Participatory Budgeting with Multiple Resources

Nima Motamed[1], *Arie Soeteman*[2], Simon Rey[2], Ulle Endriss[2]

September 15, 2022 | EUMAS 2022 | Düsseldorf, Germany

[1] Intelligent Systems, Utrecht University
[2] Institute for Logic, Language and Computation, University of Amsterdam

What's so good about PB?

Dit plan wordt uitgevoerd

Vergroenen openbare ruimte

Geuzenveld, Slotermeer, Sloterdijken

> Lees meer

€ 65.000 | 1462 stemmen

Dit plan wordt uitgevoerd

Opknappen Natuurspeeltuin Nature...

Geuzenveld, Slotermeer, Sloterdijken

> Lees meer

€ 50.000 | 1216 stemmen

Dit plan wordt uitgevoerd

Bloementuin in het Sloterpark

Geuzenveld, Slotermeer, Sloterdijken

> Lees meer

€ 5.000 | 1207 stemmen

Dit plan wordt uitgevoerd

Bijeenkomsten voor eenzame ouderen

Geuzenveld, Slotermeer, Sloterdijken

> Lees meer

€ 18.780 | 1000 stemmen

Dit plan wordt uitgevoerd

Bewoners Restaurant Armoedebestr...

Geuzenveld, Slotermeer, Sloterdijken

> Lees meer

€ 10.000 | 981 stemmen

Dit plan wordt uitgevoerd

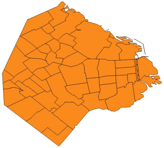
Voedselbos in het Sloterpark

Geuzenveld, Slotermeer, Sloterdijken

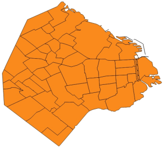
> Lees meer

€ 20.000 | 948 stemmen

Introducing Multiple Resources



Introducing Multiple Resources



Officials often need to interfere in the process (Goldfrank, 2007)

MRPB has been recognized as an important challenge (Haris Aziz & Nisarg Shah, 2020)

Usual PB framework

The 'usual' PB framework often looks like this:

- Set P of projects
- Cost function $c : P \to \mathbb{N}$
- Budget limit $b \in \mathbb{Z}_+$
- Each voter i submits some sort of ballot A_i, making a profile $\boldsymbol{A} = (A_1, \ldots, A_n)$

Project set $S \subseteq P$ is *Feasible* if $\sum_{p \in S} c(p) \leq b$

Our framework

A *d-resource PB scenario* is a tuple $\langle P, \mathbf{c}, \mathbf{b} \rangle$:

- P is a set of projects
- $\mathbf{c}$ is a vector of cost functions $c_k : P \to \mathbb{N} \cup \{0\}$ for $k = 1 \ldots d$
- $\mathbf{b}$ is a vector of budget limits $b_k \in \mathbb{N}$ for $k = 1 \ldots d$

A set $S \subseteq P$ is *feasible* if $\sum_{p \in S} c_k(p) \leqslant b_k$ for all $k = 1 \ldots d$.

Voters $i \in \{1, \ldots, N\}$ submit approval ballots $A_i \subseteq P$

Approval ballots make up a *profile* $\boldsymbol{A} = (A_1, \ldots, A_n)$

Other constraints & relations to other frameworks

Distributional: spend at most $\alpha \in [0, 1]$ of b_k on $X \subseteq P$

Incompatibility: not all projects in $X \subseteq P$ can be realised simultaneously

Dependency: p can only be realised if all projects in X are realised

Other constraints & relations to other frameworks

Distributional: spend at most $\alpha \in [0,1]$ of b_k on $X \subseteq P$

Add $k*$ with $b_{k*} = \lfloor \alpha \cdot b_k \rfloor$, and $c_{k*}(p) = \mathbb{1}_{p \in X} \cdot c_k(p)$

Incompatibility: not all projects in $X \subseteq P$ can be realised simultaneously

Add $k*$ with $b_{k*} = |X| - 1$ and $c_{k*}(p) = \mathbb{1}_{p \in X}$

Dependency: p can only be realised if all projects in X are realised

Add $k*$ with $b_{k*} = 1$, $c_{k*}(p) = |X| + 1$, and $c_{k*}(q) = -1$ for all $q \in X$

Mechanisms

A mechanism is a function F that takes as input scenarios $\langle P, \mathbf{c}, \mathbf{b} \rangle$ and profiles $\boldsymbol{A}$ and returns feasible set $F(P, \mathbf{c}, \mathbf{b}, \boldsymbol{A}) \subseteq P$

Mechanisms

A mechanism is a function F that takes as input scenarios $\langle P, \mathbf{c}, \mathbf{b} \rangle$ and profiles $\boldsymbol{A}$ and returns feasible set $F(P, \mathbf{c}, \mathbf{b}, \boldsymbol{A}) \subseteq P$

- F_{greedy}: Go through projects in order of approval score, adding them to the outcome set one by one while skipping projects making outcome infeasible
- F_{max} returns feasible set with maximal approval score

Mechanisms

A mechanism is a function F that takes as input scenarios $\langle P, \mathbf{c}, \mathbf{b} \rangle$ and profiles $\boldsymbol{A}$ and returns feasible set $F(P, \mathbf{c}, \mathbf{b}, \boldsymbol{A}) \subseteq P$

- F_{greedy}: Go through projects in order of approval score, adding them to the outcome set one by one while skipping projects making outcome infeasible
- F_{max} returns feasible set with maximal approval score
- F_{load} proceeds in steps: at each step, chooses the project minimizing the load (cost) carried by the worst-off voter

Axioms

Proportionality

All projects in set S are selected if for all $k \in R$:
$\frac{|\{i \in N; A_i = S\}|}{n} \geq \frac{c_k(S)}{b_k}$

Weak axiom only guarantees this if $|S| = 1$

Axioms

Proportionality

All projects in set S are selected if for all $k \in R$:
$\frac{|\{i \in N; A_i = S\}|}{n} \geq \frac{c_k(S)}{b_k}$

Weak axiom only guarantees this if $|S| = 1$

(Approximate) Strategyproofness

For truthful ballot S_i^*, $F(\boldsymbol{A}) \not\succ_i F(A_{-i}, S_i^*)$

Approximate: for some $p \in P$: $F(\boldsymbol{A}) \not\succ_i F(A_{-i}, S_i^*) \cup \{p\}$

Here we define different preferences $\succ_i$: prefer a Superset, or also an outcome that is better w.r.t. all resources (Paretian)

Axioms

Proportionality

All projects in set S are selected if for all $k \in R$:
$\frac{|\{i \in N; A_i = S\}|}{n} \geq \frac{c_k(S)}{b_k}$

Weak axiom only guarantees this if $|S| = 1$

(Approximate) Strategyproofness

For truthful ballot S_i^*, $F(\boldsymbol{A}) \not\succ_i F(A_{-i}, S_i^*)$

Approximate: for some $p \in P$: $F(\boldsymbol{A}) \not\succ_i F(A_{-i}, S_i^*) \cup \{p\}$

Here we define different preferences $\succ_i$: prefer a Superset, or also an outcome that is better w.r.t. all resources (Paretian)

Actually, our definitions are parameterized by a set R of relevant resources, giving more fine-grained analysis (and slightly different definitions)

Results

	Subset Preferences	Paretian Preferences	Paretian Preferences if $R = \{1 \dots d\}$
Greedy	✔	✘	✔
Max	✘	✘	✘
Load Balancing	✘	✘	✘

Approximate Strategyproofness

	Strong	Weak
Greedy	✘	✘
Max	✘	✘
Load Balancing	✔	✔

Proportionality

No mechanisms are strategyproof (even for $d = 1$)

Results

An impossibility result:

Theorem

Let $d \geqslant 1$, $m > b_k \geqslant 3$ for some resource k, then no mechanism can guarantee both weak proportionality and strategyproofness against Paretian voters for d-resource PB scenarios with budgets $(b_1, \ldots, b_k, \ldots b_d)$ and m projects.

Basecase is generated using a SAT-solving approach

Computational analysis

F_{greedy} and F_{load} are polytime computable

Computational analysis

F_{greedy} and F_{load} are polytime computable

For $F_{\max}$ multiple decision problems:

Definition (MaxAppScore)

Instance: PB scenario $\langle P, \mathbf{c}, \mathbf{b} \rangle$, profile $\boldsymbol{A}$, target $K \in \mathbb{N}$

Question: Is there feasible $S \subseteq P$ with approval score at least K?

(MaxAppScore$_d$ restricts to d-resource scenarios)

Computational analysis

F_{greedy} and F_{load} are polytime computable

For $F_{\max}$ multiple decision problems:

Definition (MaxAppScore)

Instance: PB scenario $\langle P, \mathbf{c}, \mathbf{b} \rangle$, profile $\boldsymbol{A}$, target $K \in \mathbb{N}$

Question: Is there feasible $S \subseteq P$ with approval score at least K?

(MaxAppScore$_d$ restricts to d-resource scenarios)

MaxAppScore$_1$ (and $F_{\max}$ in single-resource case) is polytime computable per Talmon & Faliszewski (2019);

MaxAppScore is strongly NP-hard;

MaxAppScore$_d$ for $d \geqslant 2$ is weakly NP-hard, and $F_{\max}$ is pseudo-polytime computable with restriction to d

Wrapping up

Summing up:

- Initiated the systematic study of PB with multiple resources
- New setting has significantly increased expressive power
- Mechanisms from single-resource setting largely carry over nice axiomatic & algorithmic properties

Wrapping up

Summing up:

- Initiated the systematic study of PB with multiple resources
- New setting has significantly increased expressive power
- Mechanisms from single-resource setting largely carry over nice axiomatic & algorithmic properties

What's next?

- Strengthen the results to e.g. other voter preferences, and other notions of proportionality
- Explore the introduction of negative costs
- Eventually implement multi-resource PB in real-world PB exercises

Wrapping up

Summing up:

- Initiated the systematic study of PB with multiple resources
- New setting has significantly increased expressive power
- Mechanisms from single-resource setting largely carry over nice axiomatic & algorithmic properties

What's next?

- Strengthen the results to e.g. other voter preferences, and other notions of proportionality
- Explore the introduction of negative costs
- Eventually implement multi-resource PB in real-world PB exercises

Thank you!

The load-balancing mechanism

For set $R \subseteq \{1, \ldots, d\}$ of relevant resources

Build outcome S in rounds. At each round, add a project that maintains feasibility of outcome S and minimises $\max_{k \in R} y_k$, where y_k is computed by linear program with variables $x_{i,k,p}$

$$\min y_k \text{ where } y_k \geqslant \frac{1}{b_k} \cdot \sum_{p \in S} x_{i,k,p} \text{ for all } i \in N \text{ with}$$

$$\sum_{i \in N} \mathbb{1}_{p \in A_i} \cdot x_{i,k,p} = c_k(p) \text{ for all } p \in S, \text{ and } x_{i,k,p} \geqslant 0$$

Intuitively, $x_{i,k,p}$ is the part of the cost $c_k(p)$ 'shouldered' by voter i