Non-Standard Models for Participatory Budgeting

Simon Rey

November 25, 2021

1. Introduction

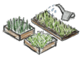
Participatory Budgeting

© Marianne de Heer Kloots

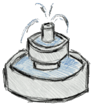
1000€

2000€

2500€



2500€



5000€

: 7000€

Participatory Budgeting

© Marianne de Heer Kloots

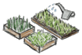
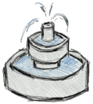
1000€ 2000€ 2500€

2500€ 5000€

: 7000€

Standard Model of Participatory Budgeting

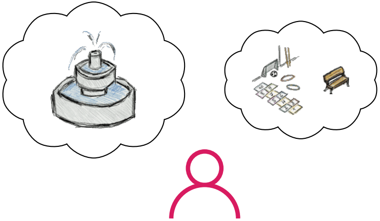
7000€

	 1000€	 2000€	 2500€	 2500€	 5000€
👤👤👤	✓✓✓	✗✗✗	✗✗✗	✗✗✗	✓✓✓
👤👤	✗✗	✓✓	✓✓	✗✗	✗✗
👤	✗	✗	✗	✗	✓
👤	✓	✗	✗	✓	✗

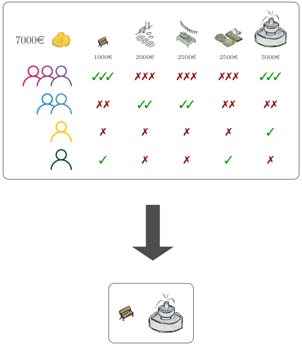
Participatory Budgeting in the ComSoC Literature

Axiomatic Analysis

Incentive Compatibility



Algorithmic Perspective

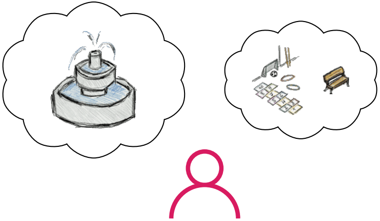


7000€	1000€	2000€	2500€	2500€	5000€
	✓✓✓	✗✗✗	✗✗✗	✗✗✗	✓✓✓
	✗✗	✓✓	✓✓	✗✗	✗✗
	✗	✗	✗	✗	✓
	✓	✗	✗	✓	✗

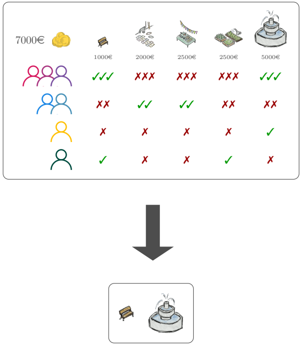
Participatory Budgeting in the ComSoC Literature

Axiomatic Analysis

Incentive Compatibility



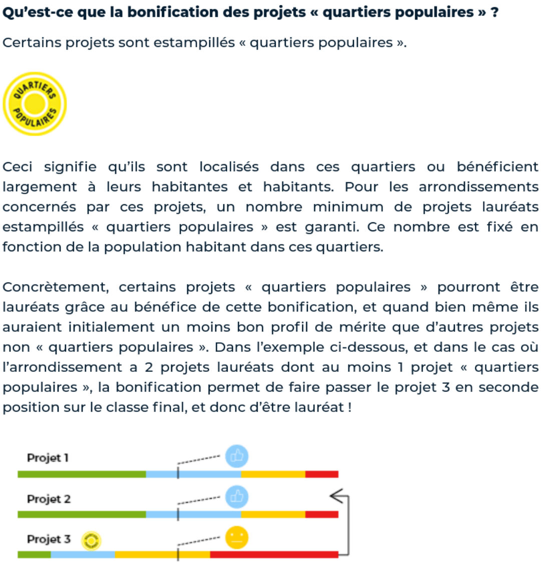
Algorithmic Perspective



7000€	1000€	2000€	2500€	2500€	5000€
	✓✓✓	✗✗✗	✗✗✗	✗✗✗	✓✓✓
	✗✗	✓✓	✓✓	✗✗	✗✗
	✗	✗	✗	✗	✓
	✓	✗	✗	✓	✗

↳ But mainly for the standard model of participatory budgeting

Sometimes Additional Constraints Are Added



Qu’est-ce que la bonification des projets « quartiers populaires » ?

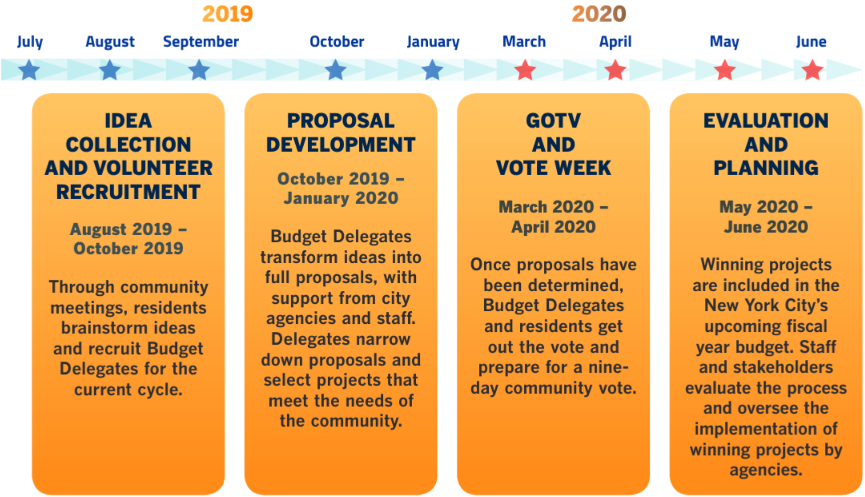
Certains projets sont estampillés « quartiers populaires ».

Ceci signifie qu’ils sont localisés dans ces quartiers ou bénéficient largement à leurs habitantes et habitants. Pour les arrondissements concernés par ces projets, un nombre minimum de projets lauréats estampillés « quartiers populaires » est garanti. Ce nombre est fixé en fonction de la population habitant dans ces quartiers.

Concrètement, certains projets « quartiers populaires » pourront être lauréats grâce au bénéfice de cette bonification, et quand bien même ils auraient initialement un moins bon profil de mérite que d’autres projets non « quartiers populaires ». Dans l’exemple ci-dessous, et dans le cas où l’arrondissement a 2 projets lauréats dont au moins 1 projet « quartiers populaires », la bonification permet de faire passer le projet 3 en seconde position sur le classe final, et donc d’être lauréat !

© Budget Participatif Paris

Sometimes Agents Propose and Vote for the Projects



IDEA COLLECTION AND VOLUNTEER RECRUITMENT

August 2019 – October 2019

Through community meetings, residents brainstorm ideas and recruit Budget Delegates for the current cycle.

PROPOSAL DEVELOPMENT

October 2019 – January 2020

Budget Delegates transform ideas into full proposals, with support from city agencies and staff. Delegates narrow down proposals and select projects that meet the needs of the community.

GOTV AND VOTE WEEK

March 2020 – April 2020

Once proposals have been determined, Budget Delegates and residents get out the vote and prepare for a nine-day community vote.

EVALUATION AND PLANNING

May 2020 – June 2020

Winning projects are included in the New York City's upcoming fiscal year budget. Staff and stakeholders evaluate the process and oversee the implementation of winning projects by agencies.

© New York Participatory Budgeting

Sometimes the Process is Repeated Over Time

© Cambridge Participatory Budgeting

PB Cycle 1 (October 2014 - April 2015)

PB Cycle 2 (June - December 2015)

PB Cycle 3 (May - December 2016)

PB Cycle 4 (May - December 2017)

PB Cycle 5 (May - December 2018)

PB Cycle 6 (May - December 2019)

PB Cycle 7 (September 2020 - January 2021)

PB Cycle 8 (June 2021 - December 2021)

Research Direction and Plan of the Talk

Can we develop the usual social choice toolkit for these non-standard PB models?

Research Direction and Plan of the Talk

Can we develop the usual social choice toolkit for these non-standard PB models?

↪ Using the expressive power of judgment aggregation for participatory budgeting to easily add extra constraints

↪ Studying a two-stage model for participatory budgeting where agents propose and vote for the projects to be implemented

↪ Developing a framework for repeated participatory budgeting processes

2. Judgment Aggregation for Participatory Budgeting

Judgment Aggregation or Binary Aggregation

Judgment aggregation is an aggregation framework where binary decisions are to be made over a set of issues:

Judgment Aggregation or Binary Aggregation

Judgment aggregation is an aggregation framework where binary decisions are to be made over a set of issues:

- Agents submit approval ballots over this set (truth assignment);

Judgment Aggregation or Binary Aggregation

Judgment aggregation is an aggregation framework where binary decisions are to be made over a set of issues:

- Agents submit approval ballots over this set (truth assignment);
- Using a judgment aggregation rule, an outcome is determined based on the ballots;

Judgment Aggregation or Binary Aggregation

Judgment aggregation is an aggregation framework where binary decisions are to be made over a set of issues:

- Agents submit approval ballots over this set (truth assignment);
- Using a judgment aggregation rule, an outcome is determined based on the ballots;
- The outcome must satisfy an integrity constraint over the issues.

Judgment Aggregation or Binary Aggregation

Judgment aggregation is an aggregation framework where binary decisions are to be made over a set of issues:

- Agents submit approval ballots over this set (truth assignment);
- Using a judgment aggregation rule, an outcome is determined based on the ballots;
- The outcome must satisfy an integrity constraint over the issues.

x_1	x_2	x_3
✓	✗	✓
✓	✗	✗
✗	✓	✓

Assume the following constraint:

$$\Gamma = (p_1 \rightarrow \neg p_3) \wedge (p_2 \rightarrow \neg p_3)$$

The admissible outcomes are then:

$$\emptyset \qquad p_1 \qquad p_2 \qquad p_3 \qquad p_1, p_2$$

Overall Idea of the Embedding

$$\left.\begin{array}{c}\text{PB profile}\\ \text{PB instance}\end{array}\right\}$$

Overall Idea of the Embedding

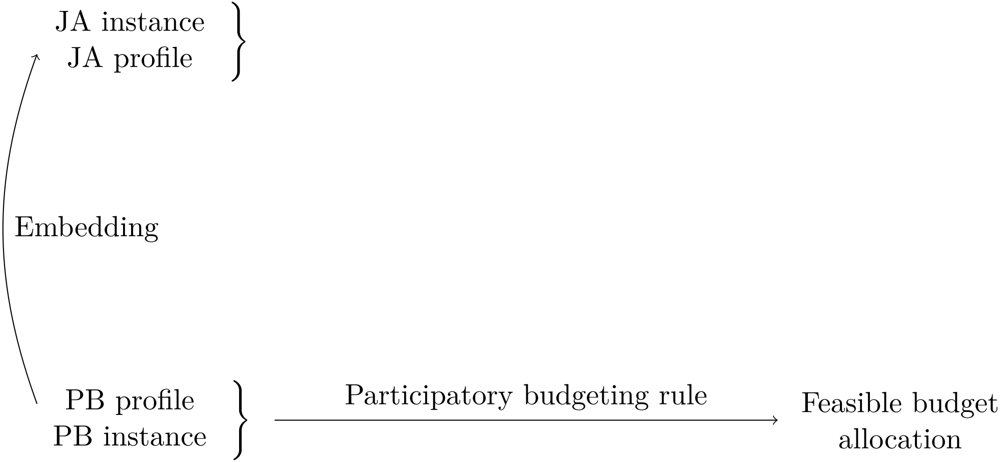
$$\left.\begin{array}{c}\text{PB profile}\\ \text{PB instance}\end{array}\right\}\xrightarrow{\text{Participatory budgeting rule}}\begin{array}{c}\text{Feasible budget}\\ \text{allocation}\end{array}$$

Overall Idea of the Embedding

$$\left.\begin{array}{c}\text{JA instance}\\ \text{JA profile}\end{array}\right\}$$

$$\left.\begin{array}{c}\text{PB profile}\\ \text{PB instance}\end{array}\right\} \xrightarrow{\text{Participatory budgeting rule}} \begin{array}{c}\text{Feasible budget}\\ \text{allocation}\end{array}$$

Overall Idea of the Embedding



JA instance
JA profile

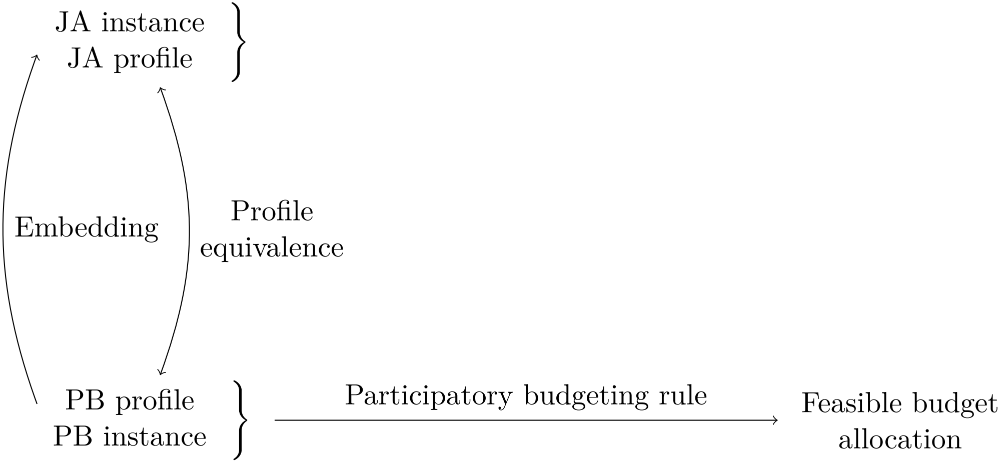
Embedding

PB profile
PB instance

Participatory budgeting rule

Feasible budget allocation

Overall Idea of the Embedding



JA instance
JA profile

Embedding

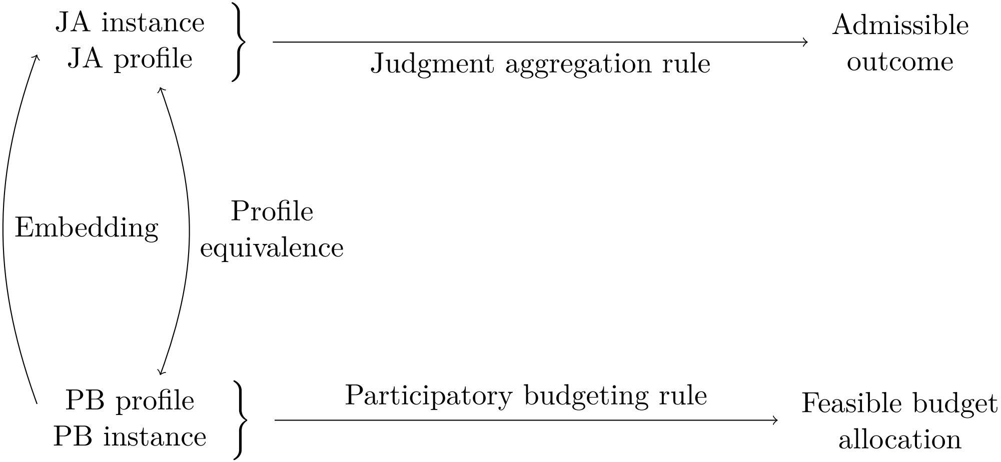
Profile equivalence

PB profile
PB instance

Participatory budgeting rule

Feasible budget allocation

Overall Idea of the Embedding



JA instance
JA profile

Judgment aggregation rule

Admissible outcome

Embedding

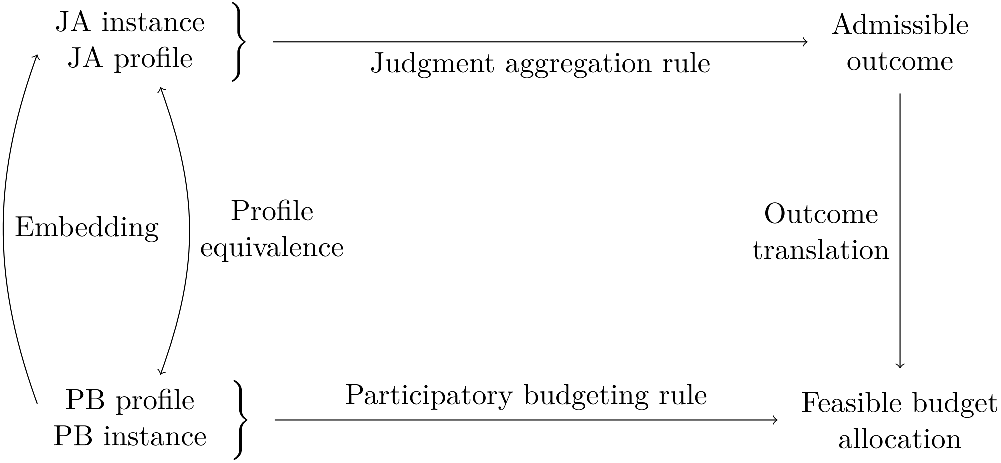
Profile equivalence

PB profile
PB instance

Participatory budgeting rule

Feasible budget allocation

Overall Idea of the Embedding



JA instance
JA profile

Judgment aggregation rule

Admissible outcome

Embedding

Profile equivalence

Outcome translation

PB profile
PB instance

Participatory budgeting rule

Feasible budget allocation

The Main Pitfall: Computational Complexity

Problem: computing the outcome of judgment aggregation rules usually is Θ_2^P-complete.

The Main Pitfall: Computational Complexity

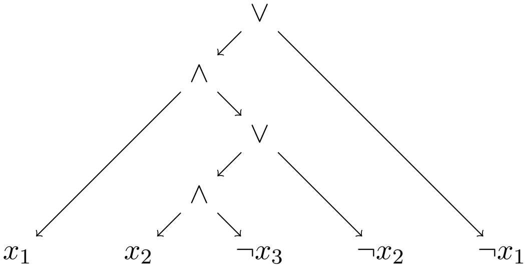
Problem: computing the outcome of judgment aggregation rules usually is Θ_2^P-complete.

↳ But not when the integrity constraint is represented as a *DNNF circuit*.

The Main Pitfall: Computational Complexity

Problem: computing the outcome of judgment aggregation rules usually is Θ_2^P-complete.

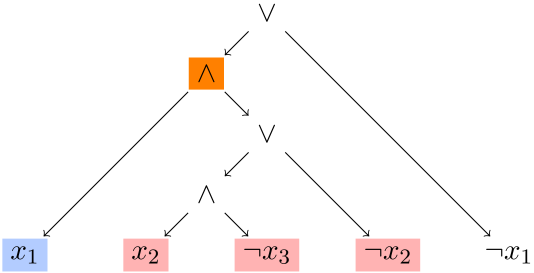
↳ But not when the integrity constraint is represented as a *DNNF circuit*.



The Main Pitfall: Computational Complexity

Problem: computing the outcome of judgment aggregation rules usually is Θ_2^P-complete.

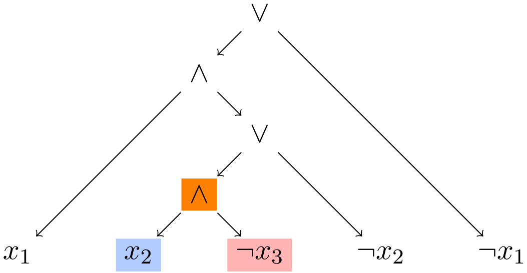
↳ But not when the integrity constraint is represented as a *DNNF circuit*.



The Main Pitfall: Computational Complexity

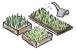
Problem: computing the outcome of judgment aggregation rules usually is Θ_2^P-complete.

↳ But not when the integrity constraint is represented as a *DNNF circuit*.



Encoding Participatory Budgeting into DNNF Circuits

5000€



2500€

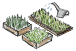
2500€

5000€

Encoding Participatory Budgeting into DNNF Circuits

5000€

$\boxed{N(0,1)}$



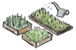
2500€

2500€

5000€

Encoding Participatory Budgeting into DNNF Circuits

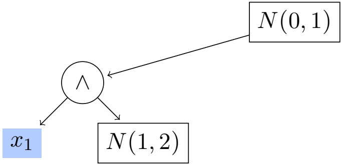
5000€



2500€

2500€

5000€



$N(0,1)$

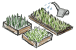
$\wedge$

x_1

$N(1,2)$

Encoding Participatory Budgeting into DNNF Circuits

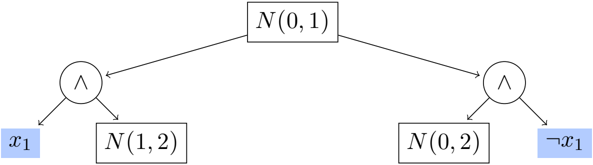
5000€



2500€

2500€

5000€



$N(0,1)$

$\wedge$

$\wedge$

x_1

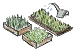
$N(1,2)$

$N(0,2)$

$\neg x_1$

Encoding Participatory Budgeting into DNNF Circuits

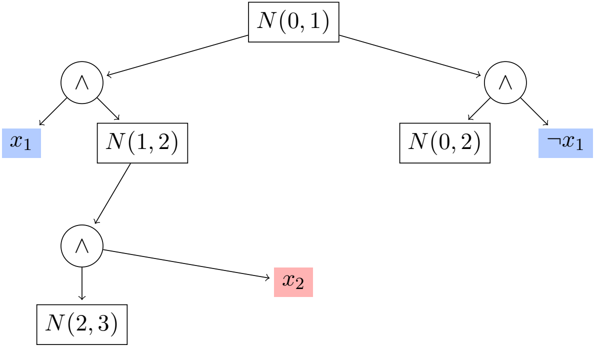
5000€



2500€

2500€

5000€



$N(0,1)$

$\wedge$

$\wedge$

x_1

$N(1,2)$

$N(0,2)$

$\neg x_1$

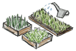
$\wedge$

x_2

$N(2,3)$

Encoding Participatory Budgeting into DNNF Circuits

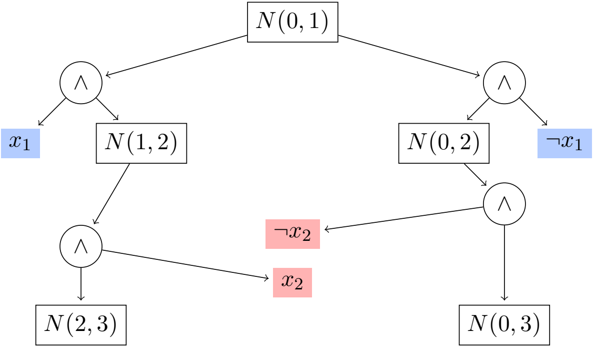
5000€



2500€

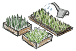
2500€

5000€



Encoding Participatory Budgeting into DNNF Circuits

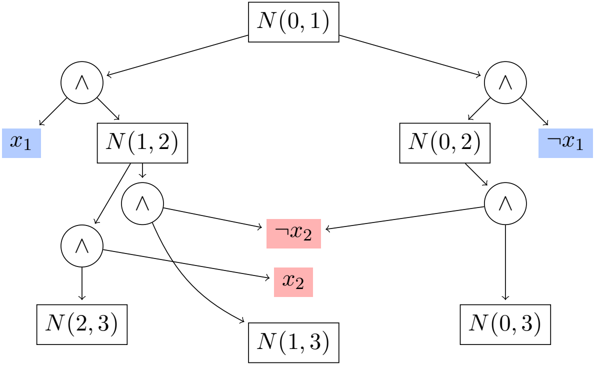
5000€



2500€

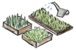
2500€

5000€



Encoding Participatory Budgeting into DNNF Circuits

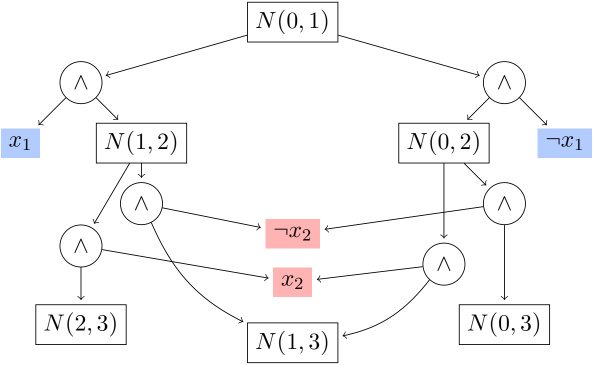
5000€



2500€

2500€

5000€



$N(0,1)$

$\wedge$ $\wedge$

x_1 $N(1,2)$ $N(0,2)$ $\neg x_1$

$\wedge$ $\wedge$

$\neg x_2$

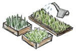
$\wedge$ $\wedge$

x_2

$N(2,3)$ $N(1,3)$ $N(0,3)$

Encoding Participatory Budgeting into DNNF Circuits

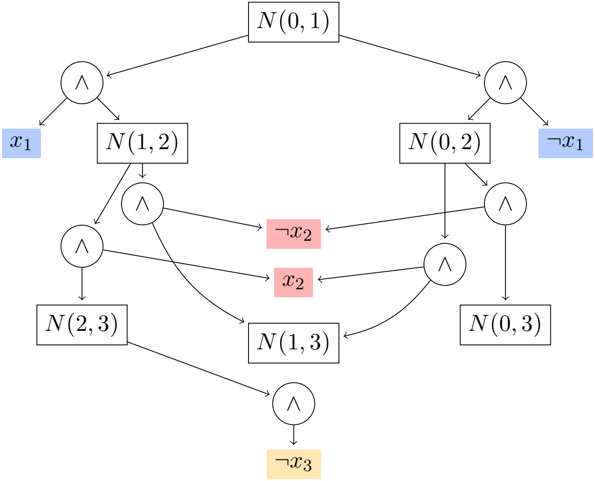
5000€



2500€

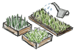
2500€

5000€



Encoding Participatory Budgeting into DNNF Circuits

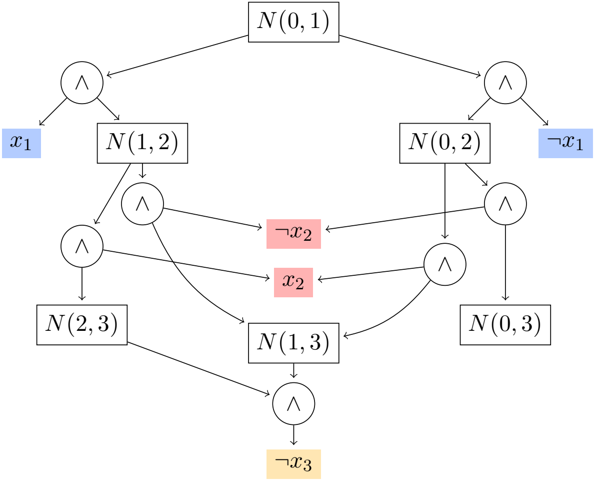
5000€



2500€

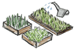
2500€

5000€



Encoding Participatory Budgeting into DNNF Circuits

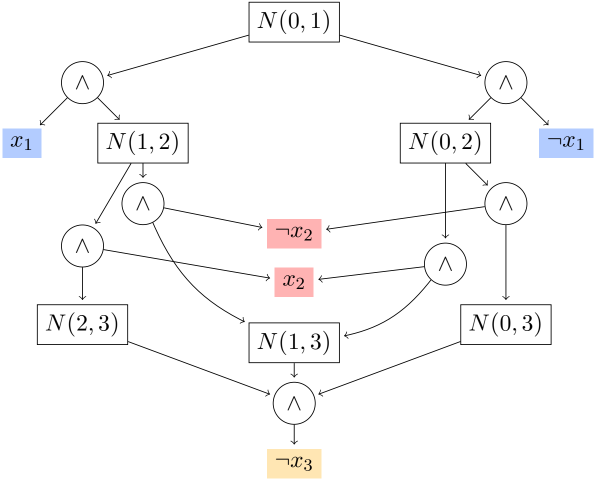
5000€



2500€

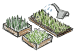
2500€

5000€



Encoding Participatory Budgeting into DNNF Circuits

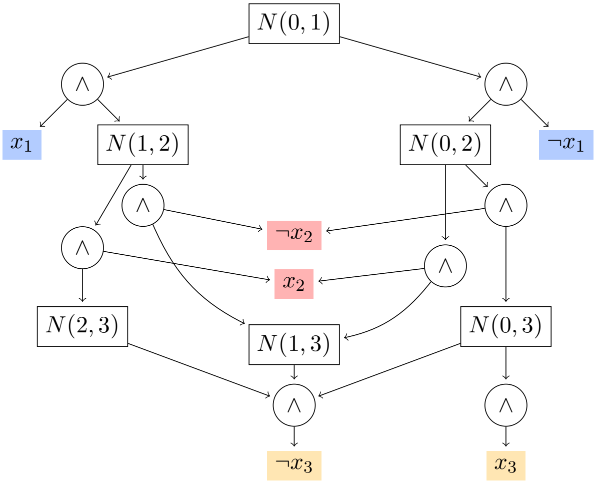
5000€



2500€

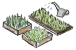
2500€

5000€



Encoding Participatory Budgeting into DNNF Circuits

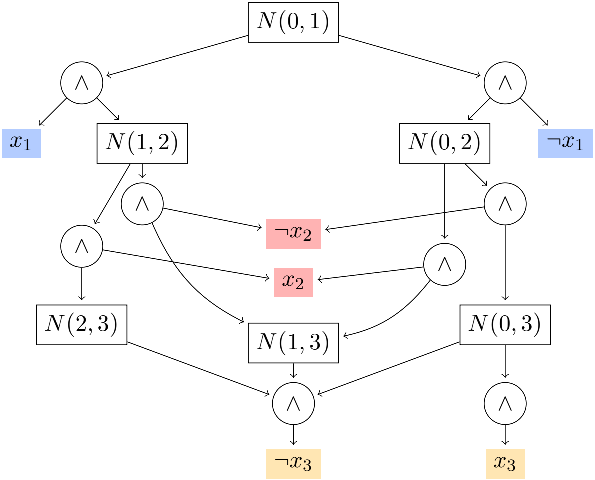
5000€



2500€

2500€

5000€



The size of DNNF circuit is in $\mathcal{O}(m \times B)$, where B is the budget limit.

Generalizing the Embedding to Add Dependencies

Dependencies between the projects: We add a dependency graph between the projects indicating that some projects can only be realized if some other also are.

Generalizing the Embedding to Add Dependencies

Dependencies between the projects: We add a dependency graph between the projects indicating that some projects can only be realized if some other also are.

We can adapt the previous embedding by keeping track of some of the decisions we already made. Note that we don't have to remember all of them.

Generalizing the Embedding to Add Dependencies

Dependencies between the projects: We add a dependency graph between the projects indicating that some projects can only be realized if some other also are.

We can adapt the previous embedding by keeping track of some of the decisions we already made. Note that we don't have to remember all of them.

We can encode this setting in a DNNF circuit of size $\mathcal{O}(m \times B \times 2^k)$, where k is the pathwidth of the dependency graph.

Generalizing the Embedding to Add Quotas

Quotas over types of projects: Projects are organized into types and we add quotas on the types. A quota indicates a lower and an upper bound on some measure (number of project selected, amount spent on the type, ...) for each type.

Generalizing the Embedding to Add Quotas

Quotas over types of projects: Projects are organized into types and we add quotas on the types. A quota indicates a lower and an upper bound on some measure (number of project selected, amount spent on the type, ...) for each type.

We can adapt the previous embedding by keeping track of the current status of the quotas.

Generalizing the Embedding to Add Quotas

Quotas over types of projects: Projects are organized into types and we add quotas on the types. A quota indicates a lower and an upper bound on some measure (number of project selected, amount spent on the type, ...) for each type.

We can adapt the previous embedding by keeping track of the current status of the quotas.

We can encode this setting in a DNNF circuit of size $\mathcal{O}(m \times B \times Q^k)$, where Q is the number of different values the quota can take and k is the pathwidth of the type overlap graph (equal to 1 when types are not overlapping).

Quick Summary

Using judgment aggregation we can:

- Reason about PB instances efficiently;
- Introduce additional resources to express the cost of projects;
- Easily introduce new constraints at the only cost of defining new embeddings.

What has been put under the rug:

- Dealing with exhaustiveness on the JA side;
- Assessing the quality of JA rules with regards to PB axioms.

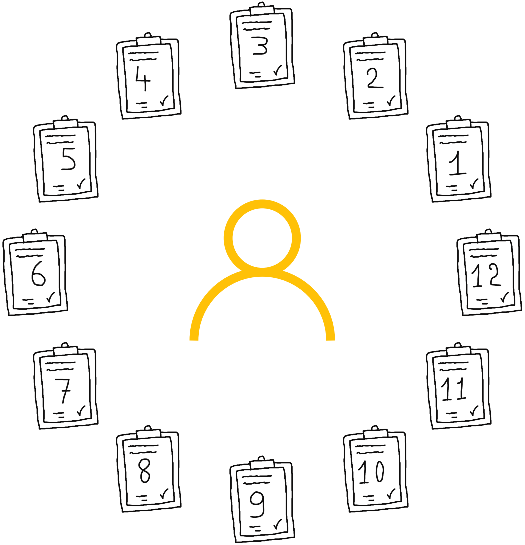
Simon Rey, Ulle Endriss and Ronald de Haan, *Designing Participatory Budgeting Mechanisms Grounded in Judgment Aggregation*, KR 2020.

3. End-to-End Model for Participatory Budgeting

Awareness Set

Awareness Set

For a specific participatory budgeting instance, a gigantic number of projects are conceivable.

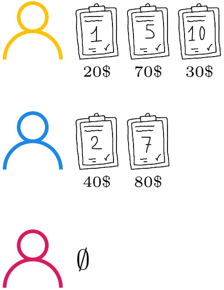


Awareness Set

For a specific participatory budgeting instance, a gigantic number of projects are conceivable.

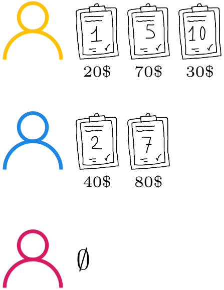
Because agents have bounded rationality, they are only able to conceive of a finite subset of these, their *awareness set*.

The Full Model

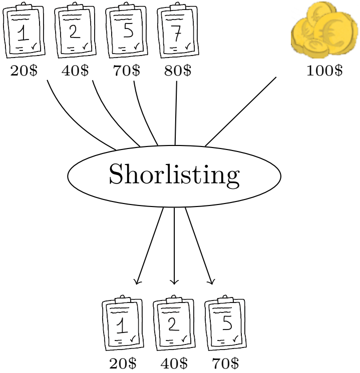


Agents come with their awareness sets

The Full Model

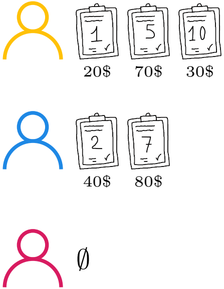


Agents come with their awareness sets

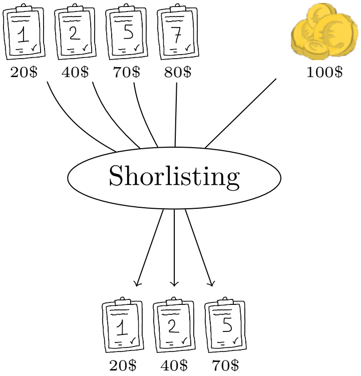


They submit subsets of their awareness sets which are then shortlisted by a *shortlisting rule*.

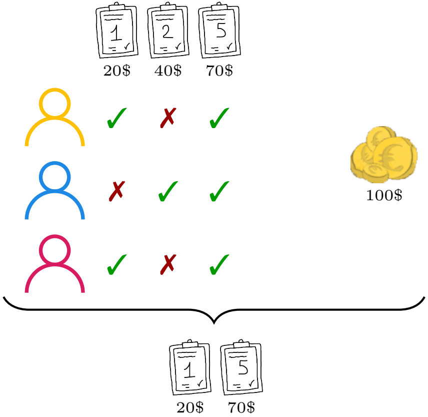
The Full Model



Agents come with their awareness sets



They submit subsets of their awareness sets which are then shortlisted by a *shortlisting rule*.



Using approval ballots, an *allocation rule* determines the final budget allocation.

End-to-End Model for Participatory Budgeting

Shortlisting Rules and Axioms

Selecting a Representative Shortlist

DEFINITION: k-EQUAL-REPRESENTATION SHORTLISTING RULE

For $k \in \mathbb{N}$, for all instances $I = \langle \mathbb{P}, c, B \rangle$, and all profiles $\boldsymbol{P} = (P_1, \ldots, P_n)$:

$$R(I, \boldsymbol{P}) = \underset{\substack{P \subseteq \bigcup \boldsymbol{P} \\ c(P) \leq kB}}{\arg\max} \sum_{P_i \in \boldsymbol{P}} \sum_{\ell=0}^{|P_i \cap P|} \frac{1}{n^{\ell}},$$

i.e., make sure all agents have at least one proposal selected before selecting a second one for some agent, and so on while satisfying the budget constraint.

Non-wastefulness

Representation efficiency

Computational Complexity

Selecting a Representative Shortlist

Definition: k-Equal-Representation Shortlisting Rule

For $k \in \mathbb{N}$, for all instances $I = \langle \mathbb{P}, c, B \rangle$, and all profiles $\boldsymbol{P} = (P_1, \ldots, P_n)$:

$$R(I, \boldsymbol{P}) = \underset{\substack{P \subseteq \bigcup \boldsymbol{P} \\ c(P) \leq kB}}{\arg\max} \sum_{P_i \in \boldsymbol{P}} \sum_{\ell=0}^{|P_i \cap P|} \frac{1}{n^\ell},$$

i.e., make sure all agents have at least one proposal selected before selecting a second one for some agent, and so on while satisfying the budget constraint.

Non-wastefulness

✓

$k \geq 2$

Representation efficiency

Computational Complexity

Selecting a Representative Shortlist

Definition: k-Equal-Representation Shortlisting Rule

For $k \in \mathbb{N}$, for all instances $I = \langle \mathbb{P}, c, B \rangle$, and all profiles $\boldsymbol{P} = (P_1, \ldots, P_n)$:

$$R(I, \boldsymbol{P}) = \underset{\substack{P \subseteq \bigcup \boldsymbol{P} \\ c(P) \leq kB}}{\arg\max} \sum_{P_i \in \boldsymbol{P}} \sum_{\ell=0}^{|P_i \cap P|} \frac{1}{n^\ell},$$

i.e., make sure all agents have at least one proposal selected before selecting a second one for some agent, and so on while satisfying the budget constraint.

Non-wastefulness	Representation efficiency	Computational Complexity
✓ $k \geq 2$	✓ $k \geq 1$	

Selecting a Representative Shortlist

Definition: k-Equal-Representation Shortlisting Rule

For $k \in \mathbb{N}$, for all instances $I = \langle \mathbb{P}, c, B \rangle$, and all profiles $\boldsymbol{P} = (P_1, \dots, P_n)$:

$$R(I, \boldsymbol{P}) = \arg\max_{\substack{P \subseteq \bigcup \boldsymbol{P} \\ c(P) \leq kB}} \sum_{P_i \in \boldsymbol{P}} \sum_{\ell=0}^{|P_i \cap P|} \frac{1}{n^\ell},$$

i.e., make sure all agents have at least one proposal selected before selecting a second one for some agent, and so on while satisfying the budget constraint.

Non-wastefulness	Representation efficiency	Computational Complexity
✓ $k \geq 2$	✓ $k \geq 1$	NP-hard

Avoiding Shortlisting Duplicates

Definition: k-Median Shortlisting Rule

Assume proposals are displayed on a metric space, with distance δ. The k-median shortlisting rule proceeds as follows:

- Gather the proposals into any number of clusters;
- Take the geometric median of each cluster to be its representative;
- Select the clustering that:
  - Minimizes the in-cluster distance (largest distance from a proposal to its representative);
  - Does not cost more than $k \times B$ (the total cost of the representatives is less than $k \times B$);
- Shortlist all the representatives.

Non-wastefulness   Representation efficiency   Computational Complexity

Avoiding Shortlisting Duplicates

Definition: k-Median Shortlisting Rule

Assume proposals are displayed on a metric space, with distance δ. The k-median shortlisting rule proceeds as follows:

- Gather the proposals into any number of clusters;
- Take the geometric median of each cluster to be its representative;
- Select the clustering that:
  - Minimizes the in-cluster distance (largest distance from a proposal to its representative);
  - Does not cost more than $k \times B$ (the total cost of the representatives is less than $k \times B$);
- Shortlist all the representatives.

Non-wastefulness

✓

$k \geq 2$

Representation efficiency

Computational Complexity

Avoiding Shortlisting Duplicates

Definition: k-Median Shortlisting Rule

Assume proposals are displayed on a metric space, with distance δ. The k-median shortlisting rule proceeds as follows:

- Gather the proposals into any number of clusters;
- Take the geometric median of each cluster to be its representative;
- Select the clustering that:
  - Minimizes the in-cluster distance (largest distance from a proposal to its representative);
  - Does not cost more than $k \times B$ (the total cost of the representatives is less than $k \times B$);
- Shortlist all the representatives.

Non-wastefulness	Representation efficiency	Computational Complexity
✓	✗	
$k \geq 2$	$\forall k \in \mathbb{N}$	

Avoiding Shortlisting Duplicates

Definition: k-Median Shortlisting Rule

Assume proposals are displayed on a metric space, with distance δ. The k-median shortlisting rule proceeds as follows:

- Gather the proposals into any number of clusters;
- Take the geometric median of each cluster to be its representative;
- Select the clustering that:
  - Minimizes the in-cluster distance (largest distance from a proposal to its representative);
  - Does not cost more than $k \times B$ (the total cost of the representatives is less than $k \times B$);
- Shortlist all the representatives.

Non-wastefulness	Representation efficiency	Computational Complexity
✓ $k \geq 2$	✗ $\forall k \in \mathbb{N}$	NP-hard For most distances

End-to-End Model for Participatory Budgeting

First-Stage Strategyproofness

Motivation

- Should I propose my fountain even though someone else also proposed one?
- Would it be beneficial for me to submit many proposals to dilute the votes?

Taxonomy of the First-Stage Strategyproofness Concepts

Shortlisting Stage

Taxonomy of the First-Stage Strategyproofness Concepts

I'm the MANIPULATOR!

Shortlisting Stage

Taxonomy of the First-Stage Strategyproofness Concepts

Shortlisting Stage

I'm the MANIPULATOR!

What information do I have?

Taxonomy of the First-Stage Strategyproofness Concepts

Restricted FSSP

I'm the MANIPULATOR!

What information do I have?

Shortlisting Stage

Only C_i, my awareness set

Taxonomy of the First-Stage Strategyproofness Concepts

Restricted FSSP

Unrestricted FSSP

I'm the MANIPULATOR!

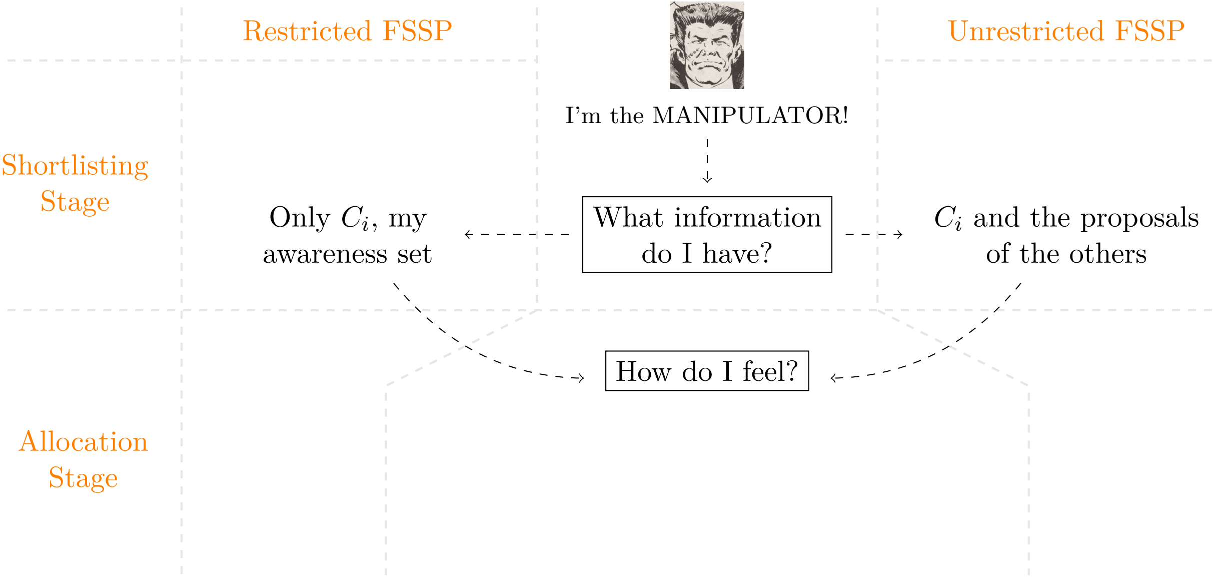
Shortlisting Stage

Only C_i, my awareness set

What information do I have?

C_i and the proposals of the others

Taxonomy of the First-Stage Strategyproofness Concepts



Restricted FSSP

Unrestricted FSSP

Shortlisting Stage

Allocation Stage

I'm the MANIPULATOR!

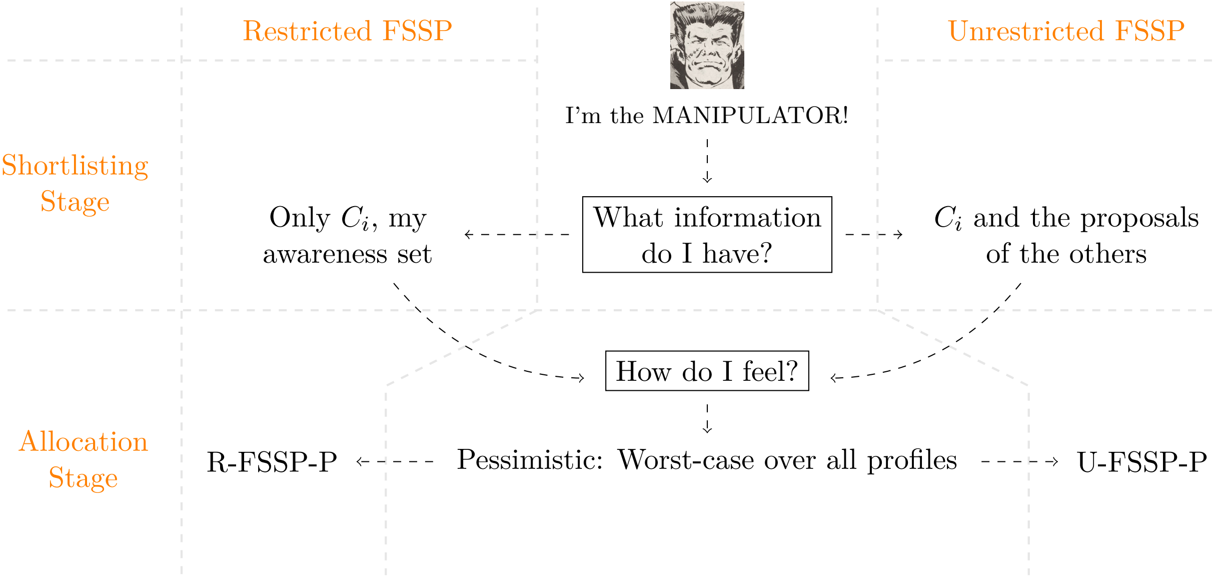
What information do I have?

Only C_i, my awareness set

C_i and the proposals of the others

How do I feel?

Taxonomy of the First-Stage Strategyproofness Concepts



Restricted FSSP

Unrestricted FSSP

Shortlisting Stage

Allocation Stage

I'm the MANIPULATOR!

What information do I have?

Only C_i, my awareness set

C_i and the proposals of the others

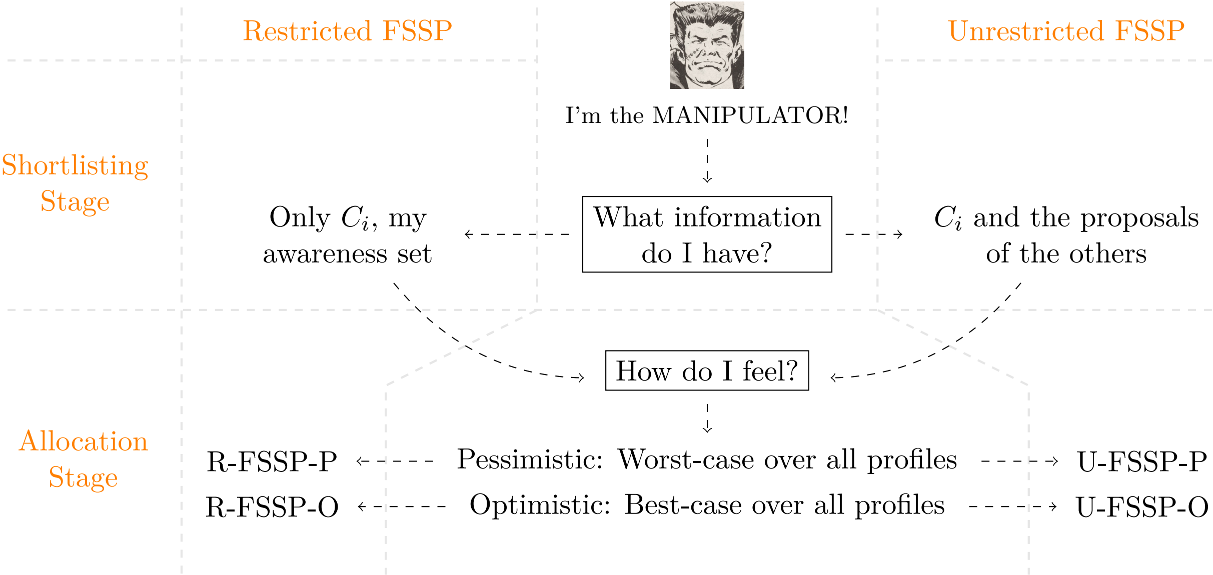
How do I feel?

Pessimistic: Worst-case over all profiles

R-FSSP-P

U-FSSP-P

Taxonomy of the First-Stage Strategyproofness Concepts



Restricted FSSP

Unrestricted FSSP

Shortlisting Stage

I'm the MANIPULATOR!

What information do I have?

Only C_i, my awareness set

C_i and the proposals of the others

Allocation Stage

How do I feel?

Pessimistic: Worst-case over all profiles

Optimistic: Best-case over all profiles

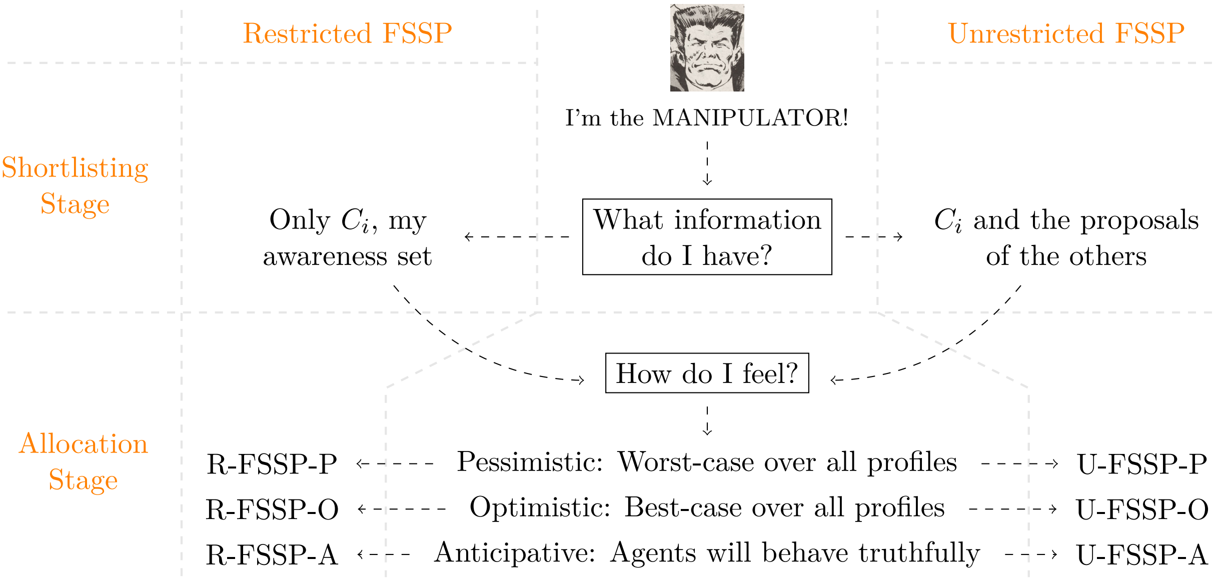
R-FSSP-P

R-FSSP-O

U-FSSP-P

U-FSSP-O

Taxonomy of the First-Stage Strategyproofness Concepts



Restricted FSSP

Unrestricted FSSP

Shortlisting Stage

I'm the MANIPULATOR!

What information do I have?

Only C_i, my awareness set

C_i and the proposals of the others

Allocation Stage

How do I feel?

R-FSSP-P ← Pessimistic: Worst-case over all profiles → U-FSSP-P

R-FSSP-O ← Optimistic: Best-case over all profiles → U-FSSP-O

R-FSSP-A ← Anticipative: Agents will behave truthfully → U-FSSP-A

Awareness-Restricted Manipulation

<u>Theorem</u>:

No pair $\langle R, F\rangle$ where R is non-wasteful and F is exhaustive can be R-FSSP-P, R-FSSP-O or R-FSSP-A.

SOON THE MOMENT WILL COME AT LAST TO STRIKE--
--AND THEN, NOT EVEN GOD CAN SAVE OUR GRACIOUS QUEEN--FROM THE POWER OF THE MANIPULATOR!!

Nomination Shortlisting Rule

THEOREM:

For every F that is exhaustive and strongly unanimous, the pair $\langle R, F \rangle$, where R is the nomination shortlisting rule, is U-FSSP-P.

MANIPULATOR! YOU AND THE HIGHWAYMAN-- YOU'VE BOTH HAD IT!
IT'S TIME I PUT PAID TO YOUR MAD DREAM OF CONQUEST!
TALK IS CHEAP, BRITAIN! WHAT COUNTS IS ACTION!

Exhaustive rule: The budget allocations returned by the rule must be maximal with respect to cost.

Unanimous rule: Whenever all agents submit the same feasible ballot A, then A should be returned by the rule.

Strongly unanimous rule: Whenever all agents but one submit the same feasible ballot A, then A should be returned by the rule.

The Quick Summary Counter-Attacks

Overall, we have:

- Presented an end-to-end model for participatory budgeting;

The Quick Summary Counter-Attacks

Overall, we have:

- Presented an end-to-end model for participatory budgeting;
- Studied the shortlisting stage by defining some shortlisting rules and studying their properties;

The Quick Summary Counter-Attacks

Overall, we have:

- Presented an end-to-end model for participatory budgeting;
- Studied the shortlisting stage by defining some shortlisting rules and studying their properties;
- Investigated the strategic interactions between the two stages.

The Quick Summary Counter-Attacks

Overall, we have:

- Presented an end-to-end model for participatory budgeting;
- Studied the shortlisting stage by defining some shortlisting rules and studying their properties;
- Investigated the strategic interactions between the two stages.

Simon Rey, Ulle Endriss and Ronald de Haan, *Shortlisting Rules and Incentives in an End-to-End Model for Participatory Budgeting*, IJCAI 2021.

4. Long-Term Participatory Budgeting

Focusing on Types of Agents

One problem with sequential decisions is that the ballots for each round are anonymous, i.e., it is impossible to know what an agent approved of in previous rounds.

Focusing on Types of Agents

One problem with sequential decisions is that the ballots for each round are anonymous, i.e., it is impossible to know what an agent approved of in previous rounds.

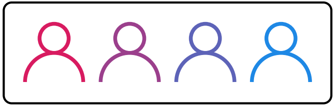
↳ One does not have access to “past satisfaction” of an agent when designing fairness criteria.

Focusing on Types of Agents

One problem with sequential decisions is that the ballots for each round are anonymous, i.e., it is impossible to know what an agent approved of in previous rounds.

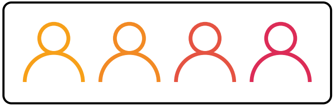
↳ One does not have access to “past satisfaction” of an agent when designing fairness criteria.

We overcome this by partitioning the agents into *types*, i.e., characteristics they share:



First district

Second district



Third district

The Model

	Year 1 🪙 = 10				Year 2 🪙 = 10				Year 3 🪙 = 10			
	1	2	3	4	5	6	7	8	9	10	11	12
Cost	6	2	2	4	5	5	3	2	7	7	$4/3$	$29/3$
Robot type	✓			✓		✓	✓	✓	✓		✓	
	✓		✓			✓		✓	✓		✓	
	✓			✓		✓	✓	✓	✓		✓	
Animal type		✓		✓	✓		✓	✓		✓	✓	✓
			✓	✓	✓		✓			✓		✓

Long-Term Participatory Budgeting

Welfare Measures

Satisfaction

Definition: Satisfaction

Let $\vec{\pi} = (\pi_1, \ldots, \pi_k)$ be a solution. The satisfaction of a type t for round j is given by:

$$sat_j(\pi_j, t) = \sum_{1 \leq j' \leq j} \frac{1}{|t|} \sum_{i \in t} c(\pi_j \cap A_j(i)).$$

Relative Satisfaction

One potential drawback of satisfaction is that approving less projects can yield to lower satisfaction. To tackle this issue, we introduce relative satisfaction.

Definition: Relative Satisfaction

Let $\vec{\pi} = (\pi_1, \ldots, \pi_k)$ be a solution. The relative satisfaction of a type t for round j is given by:

$$sat_j(\pi_j, t) = \sum_{1 \leq j' \leq j} \frac{1}{|t|} \sum_{i \in t} \frac{c(\pi_j \cap A_j(i))}{\max\{c(A) \mid A \subseteq A_j(i) \text{ s.t. } A \text{ is feasible}\}}.$$

Share

Both satisfaction and relative satisfaction are utilitarian concepts. The last welfare measure we introduce—the share—is more distributive in the sense that it aims at spending an equal amount of resources on each type.

<ins>Definition</ins>: Share

Let $\vec{\pi} = (\pi_1, \ldots, \pi_k)$ be a solution. The share of a type t for round j is given by:

$$sat_j(\pi_j, t) = \sum_{1 \leq j' \leq j} \frac{1}{|t|} \sum_{i \in t} \sum_{p \in \pi_j \cap A_j(i)} \frac{c(p)}{|\{i' \mid p \in A_j(i')\}|}.$$

Long-Term Participatory Budgeting

Achieving Fairness

Perfect Fairness

<u>Definition</u>: Equal-F

For a welfare measure F, a solution $\vec{\pi}$ satisfies equal-F if for every two types t, t' and every round j, we have:

$$F(\vec{\pi}, t, j) = F(\vec{\pi}, t', j).$$

Perfect Fairness

DEFINITION: EQUAL-F

For a welfare measure F, a solution $\vec{\pi}$ satisfies equal-F if for every two types t, t' and every round j, we have:

$$F(\vec{\pi}, t, j) = F(\vec{\pi}, t', j).$$

PROPOSITION:

There are instance where equal-F cannot be satisfied for all our welfare measures F.

Perfect Fairness

Definition: Equal-F

For a welfare measure F, a solution $\vec{\pi}$ satisfies equal-F if for every two types t, t' and every round j, we have:

$$F(\vec{\pi}, t, j) = F(\vec{\pi}, t', j).$$

Proposition:

There are instance where equal-F cannot be satisfied for all our welfare measures F.

Proposition:

Checking whether a solution can be extended to the next round so that equal-F is satisfied is strongly `NP`-complete.

Optimizing for Fairness - Computational Complexity

DEFINITION: F-GINI

The Gini coefficient of an ordered vector $v = (v_1, \ldots, v_k) \in \mathbb{R}^k$ is given by:

$$gini(v) = 1 - \frac{\sum_{i=1}^{k} (2i - 1) v_i}{k \sum_{i=1}^{k} v_i}.$$

Optimizing for Fairness - Computational Complexity

DEFINITION: F-GINI

The Gini coefficient of an ordered vector $v = (v_1, \dots, v_k) \in \mathbb{R}^k$ is given by:

$$gini(v) = 1 - \frac{\sum_{i=1}^{k} (2i - 1) v_i}{k \sum_{i=1}^{k} v_i}.$$

For a welfare measure F, the F-Gini coefficient of a solution $\vec{\pi}$ at round j is the Gini coefficient of the ordered vector containing $F(\vec{\pi}, t, j)$ for all types t.

Optimizing for Fairness - Computational Complexity

Definition: F-Gini

The Gini coefficient of an ordered vector $v = (v_1, \ldots, v_k) \in \mathbb{R}^k$ is given by:

$$gini(v) = 1 - \frac{\sum_{i=1}^{k}(2i-1)v_i}{k\sum_{i=1}^{k} v_i}.$$

For a welfare measure F, the F-Gini coefficient of a solution $\vec{\pi}$ at round j is the Gini coefficient of the ordered vector containing $F(\vec{\pi}, t, j)$ for all types t.

A solution $\vec{\pi}$ is F-Gini-optimal at round j with respect to a set S of solutions, if there is no solution in S with a lower F-Gini coefficient than $\vec{\pi}$.

Optimizing for Fairness - Computational Complexity

Definition: F-Gini

The Gini coefficient of an ordered vector $v = (v_1, \ldots, v_k) \in \mathbb{R}^k$ is given by:

$$gini(v) = 1 - \frac{\sum_{i=1}^{k}(2i-1)v_i}{k\sum_{i=1}^{k} v_i}.$$

For a welfare measure F, the F-Gini coefficient of a solution $\vec{\pi}$ at round j is the Gini coefficient of the ordered vector containing $F(\vec{\pi}, t, j)$ for all types t.

A solution $\vec{\pi}$ is F-Gini-optimal at round j with respect to a set S of solutions, if there is no solution in S with a lower F-Gini coefficient than $\vec{\pi}$.

Proposition:

Checking whether a solution is F-Gini-optimal at a given round is weakly `co-NP`-complete for all of our welfare measures.

Fairness in the Long Run

Another idea is to require perfect fairness but only in the long run.

Fairness in the Long Run

Another idea is to require perfect fairness but only in the long run.

Definition: Convergence to equal-F

For a welfare measure F, an infinite solution $\vec{\pi}$ converges to equal-F if for every two types:

$$\frac{F(\vec{\pi}, t, j)}{F(\vec{\pi}, t', j)} \xrightarrow[j \to +\infty]{} 1.$$

Fairness in the Long Run

Another idea is to require perfect fairness but only in the long run.

Definition: Convergence to equal-F

For a welfare measure F, an infinite solution $\vec{\pi}$ converges to equal-F if for every two types:

$$\frac{F(\vec{\pi}, t, j)}{F(\vec{\pi}, t', j)} \xrightarrow[j \to +\infty]{} 1.$$

Since this criterion is about infinite sequences, we will not analyze its computational complexity.

Fairness in the Long Run - 2 agents

PROPOSITION:

When there are two agents, a solution converging towards equal-Satisfaction or equal-Share can always be found (under mild assumptions).

Fairness in the Long Run - 2 agents

PROPOSITION:

When there are two agents, a solution converging towards equal-Satisfaction or equal-Share can always be found (under mild assumptions).

However, this result does not extend to three agents.

PROPOSITION:

With three agents, there exists an instance for which no solution converges towards either equal-Satisfaction or equal-Share.

Fairness in the Long Run - Relative Satisfaction

The picture for relative satisfaction is much nicer.

THEOREM:

When there are two types and with non-empty knapsack ballots, a solution converging towards equal-Relative-Satisfaction can always be found (under mild assumptions).

Fairness in the Long Run - Relative Satisfaction

The picture for relative satisfaction is much nicer.

Theorem:

When there are two types and with non-empty knapsack ballots, a solution converging towards equal-Relative-Satisfaction can always be found (under mild assumptions).

Side note: The proof is constructive, however, the solution constructed solution might not be exhaustive. Assuming exhaustive ballots solves this issue.

Fairness in the Long Run - Relative Satisfaction

The picture for relative satisfaction is much nicer.

THEOREM:

When there are two types and with non-empty knapsack ballots, a solution converging towards equal-Relative-Satisfaction can always be found (under mild assumptions).

Side note: The proof is constructive, however, the solution constructed solution might not be exhaustive. Assuming exhaustive ballots solves this issue.

Open problem: Can this result be extended to three and more types?

The Return of the Quick Summary

Picking a Gini-opti. solution at round k

Satisfaction: ✓

Relative Satisfaction: ✓

Share: ✓

co-NP-complete

Achieving equal-F at round k

Satisfaction: ✗

Relative Satisfaction: ✗

Share: ✗

NP-complete

Coverging towards equal-F

Satisfaction: ✓$(n \leq 3)$

Rel. Satisfaction: ✓$(t \leq 2)$

Share: ✓$(n \leq 3)$

Martin Lackner, Jan Maly and Simon Rey, *Fairness in Long-Term Participatory Budgeting*, IJCAI 2021.

5. Conclusion

Main Summary

We have seen different models of participatory budgeting designed to capture more closely real-world PB processes:

- Allowing for additional constraints on the outcome;
- Capturing strategic interactions between the two stages of the process;
- Studying fairness concepts for repeated instances of participatory budgeting.

Thanks