0$, then it will be selected first.

Otherwise, if $u(i, 1) = 0$ for all i, then it will not be selected, regardless of its cost.

Otherwise, we find the item $j \in A \setminus \{1\}$ which is affordable by its supporters after $t \in \mathbb{R}_{\geq 0}$ seconds for the least t, and calculate at what cost c' would item 1 would be affordable after t seconds (if no such j exists, have c' be the remaining total budget). For all costs in $[0, c']$ (this interval is closed due to our tie-breaking rule), item 1 will be selected first. If item 1 is not selected first, item j will be selected first, and we can remove j from consideration and update the budgets.

To find at what costs item 1 will be selected second is now (with j removed and the budgets modified) an identical problem to finding at what costs it would have been selected first (though we will need to remove the costs at which it would have been selected first from the resulting closed interval). The previous paragraph can therefore be repeated until item 1 is no longer affordable by the total budget or we have finished an iteration where the item that would have been chosen was not affordable by the total budget.

The union of all the intervals gives the costs at which item 1 will be selected.

Remark This process can be done efficiently.

Example We give an example where item 1 can be selected first or third depending on its cost, but never second (that is not reliant on our tie-breaking).

$A = [3]$, $c(2) = 2, c(3) = 3$, $b = 7$, $V = [2]$, $A_1 = \{1, 2\}, A_2 = \{3\}$.

Note that item 2 would be affordable by voter 1 after 2 s and item 3 would be affordable by voter 2 after 3 s. Therefore item 1 will be selected first iff $c(1) \in [0, 2]$.

If $c(1) > 2$ though, then item 2 will be selected first. At this point it will take at least 2 s for item 1 to be affordable by voter 1, but it will only take 1 s for item 3 to be selected, so item 3 will be selected second.

Remark Note that the range of costs for which item 1 will be selected is of the form $[0, c']$, and more precisely, this interval can be split into intervals

$[c_{i_0}, c_{i_1}] \cup (c_{i_1}, c_{i_2}] \cup \cdots \cup (c_{i_{k-1}}, c_{i_k}]$ for $c_{i_0} = 0$, $c_{i_k} = c'$, $i_1 < \cdots < i_k$, and $i_1 = 1$, where the interval from $c_{i_{k'-1}}$ to $c_{i_{k'}}$ are the costs for which it will be selected $i_{k'}$th.

If we consider this procedure for items j other than 1, we might get slight variations in the formatting of the result. One such change is that i_1 may not be 1. This will happen if there is an item with a lower-indexed item with cost 0. Another such change is that some of the right ends of the intervals may change from closed to open (and the following left end if it exists would then change from open to closed).

Remark The process by which we find costs at which an item will be purchased does not change significantly if we use the alternate termination condition.

7.2.2 Finding good budgets

Note that the total budget is only taken into account in the termination condition. In particular, running Sequential Phragmén with infinite budget but then taking the last constructed outcome that is affordable by the total budget, is equivalent to running Sequential Phragmén normally. We can therefore easily see what the minimum required budget b' is for an item j to be selected by running Sequential Phragmén with infinite budget and finding how much of the total budget is used immediately after j is added to the outcome. We then have that item j will be selected iff $b \in [b', \infty)$.

Remark The above given process by which we find budgets at which an item will be purchased does not work if we use the alternate termination condition, as shown by the following example. In the case where there is only one voter though, this becomes equivalent to Equal Shares with one voter and utilities in $\{0, 1\}$, for which we have results later (without the assumption on the utilities).

Example Let $A = [2]$, $c(1) = 3, c(2) = 2$, $V = [2]$, $A_1 = \{1, 2\}, A_2 = \{1\}$. Using the alternate termination condition, note that if $b = 2$, the outcome will be $\{2\}$, but if $b = 3$ the outcome will be $\{1\}$, so the budgets for which item 2 is selected are not of the form $[b', \infty)$.

7.3 Equal Shares

Peters et al. [11] extend the previously described Equal Shares method to our standard participatory budgeting setup, but with costs in $\mathbb{Q}^+$, a budget of 1, and utilities in $[0, 1]$.

Equal Shares Each voter i is given $b_i = \frac{b}{n}$ dollars.

We say that an item j is ρ-affordable for $\rho \geq 0$ if ρ is the least value such that

$$\sum_{i \in V} \min \{b_i, u(i,j) \cdot \rho\} = c(j),$$

and we say it is affordable if it is ρ-affordable for any ρ, or equivalently

$$\sum_{i \in V} b_i \cdot \mathbb{1}\left[u(i,j) > 0\right] \geq c(j).$$

Starting with an empty outcome $B = \emptyset$, we take an item j which is ρ-affordable for the least ρ (in this paper we by default break ties based on the natural ordering of $\{1, 2, ..., m\}$, where 1 wins any tie it is in), add j to B, and subtract $\min \{b_i, u(i,j) \cdot \rho\}$ from each b_i (ρ can be seen as the ratio dollars spent/utility gained). We repeat this until no item is ρ-affordable for any ρ, then terminate.

Note that finding a ρ for which an item j is ρ-affordable (or whether no such ρ exists) is solvable in $O(n)$ time as for fixed $b_1, ..., b_n$,

$$f_j(\rho) = \sum_{i \in V} \min \{b_i, u(i,j) \cdot \rho\}$$

is a continuous non-decreasing piecewise linear function with breakpoints exactly at

$$\left\{ \frac{b_i}{u(i,j)} : i \in V, u(i,j) > 0 \right\}.$$

Peters et al. [11] show that this extension of Equal Shares satisfies a few nice properties related to the following definitions, some of which are extensions on previous definitions which were in more restrictive settings.

Definition We say that a group of voters V' is (α, C)-cohesive for $\alpha : A \to [0, 1]$ and $C \subseteq A$, if $|V'| \geq c(C) \cdot n$ and it holds that $u(i, j) \geq \alpha(j)$ for all voters $i \in V'$ and items $j \in C$.

Using this definition, they extend the definition of EJR to non-uniform costs and utilities.

Definition A rule satisfies extended justified representation (up to one item) if for all scenarios A, c, b, V, u and resulting bundles B, for all $\alpha : A \to [0, 1]$, for all $C \subseteq A$, for each (α, C)-cohesive group of voters V', there exists a voter $i \in V'$ such that $u(i, B) \geq \alpha(C)$ (where $\alpha(C)$ is shorthand for $\sum_{j \in C} \alpha(j)$) or for some $j \in A$ it holds that $u(i, B \cup \{j\}) > \alpha(C)$.

Intuitively, a group of voters V' is (α, C)-cohesive means they all agree that each item $j \in C$ is worth at least $\alpha(j)$, and satisfying EJR up to one project means that such a cohesive group should have at least one of their voters have as much utility as α (the group consensus on each of the items) if they pooled their funds to get C, allowing a wiggle room of one item.

Remark Peters et al. [11] show that this extension of Equal Shares satisfies EJR up to one item.

Definition We say that an outcome B is in the core if for all $V' \subseteq V$ and $C \subseteq A$ with $|V'| \geq c(C) \cdot n$, there exists $i \in V'$ such that $u(i, B) \geq u(i, C)$. An election rule satisfies the core property if it always returns an outcome in the core.

Intuitively, if the outcome is in the core, no group of voters V' is entirely motivated to take their share of funds and purchase their own bundle.

Unfortunately, Peters et al. [11] give the following example where no outcomes are in the core, even with uniform costs.

Example $A = [6], c(j) = \frac{1}{3}$ for all $j \in A, V = [6]$.

$$u(1,1) > u(1,2) > 0, \quad u(2,2) > u(2,3) > 0, \quad u(3,3) > u(3,1) > 0,$$
$$u(4,4) > u(4,5) > 0, \quad u(5,5) > u(5,6) > 0, \quad u(6,6) > u(6,4) > 0,$$

and all other utilities 0. Note that any feasible outcome B has $|B \cap \{1,2,3\}| \leq 1$ or $|B \cap \{4,5,6\}| \leq 1$. Without loss of generality, suppose $|B \cap \{1,2,3\}| \leq 1$, and again without loss of generality, suppose $B \cap \{1,2,3\} = \{1\}$. Voters $\{2,3\}$ with bundle $\{3\}$ show that B is not in the core, and for any other feasible outcome we have a symmetric argument.

Definition For $a \geq 1$, Peters et al. [11] say that an outcome B is in the <u>a-core</u> if for all $V' \subseteq V$ and $C \subseteq A$ with $|V'| \geq c(C) \cdot n$, there exists $i \in V'$ and $j \in C$ such that $u(i, B \cup j) \geq \frac{u(i,C)}{a}$. An election rule satisfies the <u>core property</u> if it always returns an outcome in the core.

Remark Peters et al. [11] show that for

$$u_{\max} := \max_{i \in V} \max_{B \in F} u(i,B) \text{ and } u_{\min} := \min_{i \in V} \min_{B \in F : u(i,B) > 0} u(i,B)$$

the highest total utility a voter can get and the lowest positive utility a voter can get, respectively, Equal shares satisfies the a-core property for $a = 4 \log\left(2 \cdot \frac{u_{\max}}{u_{\min}}\right)$.

Peters et al. [11] then give the following definition from Peters and Skowron [12].

Definition A <u>price system</u> is a pair $(b, \{p_i\}_{i \in V})$, where $b \geq 1$ is the initial budget (with each voter controlling an equal share of the budget), and for each voter $i \in V$, the <u>payment function</u> $p_i : A \to \mathbb{R}_{\geq 0}$ [4] specifies the amount of money voter i pays for each item. An outcome B is <u>supported</u> by a price system $(b, \{p_i\}_{i \in V})$ if

[4]Peters et al. [11] miss the condition that non-negative payments are not allowed.

- voters do not pay for items they do not like: $u(i, j) = 0 \implies p_i(j) = 0$ for all $i \in V$ and $j \in A$,
- each voter gets a fair share of the budget: $\sum_{j \in A} p_i(j) \leq \frac{b}{n}$ for all $i \in V$,
- each selected item is paid for: $\sum_{i \in V} p_i(j) = c(j)$ for all $j \in B$,
- each unselected item is not paid for: $\sum_{i \in V} p_i(j) = 0$ for all $j \notin B$,
- and each unselected item can no longer be paid for by its supporters: $\sum_{i \in V : u(i,j) > 0} \left(\frac{b}{n} - \sum_{j \in B} p_i(j) \right) < c(j)$ for all $j \notin B$.

An outcome B is priceable if it is supported by some price system.

Remark Peters et al. [11] note that any rule that, like Equal Shares, equally splits the budget between the voters, then purchases items using the funds of supporters, will be priceable.

For these reasons Equal Shares seems like a nice rule to study.

7.3.1 Finding affordable pricing

To find at what costs item 1 will be selected by Equal Shares given all other pieces of a scenario, we actually go about answering the somewhat more detailed question of "For which costs will it be selected first? second? ... mth?" It is a similar procedure to the one for Sequential Phragmén, but we present it still for a few important details and completeness.

Note that if $c(1) = 0$, then it will be selected first.

Otherwise, if $u(i, 1) = 0$ for all i with remaining budget, then it will not be selected going forward, regardless of its cost. Otherwise, we find the item $j \in A \setminus \{1\}$ which is ρ-affordable for the least ρ, and calculate at what cost c' item 1 would be ρ-affordable (if no such j exists, have c' be the maximum cost such that item 1 is affordable). For all costs in $[0, c']$ (closed due to our tie-breaking rule), item 1 will be selected first. If item 1 is not selected first, item j will be selected first, and we can remove j from consideration and subtract the corresponding amounts from the voters' budgets.

To find at what costs item 1 will be selected second is now (with j removed and the budgets modified) an identical problem to finding at what costs it would have been selected first (though we will need to remove the costs at which it would have been selected first from the resulting closed interval). The previous paragraph can therefore be repeated until an iteration where no other item is affordable has been done or the maximum cost at which 1 is affordable is within one of the previously calculated intervals (note the maximum cost at which 1 is affordable is non-increasing during a run of the algorithm).

The union of all the intervals gives the costs at which item 1 will be selected.

Remark For fixed $b_1, ..., b_n$, as

$$f_j(\rho) = \sum_{i \in V} \min \{b_i, u(i,j) \cdot \rho\}$$

is a continuous non-decreasing piecewise linear function with breakpoints exactly at

$$P = \left\{ \frac{b_i}{u(i,j)} : i \in V, u(i,j) > 0 \right\},$$

and it is strictly increasing on $[0, \max(P)]$, $f_j^{-1}(q)$ is a well-defined continuous strictly increasing piecewise linear function on

$$\left[0, \sum_{i \in V} b_i \cdot \mathbb{1}\left[u(i,j) > 0\right] \right]$$

with breakpoints exactly at

$$\left\{ f_j\left(\frac{b_i}{u(i,j)} \right) : i \in V, u(i,j) > 0 \right\},$$

so it is efficiently computable.

Remark The process for finding at which costs item 1 will be selected can be done efficiently.

Example We give an example where item 1 can be selected first or third depending on its price, but never second (that is not reliant on our tie-breaking).

$A = [3]$, $c(2) = c(3) = 1$, $b = 3$, $V = [3]$,
$u(1, 1) = u(2, 1) = 1, u(2, 2) = 3, u(3, 3) = 2$, and all other utilities are 0.

Note that initially item 2 is $\frac{1}{3}$-affordable and item 3 is $\frac{1}{2}$-affordable. Therefore item 1 will be selected first iff $c(1) \in \left[0, \frac{2}{3}\right]$.

If $c(1) > \frac{2}{3}$ though, then item 2 will be selected first, paid for exactly by all of voter 2's budget. At this point item 1 is $c(1)$-affordable, so item 3 will be selected second, with item 1 being selected third if $\frac{2}{3} < c(1) \leq 1$.

Remark Note that the range of costs for which item 1 will be selected is of the form $[0, c']$, and more precisely, this interval can be split into intervals
$[c_{i_0}, c_{i_1}] \cup (c_{i_1}, c_{i_2}] \cup \cdots \cup (c_{i_{k-1}}, c_{i_k}]$ for $c_{i_0} = 0$, $c_{i_k} = c'$, $i_1 < \cdots < i_k$, and $i_1 = 1$, where the interval from $c_{i_{k'-1}}$ to $c_{i_{k'}}$ are the costs for which it will be selected $i_{k'}$th.

If we consider this procedure for items j other than 1, we might get slight variations in the formatting of the result. One such change is that i_1 may not be 1. This will happen if there is an item with a lower number with cost 0. Another such change is that some of the right ends of the intervals may change from closed to open (and the following left end if it exists would then change from open to closed).

7.3.2 Finding good budgets

We consider the case where there is one voter. When there is one voter we have that an affordable item j is ρ-affordable for $\rho = \frac{c(j)}{u(1,j)}$. In particular the ρ does not change throughout the Equal Shares algorithm unless it becomes unaffordable. Therefore running the Equal Shares algorithm with one voter is equivalent to sorting the items into a list by their ρ value (breaking ties however Equal Shares does) and greedily taking affordable items.

Proposition 7.1 *Suppose we have a list of items $j_1, ..., j_m$ and an algorithm which starting with a budget $b_0 = b$ and an outcome $B_0 = \emptyset$, iterates through the list and after reaching item j_k sets $b_k = b_{k-1} - c(j_k)$ and $B_k = B_{k-1} \cup j_k$ if $c(j_k) \leq b_{k-1}$, and $b_k = b_{k-1}$ and $B_k = B_{k-1}$ otherwise.*

Let $f : \mathbb{R}_{\geq 0} \times (\{0\} \cup [m]) \to \mathbb{R}_{\geq 0}$ be such that $f(b, k)$ is the value of b_k if this algorithm is run with budget b.
Let $g : \mathbb{R}_{\geq 0} \times (\{0\} \cup [m]) \to \mathbb{R}_{\geq 0}$ be such that $g(b', k) = \min\{b : f(b, k) \geq b'\}$.

We claim that for all $k \in \{0\} \cup [m]$ and $b' \in \mathbb{R}_{\geq 0}$, $g(b', k)$ exists,
for all $k \in [m]$ and $b' \in \mathbb{R}_{\geq 0}$

$$g(b', k) = \begin{cases} g(b' + c(j_k), k-1) & \text{if } c(j_k) \leq b', \\ g(b', k-1) & \text{else,} \end{cases}$$

and that $f(g(b', k), k) = b'$.

Proof We use a proof by induction on k to show that the $g(b', k)$ exist and that $f(g(b', k), k) = b'$, and in doing so prove the recurrence on g.

Let $k \geq 0$ and $b' \in \mathbb{R}_{\geq 0}$ be arbitrary.

Base case

If $k = 0$, note that $f(b, 0) = b$, so $g(b', 0) = b'$, and $f(g(b', 0), 0) = b'$.

Inductive hypothesis

For all $0 \leq k' < k$ and $b' \in \mathbb{R}_{\geq 0}$,
$g(b', k')$ exists and $f(g(b', k'), k') = b'$.

Inductive step

We now consider what happens when $k \geq 1$.

Case 1 ($c(j_k) \leq b'$)

Let $b \in \{b : f(b,k) \geq b'\}$ be arbitrary. Note that this set is non-empty as $b' + \sum_{j \in [k]} c(j)$ is in it as

$$f(b,k) \geq f(b,k-1) - c(j_k) \geq f(b,k-2) - c(j_{k-1}) - c(j_k) \geq \cdots \geq f(b,0) - \sum_{j \in [k]} c(j).$$

We have

$$f(b,k-1) \geq f(b,k) \geq b' \geq c(j_k),$$

so $f(b,k) = f(b,k-1) - c(j_k)$. Therefore $b \in \{b : f(b,k-1) \geq b' + c(j_k)\}$.

Hence $\{b : f(b,k) \geq b'\} \subseteq \{b : f(b,k-1) \geq b' + c(j_k)\}$, so (using the inductive hypothesis for the existence of $g(b' + c(j_k), k-1)$)

$$\inf \{b : f(b,k) \geq b'\} \geq g(b' + c(j_k), k-1).$$

As

$$f(g(b' + c(j_k), k-1), k-1) \geq b' + c(j_k) \geq c(j_k)$$

(by definition of g), we have that

$$f(g(b' + c(j_k), k-1), k) = f(g(b' + c(j_k), k-1), k-1) - c(j_k) \geq b',$$

so $g(b' + c(j_k), k-1) \in \{b : f(b,k) \geq b'\}$.

Therefore

$$\inf \{b : f(b,k) \geq b'\} = g(b' + c(j_k), k-1) = \min \{b : f(b,k) \geq b'\},$$

so $g(b', k)$ exists and

$$g(b', k) = g(b' + c(j_k), k - 1).$$

Finally, we note that (using our inductive hypothesis that $f(g(b'', k-1), k-1) = b''$)

$$\begin{aligned} f\left(g(b', k), k\right) &= f\left(g(b' + c(j_k), k - 1), k\right) \\ &= f\left(g(b' + c(j_k), k - 1), k - 1\right) - c(j_k) \\ &= b' + c(j_k) - c(j_k) \\ &= b'. \end{aligned}$$

Case 2 ($c(j_k) > b'$)

Let $b \in \{b : f(b, k) \geq b'\}$ be arbitrary. Note that this set is non-empty as shown previously.

We have

$$f(b, k - 1) \geq f(b, k) \geq b',$$

so $b \in \{b : f(b, k - 1) \geq b'\}$.

Therefore $\{b : f(b, k) \geq b'\} \subseteq \{b : f(b, k - 1) \geq b'\}$, so (using the inductive hypothesis for the existence of $g(b', k - 1)$)

$$\inf \{b : f(b, k) \geq b'\} \geq g(b', k - 1).$$

By our inductive hypothesis, $f(g(b', k - 1), k - 1) = b'$, so since $f(g(b', k - 1), k - 1) < c(j_k)$, we have that $f(g(b', k - 1), k) = f(g(b', k - 1), k - 1)$, so $g(b', k - 1) \in \{b : f(b, k) \geq b'\}$.

Therefore

$$\inf \{b : f(b, k) \geq b'\} = g(b', k - 1) = \min \{b : f(b, k) \geq b'\},$$

so $g(b', k)$ exists and

$$g(b', k) = g(b', k-1).$$

Finally, we note that (using the inductive hypothesis in the second to last step)

$$\begin{aligned}
f\left(g(b', k), k\right) &= f\left(g(b', k-1), k\right) \\
&= \begin{cases} f\left(g(b', k-1), k-1\right) - c(j_k) & \text{if } c(j_k) \leq f\left(g(b', k-1), k-1\right), \\ f\left(g(b', k-1), k-1\right) & \text{else,} \end{cases} \\
&= \begin{cases} b' - c(j_k) & \text{if } c(j_k) \leq b', \\ b' & \text{else,} \end{cases} \\
&= b'.
\end{aligned}$$

End of induction

Therefore by induction the $g(b', k)$ exist and $f\left(g(b', k), k\right) = b'$. Along the way, we also showed that for $k \geq 1$,

$$g(b', k) = \begin{cases} g(b' + c(j_k), k-1) & \text{if } c(j_k) \leq b', \\ g(b', k-1) & \text{else,} \end{cases}$$

∎

Remark Note that using this recurrent definition of g we can calculate $g(b', k)$ in time linear in k. Also note that item j_k is bought iff $f(b, k-1) \geq c(j_k)$, so in particular, $g(c(j_k), k-1)$ is the least budget such that item j_k is bought.

Example We give an example where each item can be selected first depending on the budget.

$A = [m]$, $c(j) = 1$ for all $j \in A$, $V = [m]$, $u(i, j) = \mathbb{1}\left[i \leq j\right] \cdot \frac{m-j+1}{j}$.

Consider before the first item is bought, when $b_i = \frac{b}{m}$ for all i. Note that item j is affordable iff $b \geq \frac{m}{j}$ as j voters give it non-zero utility.

If $b \geq \frac{m}{j}$ though, note that

$$\begin{aligned}
&\min\left\{\rho : \sum_{i \in V} \min\left\{b_i, u(i,j) \cdot \rho\right\} = c(j)\right\} \\
&= \min\left\{\rho : \sum_{i \in V} \min\left\{\frac{b}{m}, \mathbb{1}\left[i \leq j\right] \cdot \frac{m-j+1}{j} \cdot \rho\right\} = 1\right\} \\
&= \min\left\{\rho : \sum_{i \in [j]} \min\left\{\frac{b}{m}, \frac{m-j+1}{j} \cdot \rho\right\} = 1\right\} \\
&= \min\left\{\rho : \min\left\{\frac{bj}{m}, (m-j+1) \cdot \rho\right\} = 1\right\} \\
&= \min\left\{\rho : (m-j+1) \cdot \rho = 1\right\} \\
&= \min\left\{\rho : \rho = \frac{1}{m-j+1}\right\} \\
&= \frac{1}{m-j+1},
\end{aligned}$$

so it is $\frac{1}{m-j+1}$-affordable.

Therefore the least j that is affordable will be the first one selected, so for all $j \in A \setminus \{1\}$, item j will be selected first iff $b \in \left[\frac{m}{j}, \frac{m}{j-1}\right)$, and item 1 will be selected first iff $b \geq m$.

Example We give an example with one voter where varying the budget has some potentially undesirable properties. In particular, all outcomes in $\mathcal{P}(A)$ can be realized and the budgets for which item 1 is selected form 2^{m-1} connected components (not relying on tie-breaking).

$A = [m]$, $c(j) = 2^{j-1}$, $V = [1]$, $u(1,j) = 2^{2j-1}$.

Note when there is one voter we have that an affordable item j is ρ-affordable for $\rho = \frac{c(j)}{u(1,j)}$, and in this scenario we have $\frac{c(j)}{u(1,j)} = 2^{-j}$. In particular the largest j such that item j is affordable will be bought first. Therefore if $b \in [0, 2^m - 1] \cap \mathbb{Z}$ and we write $b = \sum_{j=1}^{m} d_j \cdot 2^{j-1}$ for $d_j \in \{0,1\}$ (b in binary), we will have that item j is bought iff $d_j = 1$.

Therefore for any $B \subseteq A$, a budget of $b = \sum_{j=1}^{m} \mathbb{1}[j \in B] \cdot 2^{j-1}$ will yield outcome B.

Note that for $b \in [0, 2^m - 1] \backslash \mathbb{Z}$, at every point of the Equal Shares algorithm all the values of ρ will be the same as they would have been for budget $\lfloor b \rfloor$, and for $b \geq 2^m - 1$, at every point of the Equal Shares algorithm all the values of ρ will be the same as they would have been for budget $2^m - 1$, as the affordability of items will not change.

Therefore item 1 will be in the outcome for all budgets in

$$\bigcup_{k=1}^{2^{m-1}-1} [2k-1, 2k) \cup [2^m - 1, \infty).$$

Remark The previous example shows we cannot always list all the intervals of the budgets for which an item is affordable in polynomial time.

Remark The process by which we found the minimum budget at which an item is purchased when there is one voter unfortunately does not easily extend to even two voters, even with utilities in $\{0,1\}$. Key differences are that in the single voter case, every item is either not affordable or has a fixed ρ, value, so they can be sorted by ρ value. With 2 or more voters, the ρ values can change over the course of the algorithm. Also, when working backwards, as we did with the g function, the case in which an item was purchased does not always leave just one possible value for the remaining budget before the purchase, as it can potentially have an uncountable set of values. For example, if there are 2 voters each of which give utility 1 to an item which costs 1, and both voters end with \$0, they could potentially have started with any budgets in $\{(b_1, 1 - b_1) : b_1 \in [0,1]\}$.

This combined with our remarks about how many different first purchases there can be, how many different final outcomes, and how frequently the outcome can change for small variations in the budget, makes it seem like a more comprehensive analysis of the behaviour for varying budgets would be difficult.

8 Future work

8.1 Stability for mixed utilities

In section 5 we considered notions of stability when voters had negative preferences, but unfortunately none of the results were particularly satisfying as they either did not take sufficient advantage of the information provided by the voters, or because they resulted in inefficient outcomes. One potential avenue for further research is to generalize the notion of the core for when there are both positive and negative utilities.

For example, we could consider this definition in the case where we wish to select a committee of k people.

Definition An outcome B is in the core if for every subset of voters $V' \subseteq V$, for $\ell = \lfloor |V'| \cdot \frac{k}{n} \rfloor$ and $\ell' = \lfloor |V - V'| \cdot \frac{k}{n} \rfloor$, then for all $S \subseteq A$ with $|S| \leq \ell$, there exists a voter $i \in V'$ such that there exists $R \subseteq A - S$ with $|R| \leq \ell'$ such that

$$u(i, B) \geq u(i, S \cup R).$$

This notion of the core is a generalization of previous notions of the core, which is already a very demanding notion as mentioned earlier, but analogous notions of fairness such and justified representation, proportional justified representation, and extended justified representation could be defined and investigated.

8.2 Explainability

In section 7, we could not find an efficient way to find a minimum budget at which an item would be selected in Equal Shares, but on top of that, these ideas could be extended to other aggregation rules to help project proposers better understand the results.

9 Personal reflections

After having spent a significant quantity of time on the original idea for the dissertation without making satisfactory progress, we ended up pivoting to other topics, of which we ended up working on several. I believe changing topics was the right decision to make, as knowing when to cut your losses can help prevent getting stuck in a rut. It was interesting going over the vast amount of literature on the subject and seeing just how many different approaches there are, and I enjoyed working on a variety of different topics.

In terms of the relevant coursework leading up to this project, certainly the cooperative section of Computational Game Theory was the most relevant, but my experience prior to the program was also key.

References

[1] Rumen Andonov, Vincent Poirriez, and Sanjay Rajopadhye. Unbounded knapsack problem: Dynamic programming revisited. European Journal of Operational Research, 123(2):394–407, 2000.

[2] Haris Aziz, Markus Brill, Vincent Conitzer, Edith Elkind, Rupert Freeman, and Toby Walsh. Justified representation in approval-based committee voting. Social Choice and Welfare, 48(2):461–485, 2017.

[3] Haris Aziz, Edith Elkind, Shenwei Huang, Martin Lackner, Luis Sánchez-Fernández, and Piotr Skowron. On the complexity of extended and proportional justified representation. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018.

[4] Haris Aziz, Barton Lee, and Nimrod Talmon. Proportionally representative participatory budgeting: Axioms and algorithms. arXiv preprint arXiv:1711.08226, 2017.

[5] Haris Aziz and Jérôme Monnot. Computing and testing pareto optimal committees. Autonomous Agents and Multi-Agent Systems, 34(1):1–20, 2020.

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

[7] Markus Brill, Rupert Freeman, Svante Janson, and Martin Lackner. Phragmén's voting methods and justified representation. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 31, 2017.

[8] Michael R. Garey and David S. Johnson. Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., USA, 1990.

[9] Richard M Karp. Reducibility among combinatorial problems. In Complexity of computer computations, pages 85–103. Springer, 1972.

[10] Ariel Kulik and Hadas Shachnai. There is no EPTAS for two-dimensional knapsack. Information Processing Letters, 110(16):707–710, 2010.

[11] Dominik Peters, Grzegorz Pierczynski, and Piotr Skowron. Proportional participatory budgeting with cardinal utilities. CoRR, abs/2008.13276, 2020.

[12] Dominik Peters and Piotr Skowron. Proportionality and the limits of welfarism.

In Proceedings of the 21st ACM Conference on Economics and Computation, pages 793–794, 2020.

[13] Edvard Phragmén. Sur une méthode nouvelle pour réaliser, dans les élections, la représentation proportionnelle des partis. 1894.

[14] Michael D Plummer and László Lovász. Matching theory. Elsevier, 1986.

[15] Luis Sánchez-Fernández, Edith Elkind, Martin Lackner, Norberto Fernández, Jesús Fisteus, Pablo Basanta Val, and Piotr Skowron. Proportional justified representation. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 31, 2017.

[16] Nimrod Talmon and Piotr Faliszewski. A framework for approval-based budgeting methods. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 2181–2188, 2019.

[17] Nimrod Talmon and Rutvik Page. Proportionality in committee selection with negative feelings, 2021.

[18] Vijay V. Vazirani. Knapsack. In Approximation Algorithms, pages 68–73, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

A Code for computing a Pareto optimal outcome when there is a constant number of distinct costs

```
def get_cost_tuples(distinct_costs: list[int],
                    costs_to_items: dict[int, list[int]],
                    budget: int, i: int) -> tuple[list[tuple[int, ...]], bool]:
    """
```

```
:param distinct_costs: is a list of the distinct costs, sorted from greatest
to least. M = len(distinct_costs)
:param costs_to_items: maps costs to sets of items with that cost
:param budget:
:param i: 0 <= i <= M - 1
:return: a tuple containing a list of all maximal tuples of
len(distinct_costs) - i nonnegative integers such that each tuple
(mult_i, ..., mult_{M - 1}) is such that
sum_{p = i}^{M - 1} distinct_costs[p] * mult_p <= budget,
and a bool stating if all items with costs in distinct_costs[i:] are
simultaenously affordable with the budget

>>> cost_tuples, fully_affordable = get_cost_tuples([3, 2, 1],
>>>                                                 {1: [1, 2, 3, 10],
>>>                                                  2: [4, 5, 6],
>>>                                                  3: [7, 8, 9]}, 6, 0)
>>> fully_affordable
False
>>> cost_tuples == [(2, 0, 0), (1, 1, 1), (1, 0, 3), (0, 3, 0), (0, 2, 2),
>>>                 (0, 1, 4)]
True
"""
cost = distinct_costs[i]

if i == len(distinct_costs) - 1:
    multiplicity = min(len(costs_to_items[cost]), budget // cost)
    return [(multiplicity,)], multiplicity == len(costs_to_items[cost])

cost_tuples: list[tuple[int, ...]] = []

for multiplicity in range(min(len(costs_to_items[cost]), budget // cost),
                          -1, -1):
    sub_cost_tuples, fully_affordable\
        = get_cost_tuples(distinct_costs, costs_to_items,
```

```
                        budget - cost * multiplicity, i + 1)
        cost_tuples += [(multiplicity,) + sub_cost_tuple
                        for sub_cost_tuple in sub_cost_tuples]
        if fully_affordable:
            return cost_tuples, multiplicity == len(costs_to_items[cost])
    return cost_tuples, False


def best_selection(num_items: int, fixed_items: list[int],
                   decision_items: list[int], optional: bool,
                   utilities: list[int]) -> tuple[int, list[int],
                                                  list[int], bool]:
    """
    :param num_items: the max number of items to select
    :param fixed_items: list of items that must be selected
    :param decision_items: list of items from which we take a subset to add to
    the selection
    :param optional: if this is False then the selection needs to include as
    many of the tied_items as possible
    :param utilities: utilities[j - 1] is the utility given to item j
    :return: a tuple representing all best possible selections of items
    according to the given utility function, subject to the given contraints.
    This tuple contains the utility of these selections,
    which items must be selected, which items still need to be decided on,
    and whether or not a maximal subset of the items that still need to be
    decided on must be chosen
    """
    num_remaining_items = num_items - len(fixed_items)

    # prune disliked items if allowed and sort from best to worst
    decision_items = sorted([item for item in decision_items
                             if not optional or utilities[item - 1] >= 0],
                            key=lambda item: utilities[item - 1], reverse=True)
```

```
# the smallest utility that this voter would allow subject to the current
# constraints
smallest_selected_item_utility = utilities[
    decision_items[min(num_remaining_items, len(decision_items)) - 1] - 1]

if (len(decision_items) < num_remaining_items
        and (smallest_selected_item_utility > 0 or not optional)):
    new_fixed_items = fixed_items + decision_items
    new_decision_items = []
else:
    new_decision_items_start = -1
    new_decision_items_end = -1

    for item_index, item in enumerate(decision_items):
        if (utilities[item - 1] == smallest_selected_item_utility
                and new_decision_items_start == -1):
            new_decision_items_start = item_index
        elif (utilities[item - 1] < smallest_selected_item_utility
              and new_decision_items_end == -1):
            new_decision_items_end = item_index
            break

    if (new_decision_items_start + 1 == new_decision_items_end
            and smallest_selected_item_utility > 0):
        new_fixed_items = fixed_items + decision_items[
                                            :new_decision_items_end]
        new_decision_items = []
    else:
        new_fixed_items = fixed_items + decision_items[
                                            :new_decision_items_start]
        new_decision_items = (decision_items[new_decision_items_start:
                              new_decision_items_end]
                              if len(new_fixed_items) < num_items else [])
```

```python
    return ((sum(utilities[item - 1] for item in fixed_items)
              + smallest_selected_item_utility
              * (num_items - len(new_fixed_items))),
             new_fixed_items,
             new_decision_items,
             optional and smallest_selected_item_utility == 0)


def compute_sp_po_outcomes_when_num_costs_bounded(distinct_costs: list[int],
                                                  costs_to_items: dict[
                                                      int, list[int]],
                                                  budget: int,
                                                  utilities: list[list[int]]
                                                  ) -> dict[
                                                        list[tuple[int, ...]],
                                                        list[tuple[set,
                                                                   set, bool]]]:
    """
    :param distinct_costs: a list of the distinct costs that items may have
    :param costs_to_items: a dictionary which takes a cost in distinct costs and
    gives a list of items with that cost
    :param budget: the total budget
    :param utilities: utilities[i - 1][j - 1] is the utility of voter i for
    item j
    :return: a Pareto optimal outcome that was selected via serial dictatorship
    """
    num_voters = len(utilities)
    distinct_costs = sorted(distinct_costs, reverse=True)

    cost_tuples, _ = get_cost_tuples(distinct_costs, costs_to_items, budget, 0)
    # takes a cost tuple and gives a representation of a set of outcomes,
    # initially all possible outcomes with at most
    # cost_tuples[i] items with cost distinct_costs[i].
    # A set of outcomes is represented by a list, where the
```

```
# represented set is the Cartesian product of the sets represented by each
# element of the list.
# outcomes[i] = (fixed_items, decision_items, optional) represents all sets
# of size at most cost_tuples[i]
# containing fixed_items and a subset of decision_items, where that subset
# must be maximal subject to the size
# constraint if not optional
cost_tuple_to_outcomes = {cost_tuple: [(set(), costs_to_items[cost], True)
                                        for cost in distinct_costs] for
                          cost_tuple in cost_tuples}
for voter in range(num_voters):
    # the best utility this voter can realize
    best_utility = float('-inf')
    temp_cost_tuple_to_outcomes = {}

    # check what the best utility each cost tuple can give subject to the
    # current constraints
    for cost_tuple, outcomes in cost_tuple_to_outcomes.items():
        current_utility = 0
        new_outcomes = []
        for cost_index, chosen_items in enumerate(outcomes):
            num_items = cost_tuple[cost_index]
            fixed_items, decision_items, optional = chosen_items
            delta_utility, new_fixed_items, new_tied_items, new_optional\
                = best_selection(num_items, fixed_items, decision_items,
                                 optional, utilities[voter])
            current_utility += delta_utility
            new_outcomes.append((new_fixed_items, new_tied_items,
                                 new_optional))

        # if this cost tuple can give this voter better utility than the
        # previous ones, delete the previous ones and
        # restrict the outcomes for this cost tuple to the optimal ones
        if current_utility > best_utility:
```

```
            temp_cost_tuple_to_outcomes = {cost_tuple: new_outcomes}
            best_utility = current_utility

        # if this cost tuple can give this voter the same utility as the
        # best utility so far,
        # restrict the outcomes for this cost tuple to the optimal ones
        # and add it
        elif current_utility == best_utility:
            temp_cost_tuple_to_outcomes[cost_tuple] = new_outcomes
    cost_tuple_to_outcomes = temp_cost_tuple_to_outcomes
return cost_tuple_to_outcomes
```