Computational Social Choice 2021

Ulle Endriss
Institute for Logic, Language and Computation
University of Amsterdam

[http://www.illc.uva.nl/~ulle/teaching/comsoc/2021/]

Plan for Today

Suppose a municipality wants to consult residents on how to spend some of its *budget*. There are several *projects*, each with some *cost*. People *vote*. We need a *mechanism* to choose which projects to fund.

This lecture is an introduction to this idea of *participatory budgeting:*

- formal *model* for PB, with a focus on *approval ballots*
- discussion of various *mechanisms* and their properties:
  - algorithmic considerations: complexity of computing outcomes
  - axiomatic considerations: strategyproofness, proportionality, ...

Remark: In practice, voting given a set of projects is just the final stage of PB (after asking people to submit proposals for shortlisting).

History

First used in the city of Porto Alegre in Brazil in the late 1980s. Widely considered a success, with noticeable improvements to living standards and high levels of participation by the electorate.

Since then used in many dozens of cities and towns all over the world. First (?) used in Amsterdam in late 2019 (in Slotermeer Noordoost). Used all over Amsterdam in 2020 ("*buurtbudget*").

Fairly recent interest in the scientific community to study PB more systematically also from a social choice perspective.

Y. Cabannes. Participatory Budgeting: A Significant Contribution to Participatory Democracy. *Environment and Urbanization*, 2004.

A. Shah (editor). Participatory Budgeting. *Public Sector Governance and Accountability Series*. The World Bank, 2007.

The Model

Fix a finite set $P = \{p_1, p_2, p_3, \ldots\}$ of *projects* a community may wish to realise, a *cost function* $c : P \to \mathbb{N}$, and a *budget* $b \in \mathbb{N}$.

The set of *feasible outcomes* is the set of affordable sets of projects:

$$\textsc{Feas}(P, c, b) \quad = \quad \{\, S \subseteq P \mid c(S) \leqslant b \,\}, \text{ where } c(S) = \textstyle\sum_{x \in S} c(x)$$

To select a feasible outcome from this set, we query the members of a set $N = \{1, \ldots, n\}$ of *voters* for their preferences. Options include:

- could ask each voter for her k *favourite projects* (e.g., $k = 5$)
- could ask each voter for her *favourite feasible outcome*
- could ask each voter to *approve* any number of *projects*
- could ask each voter to *approve* any number of *feasible outcomes*
- could ask each voter to *rank* all/some *projects*
- could ask each voter to *rank* all/some *feasible outcomes*

Exercise: *Which of these balloting regimes seem reasonable to you?*

Exhaustiveness

Are all feasible outcomes also reasonable outcomes? Maybe not.
Maybe we want to *exhaust* the available budget:

$$\textsc{Exh}(P, c, b) \quad = \quad \{\, S \in \textsc{Feas}(P, c, b) \mid c(S^+) > b \text{ for all } S^+ \supset S \,\}$$

Possible variations regarding exhaustiveness:

- only exhaust the budget with projects 'liked' by at least one voter (actually: in practice this will the the case for *all* projects)
- assign value to saving money (could be modelled via 'projects' that simply amount to saving a certain amount of money)

Extensions of the Basic Model

Various extensions of the basic model are conceivable, but have not yet received a lot of attention in the literature:

- Impose *multidimensional budget limits*, i.e., limits w.r.t. several resources, not just money (but also time, space, ...).
- Impose *non-budgetary constraints* on the set of feasible outcomes (such as dependencies, mutual exclusivity, diversity, ...)
- Permit *fractional completion* of projects, based on the amount of funding each project receives for a given outcome.
- Consider *non-additive cost functions* to model economies of scale or other synergies between projects.

Mechanisms for Approval Ballots

For the remainder of today, we restrict attention to *approval ballots* (voters each approve of some projects). This is the most common ballot regime used in practice and analysed in the literature.

So a *ballot* is a set $A_i \subseteq P$ and a *profile* is a vector $\boldsymbol{A} = (A_1, \ldots, A_n)$.

Two common restrictions:

- Voters must pick $\leqslant k$ projects. (No good normative argument.)
- Voters must pick a *feasible* set of projects. (Sounds reasonable.)

A PB *mechanism* for N and $\langle P, c, b \rangle$ selects one or (in case of a tie) more feasible outcomes for any given profile of approval ballots:

$$F : (2^P)^n \to 2^{\textsc{Feas}(P,c,b)} \setminus \{\emptyset\}$$

Must use a tie-breaking rule to achieve resoluteness.

The Greedy Approval Mechanism

The greedy approval mechanism works as follows:

accept projects in order or their approval score
(using lexicographic tie-breaking),
skipping projects rendering the outcome infeasible

Observation: Outcomes are *feasible* and *exhaustive* by construction.

Remark: With the (*ad hoc*) restriction to ballots of *cardinality* $\leqslant k$, greedy approval is the most common mechanism used in practice.

Remark: With the (reasonable) restriction of ballots to *feasible sets*, greedy approval is known as *knapsack voting* (Goel et al., 2019).

A. Goel, A.K. Krishnaswamy, S. Sakshuwong, and T. Aitamurto. Knapsack Voting for Participatory Budgeting. *ACM Trans. on Economics and Computation*, 2019.

Other Simple Mechanisms

The *max-approval* mechanism seeks to maximise total approval:

$$F_{\max}(\boldsymbol{A}) \quad \in \quad \operatorname*{argmax}_{S \in \textsc{Feas}(P,c,b)} \sum_{i \in N} |S \cap A_i|$$

Outcomes under max-approval and greedy approval can differ a lot: think of scenarios with one very popular but expensive project ...

Other simple mechanisms may differ in how they weigh approvals of cheap *vs.* expensive projects when determining the 'best' outcome.

Exercise: *Suppose we 'correct' approval scores to account for costs. Argument for multiplying by project cost? For dividing instead?*

Talmon and Faliszewski (2019), using randomly generated data, study experimentally the extent to which some of these mechanisms differ.

N. Talmon and P. Faliszewski. A Framework for Approval-based Budgeting Methods. AAAI-2019.

Computational Complexity

Computing *greedy approval* outcomes is straightforward. Nice. Also:

Proposition 1 (Talmon and Faliszewski, 2019) *The max-approval mechanism can be executed in polynomial time.*

Proof: We use the well-known *dynamic programming* approach for knapsack problems. Compute for each $k \leqslant |P|$ and $t \leqslant n \cdot |P|$ the smallest budget $b[k,t]$ that can achieve a total approval of t using only projects in $\{p_1, \ldots, p_k\}$ (and keep track of which outcome does):

$$
\begin{aligned}
b[1,t] &:= c(p_1) \text{ if } t = \#\{i \mid p_1 \in A_i\} \text{ and undefined otherwise} \\
b[k,t] &:= \min\{b[k-1,t],\ b[k-1, t - \#\{i \mid p_k \in A_i\}] + c(p_k)\}
\end{aligned}
$$

Here undefined values within $\min\{\cdot\}$ are taken to be ∞ (i.e., omitted).

Return the outcome associated with any $b[k,t] \leqslant b$ maximising t. ✓

N. Talmon and P. Faliszewski. A Framework for Approval-based Budgeting Methods. AAAI-2019.

Strategyproofness

Suppose each voter $i \in N$ has one most preferred outcome $S_i^\star \subseteq P$ and she (weakly) prefers outcome S to S' iff $c(S \cap S_i^\star) \geqslant c(S' \cap S_i^\star)$.

Exercise: *Give an intuitive motivation for this formal definition.*

Mechanism F is called *strategyproof* if for every voter i and profile $\boldsymbol{A}$ it is the case that i (weakly) prefers $F(\boldsymbol{A}_{-i}, S_i^\star)$ to $F(\boldsymbol{A})$.

This example shows that *greedy approval* is *not* strategyproof:

> $P = \{p_1, p_2, p_3\}$ with $c(p_1) = 2$, $c(p_2) = 4$, $c(p_3) = 2$ and $b = 4$.
> You want outcome $\{p_1, p_2\}$. Suppose: tie w/o your vote.
> Under lexicographic tie-breaking, you are best off voting $\{p_2\}$!

Same problem for *max-approval* (just imagine you are the only voter).

Strategyproofness for Special Cases

Our counterexample (which works even with just one voter!) clearly suggests that finding a strategyproof mechanism is impossible.

Nevertheless, the proof of this simple result is straightforward:

Lemma 2 *If all projects have cost 1, greedy approval is strategyproof.*

Remark: This result corresponds to a basic (folklore) result in the area of *multiwinner voting* (to elect a committee of size k).

Approximate Strategyproofness

Let us call a mechanism F *approximately strategyproof* if for every voter i and profile $\boldsymbol{A}$ there exists a project $p^\star$ such that this voter i (weakly) prefers $F(\boldsymbol{A}_{-i}, S_i^\star) \cup \{p^\star\}$ to $F(\boldsymbol{A})$.

Results by Goel et al. (2019), who mostly are concerned with knapsack voting for fractional PB, (essentially) entail:

Theorem 3 *The greedy approval rule is approximately strategyproof.*

Proof: Think of each project as a set of *subprojects* of cost 1 each. So on the *input side*, non-unit costs act as a domain restriction. *Good!* On the *output side*, non-unit costs act as constraints on accepting subprojects together. *Bad!* But: subprojects of same project have *same score*, so subprojects of *at most one* project will get rejected that would have been accepted one-by-one. Let $p^\star$ be that project. ✓

A. Goel, A.K. Krishnaswamy, S. Sakshuwong, and T. Aitamurto. Knapsack Voting for Participatory Budgeting. *ACM Trans. on Economics and Computation*, 2019.

Monotonicity Axioms

We would like our mechanism to be 'well-behaved' when responding to small changes in the specification of a PB scenario. Examples:

- increasing the budget limit
- decreasing the cost of a project
- splitting a project into subprojects (or merging subprojects)

Talmon and Faliszewski (2019) formulate several *monotonicity axioms* to account for such changes. Here we discuss just one of them:

> A resolute mechanism $F(\cdot,\cdot)$ defined across PB instances is *discount-monotonic* if $p \in F(\boldsymbol{A}, \langle P, c, b\rangle) \Rightarrow p \in F(\boldsymbol{A}, \langle P, c', b\rangle)$ whenever $c(p) > c'(p)$ and $c(p') = c'(p')$ for all $p' \in P \setminus \{p\}$.

Exercise: *Show that greedy approval and max-approval are DM.*

N. Talmon and P. Faliszewski. A Framework for Approval-based Budgeting Methods. AAAI-2019.

Proportionality

We should be careful to protect minorities during PB. Thus:

Any *sufficiently large group* of *sufficiently like-minded voters* should be *sufficiently well represented* when we decide which projects to fund.

Aziz et al. (2018) formulate several (fairly demanding) axioms that attempt to formalise this intuition. Here is a fairly weak formulation:

> A resolute mechanism F is *proportional* if, for any coalition $C \subseteq N$, project set $S \subseteq P$ with $c(S) \leqslant \frac{|C|}{n} \cdot b$, and profile $\boldsymbol{A}$ with $A_i = S$ for all $i \in C$, we have $S \subseteq F(\boldsymbol{A})$.

Exercise: *Is the greedy approval mechanism proportional in this sense?*

H. Aziz, B.E. Lee, and N. Talmon. Proportionally Representative Participatory Budgeting: Axioms and Algorithms. AAMAS-2018.

The Greedy Load-Balancing Mechanism

Aziz et al. (2018) propose using load-balancing mechanisms based on ideas of Swedish mathematician Lars Edvard Phragmén (1863–1937).

Here is one possible instantiation of this approach:

- Initialise: accepted projects $A := \emptyset$ and voter loads $L_i := 0$.
- Accepting a project p means making these updates:

$$A := A \cup \{p\} \qquad L_i := \frac{c(p) + \sum\{L_j \mid p \in A_j\}}{\#\{j \mid p \in A_j\}} \text{ if } p \in A_i$$

- Accept projects one-by-one, picking one *minimising* $\max_i L_i$ in each round, while always ensuring that A remains feasible.

H. Aziz, B.E. Lee, and N. Talmon. Proportionally Representative Participatory Budgeting: Axioms and Algorithms. AAMAS-2018.

M. Brill, R. Freeman, S. Janson, and M. Lackner. Phragmén's Voting Methods and Justified Representation. AAAI-2017.

Proportionality of Greedy Load-Balancing

Good news:

Proposition 4 *The greedy load-balancing mechanism is proportional.*

<u>Proof:</u> Consider $\boldsymbol{A}$, S, C s.t. $A_i = S$ for all $i \in C$ and $c(S) \leqslant \frac{|C|}{n} \cdot b$.
W.l.o.g., suppose $S \cap A_i = \emptyset$ for all $i \in N \setminus C$ (as this is worst case).
What if the voters in $N \setminus C$ get more than $b - \frac{|C|}{n} \cdot b$? Then at least one of them must have a load L_i of strictly more than this:

$$b \cdot (1 - \tfrac{|C|}{n}) \;\; / \;\; |N \setminus C| \;\; = \;\; \tfrac{b}{n}$$

But before the load of any voter reaches that level, the mechanism would first accept all of S with a load for each $i \in C$ of at most $\frac{b}{n}$. ✓

Aziz et al. (2018) show that this result extends also to a more demanding formulation of the proportionality axiom.

H. Aziz, B.E. Lee, and N. Talmon. Proportionally Representative Participatory Budgeting: Axioms and Algorithms. AAMAS-2018.

Other Properties

What about the other properties we have considered?

- Greedy load-balancing is computationally *efficient:* the mechanism clearly can be executed in *polynomial time*.
- Greedy load-balancing also clearly is *discount-monotonic:* making a project cheaper only ever eases the load to be balanced.
- However, greedy load-balancing is *not strategyproof*, not even approximately so: you can manipulate by not approving of popular projects, letting other supporters shoulder the load ("free-riding").

Participatory Budgeting in Amsterdam in 2019

Slotermeer Noordoost (∼9,300 inhabitants). Budget of € 500k.

Greedy approval with 40 projects, costing between € 1.7k and € 250k.

1,273 citizens voted. 13 projects funded (6/13 concern *Plein '40–'45*).

https://buurtbudgetsno.amsterdam.nl/

Summary

This has been an introduction to the topic of *participatory budgeting*, with a focus on mechanisms for *approval ballots*. Properties considered:

- computational *complexity* of computing outcomes
- *strategyproofness* and approximate strategyproofness
- basic 'well-behavedness' axioms, notably *discount monotonicity*
- *proportional representation* of coherent groups of voters

The main mechanism we have looked into is the basic and widely used *greedy approval* mechanism, but we have also discussed the basic *max-approval* mechanism and a *greedy load-balancing* mechanism.

$\hookrightarrow$ *Have a look at* `Pabulib.org` *for data on real-world PB exercises.*

What next? Multiwinner voting (to elect a committee of size k).