Participatory Budgeting

Collective decision making on a budget

Martin Durand

April 20 2022

LIP6 - PhD Seminar Operations Research / Decision

Introduction

An introductive example

A laboratory director "FK" manages the budget of his lab throughout the year. At the beginning of December, he realises that 20 000 € were not spent as intended and are available for other projects.

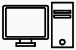
An introductive example

A laboratory director "FK" manages the budget of his lab throughout the year. At the beginning of December, he realises that 20 000 € were not spent as intended and are available for other projects.

Project	Cost	Utility

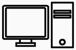
An introductive example

A laboratory director “FK” manages the budget of his lab throughout the year. At the beginning of December, he realises that 20 000 € were not spent as intended and are available for other projects.

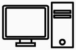
Project	Cost	Utility
	6000	3

An introductive example

A laboratory director “FK” manages the budget of his lab throughout the year. At the beginning of December, he realises that 20 000 € were not spent as intended and are available for other projects.

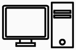
Project	Cost	Utility
	6000	3
	6600	6

A laboratory director “FK” manages the budget of his lab throughout the year. At the beginning of December, he realises that 20 000 € were not spent as intended and are available for other projects.

Project	Cost	Utility
	6000	3
	6600	6
	8000	8

An introductive example

A laboratory director “FK” manages the budget of his lab throughout the year. At the beginning of December, he realises that 20 000 € were not spent as intended and are available for other projects.

Project	Cost	Utility
	6000	3
	6600	6
	8000	8
	500	1

An introductive example

A laboratory director "FK" manages the budget of his lab throughout the year. At the beginning of December, he realises that 20 000 € were not spent as intended and are available for other projects.

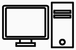
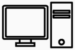
Project	Cost	Utility
	6000	3
	6600	6
	8000	8
	500	1
	18000	10

Table 1: Projects available

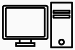
An introductive example - Step 1 : Knapsack

The goal is to select a subset of projects that fits in the budget and maximizes the utility.

Project	Cost	Utility
	6000	3
	6600	6
	8000	8
	500	1
	18000	10

Solution	Cost	Feasible	Utility

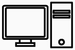
The goal is to select a subset of projects that fits in the budget and maximizes the utility.

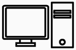
Project	Cost	Utility
	6000	3
	6600	6
	8000	8
	500	1
	18000	10

Solution	Cost	Feasible	Utility
 + 	26000	✗	18

An introductive example - Step 1 : Knapsack

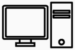
The goal is to select a subset of projects that fits in the budget and maximizes the utility.

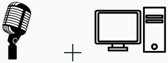
Project	Cost	Utility
	6000	3
	6600	6
	8000	8
	500	1
	18000	10

Solution	Cost	Feasible	Utility
 + 	26000	✗	18
 + 	14000	✓	11

An introductive example - Step 1 : Knapsack

The goal is to select a subset of projects that fits in the budget and maximizes the utility.

Project	Cost	Utility
	6000	3
	6600	6
	8000	8
	500	1
	18000	10

Solution	Cost	Feasible	Utility
	26000	✗	18
	14000	✓	11
	15100	✓	15

The power of democracy ?

Since FK is a very nice laboratory director, he thinks that he should not take the decision alone and, since this impacts the whole lab, he decides to ask all the members how the budget should be spent.

Since FK is a very nice laboratory director, he thinks that he should not take the decision alone and, since this impacts the whole lab, he decides to ask all the members how the budget should be spent.

How ?

Participatory Budgeting - Goal

- An organization (company, city, country $\cdots$) consults its members on how a part of its budget should be spent.
- The members and the organization suggest different projects, each having a specific cost.
- Once all the projects are known, the members vote.
- The organization selects projects according to the votes

Participatory Budgeting - Goal

- An organization (company, city, country $\cdots$) consults its members on how a part of its budget should be spent.
- The members and the organization suggest different projects, each having a specific cost.
- Once all the projects are known, the members vote.
- The organization selects projects according to the votes

Two major questions :

- How do the members express their preferences ?
- Once we have the votes, how do we select the winning projects ?

Outline

1. Gathering preferences

   Particular case I - Knapsack

   Particular case II - Multiwinner voting

2. Principles and methods of resolution

   Maximizing satisfaction

   Axioms

3. Perspectives

Gathering preferences

Particuliar case I : 1 voter - Knapsack

Knapsack problem

Data :

- a set $P = \{p_1, \cdots, p_n\}$ of n projects, project p_i has a cost c_i.
- an utility function $u : P \to \mathbb{R}^+$
- an integer K (size of the knapsack)

Objective : Return a subset S of projects in P such that $\sum_{p_i \in S} c_i \leq K$ and that $\sum_{p_i \in S} u(p_i)$ is maximized.

Particuliar case I : 1 voter - Knapsack

Knapsack problem

Data :

- a set $P = \{p_1, \cdots, p_n\}$ of n projects, project p_i has a cost c_i.
- an utility function $u : P \rightarrow \mathbb{R}^+$
- an integer K (size of the knapsack)

Objective : Return a subset S of projects in P such that $\sum_{p_i \in S} c_i \leq K$ and that $\sum_{p_i \in S} u(p_i)$ is maximized.

Proposition

The KNAPSACK *problem is NP-hard. [4]*

Particuliar case I : 1 voter - Knapsack

Knapsack problem

Data :

- a set $P = \{p_1, \cdots, p_n\}$ of n projects, project p_i has a cost c_i.
- an utility function $u : P \to \mathbb{R}^+$
- an integer K (size of the knapsack)

Objective : Return a subset S of projects in P such that $\sum_{p_i \in S} c_i \leq K$ and that $\sum_{p_i \in S} u(p_i)$ is maximized.

Proposition

The KNAPSACK *problem is NP-hard. [4]*

It can be solved exactly efficiently with Branch-and-Bound or dynamic programming algorithms.
There is also a FPTAS (Fully Polynomial Time Approximation Scheme) that works very well in practice.

Knapsack/Participatory Budgeting

Knapsack problem

Data :

- a set $P = \{p_1, \cdots, p_n\}$ of n projects, project p_i has a cost c_i.
- an utility function $u : P \to \mathbb{R}^+$
- an integer K (size of the knapsack)

Objective : Return a subset S of projects in P such that $\sum_{p_i \in S} c_i \leq K$ and that $\sum_{p_i \in S} u(p_i)$ is maximized.

Knapsack/Participatory Budgeting

Knapsack problem

Data :

- a set $P = \{p_1, \cdots, p_n\}$ of n projects, project p_i has a cost c_i.
- an utility function $u : P \to \mathbb{R}^+$
- an integer K (size of the knapsack)

Objective : Return a subset S of projects in P such that $\sum_{p_i \in S} c_i \leq K$ and that $\sum_{p_i \in S} u(p_i)$ is maximized.

Participatory budgeting problem

Data :

- a set $P = \{p_1, \cdots, p_n\}$ of n projects, project p_i has a cost c_i.
- the preferences of the voters
- an integer K (budget)

Objective : Return a subset S of projects in P satisfying all the voters as much as possible

Example - Step 2 : Utility

Gathering utilities :

- Ask the voters to give their utility (a score) for each project
- Sum up the utilities to get a global utility function u

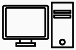
Project	Cost	u_1	u_2	u_3	u
	6000	3	9	7	19
	6600	6	8	0	14
	8000	8	10	6	24
	500	1	1	1	3
	18000	10	0	8	18

Table 2: Projects available - Voters give a score to each project

Utility - Problems

There are a few issues with the vote by utility :

- Giving a score to each project can be tedious
- It is easy to manipulate
- We still have to solve a NP-hard problem

Scoring vote seems to be suited only for instances with a low number of projects.

If all the projects have the same cost, the problem is much simpler.

- We know how many projects we can select
- Each project can be exchanged with another
- This is a polynomial case of the KNAPSACK problem

Particuliar case II : unitary costs - Multiwinner voting

If all the projects have the same cost, the problem is much simpler.

- We know how many projects we can select
- Each project can be exchanged with another
- This is a polynomial case of the KNAPSACK problem

It is also a known problem in computational social choice : selecting a set of k candidates according to a population's preferences : the multiwinner voting problem [3]. There are two main ways of gathering preferences for that problem :

- Rankings
- Approval votes

Particuliar case II : unitary costs - Multiwinner voting

If all the projects have the same cost, the problem is much simpler.

- We know how many projects we can select
- Each project can be exchanged with another
- This is a polynomial case of the KNAPSACK problem

It is also a known problem in computational social choice : selecting a set of k candidates according to a population's preferences : the multiwinner voting problem [3]. There are two main ways of gathering preferences for that problem :

- Rankings
- Approval votes

Multiwinner voting - Approval voting

Each voter indicates if they approve (vote 1) or disapprove (vote 0) each candidate.

We can see approval votes as a specific case of scoring votes :

- Simpler to understand and use
- The binary scale reduces the ability to manipulate
- We have less information than with a scoring vote but still enough to find interesting resolution principles

From now on, we will focus on rules based on approval votes.

Principles and methods of resolution

Aggregation rules

Participatory budgeting problem

Data :

- a set $P = \{p_1, \cdots, p_n\}$ of n projects, project p_i has a cost c_i.
- a set $A = \{V_1, \cdots, V_v\}$ of v preferences, V_i is the subset of projects approved by the voter i.
- an integer K

Objective : Return a subset S of projects in P satisfying all the voters as much as possible

An aggregation rule r is a mapping between a set of preferences over P and a feasible solution.

Aggregation rules

Participatory budgeting problem

Data :

- a set $P = \{p_1, \cdots, p_n\}$ of n projects, project p_i has a cost c_i.
- a set $A = \{V_1, \cdots, V_v\}$ of v preferences, V_i is the subset of projects approved by the voter i.
- an integer K

Objective : Return a subset S of projects in P satisfying all the voters as much as possible

An aggregation rule r is a mapping between a set of preferences over P and a feasible solution.
What does "satisfying all the voters as much as possible" means ?

- Formulate satisfaction as an optimization function
- Find desirable properties that aggregation rules should follow and chose an aggregation rule accordingly

Example - Step 3 : Approbation score

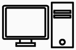
Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	*Score*
	6000	0	0	1	0	1	0	2
	6600	0	1	0	1	0	1	3
	8000	1	1	0	1	0	1	4
	500	1	0	1	1	1	1	5
	18000	0	0	1	0	1	0	2

Table 3: Approval votes

Maximizing the sum of approbation scores is equivalent to solving a KNAPSACK problem. We can also imagine a heuristic picking the items in decreasing order of their ratio *approbation*/*cost*.

Axioms - Interest

Axioms are desirable mathematical properties that the aggregation rules can follow or not. They do not describe methods to solve the problem but they indicate if a method is satisfactory or not.
These properties can ensure equity between the candidates, resistance to manipulation, robustness of the solution, consistency between the votes and the solution, non-dictatorship $\cdots$

Some combinations of axioms are impossible to fulfill entirely. Some combinations can characterise a set of rules, sometimes exactly one rule. Each combination of axioms matches a specific situation, we may want to fulfill them in some cases and to avoid them in others.

A few axioms for the participatory budgeting

Anonimity [2] Votes all have the exact same importance for the rule

Discount monotonicity [7] If a project is selected by the rule, reducing its cost does not make it unselected

Preference monotonicity [2] If a project is selected by the rule, improving its position in the preference of a voter (making it approved by the voter) does not make it unselected

Exhaustiveness [1] The rule does not leave unused budget, if a project can fit in the remaining budget, we have to select it

Axioms - Maximizing the approbation score

Does the rule which returns the subset of items maximizing the approbation score fulfill these axioms ?

- Anonimity [2] Votes all have the exact same importance for the rule
- Discount monotonicity [7] If a project is selected by the rule, reducing its cost does not make it unselected
- Preference monotonicity [2] If a project is selected by the rule, improving its position in the preference of a voter (making it approved by the voter) does not make it unselected
- Exhaustiveness [1] The rule does not leave unused budget, if a project can fit in the remaining budget, we have to select it

Axioms - Maximizing the approbation score

Does the rule which returns the subset of items maximizing the approbation score fulfill these axioms ?

- ✓ Anonimity [2] Votes all have the exact same importance for the rule
- Discount monotonicity [7] If a project is selected by the rule, reducing its cost does not make it unselected
- Preference monotonicity [2] If a project is selected by the rule, improving its position in the preference of a voter (making it approved by the voter) does not make it unselected
- Exhaustiveness [1] The rule does not leave unused budget, if a project can fit in the remaining budget, we have to select it

Axioms - Maximizing the approbation score

Does the rule which returns the subset of items maximizing the approbation score fulfill these axioms ?

- ✓ Anonimity [2] Votes all have the exact same importance for the rule
- ✓ Discount monotonicity [7] If a project is selected by the rule, reducing its cost does not make it unselected
- Preference monotonicity [2] If a project is selected by the rule, improving its position in the preference of a voter (making it approved by the voter) does not make it unselected
- Exhaustiveness [1] The rule does not leave unused budget, if a project can fit in the remaining budget, we have to select it

Axioms - Maximizing the approbation score

Does the rule which returns the subset of items maximizing the approbation score fulfill these axioms ?

- ✓ Anonimity [2] Votes all have the exact same importance for the rule
- ✓ Discount monotonicity [7] If a project is selected by the rule, reducing its cost does not make it unselected
- ✓ Preference monotonicity [2] If a project is selected by the rule, improving its position in the preference of a voter (making it approved by the voter) does not make it unselected
- Exhaustiveness [1] The rule does not leave unused budget, if a project can fit in the remaining budget, we have to select it

Axioms - Maximizing the approbation score

Does the rule which returns the subset of items maximizing the approbation score fulfill these axioms ?

- ✓ Anonimity [2] Votes all have the exact same importance for the rule
- ✓ Discount monotonicity [7] If a project is selected by the rule, reducing its cost does not make it unselected
- ✓ Preference monotonicity [2] If a project is selected by the rule, improving its position in the preference of a voter (making it approved by the voter) does not make it unselected
- ✓ Exhaustiveness [1] The rule does not leave unused budget, if a project can fit in the remaining budget, we have to select it

Axioms - Maximizing the approbation score

Does the rule which returns the subset of items maximizing the approbation score fulfill these axioms ?

- ✓ Anonimity [2] Votes all have the exact same importance for the rule
- ✓ Discount monotonicity [7] If a project is selected by the rule, reducing its cost does not make it unselected
- ✓ Preference monotonicity [2] If a project is selected by the rule, improving its position in the preference of a voter (making it approved by the voter) does not make it unselected
- ✓ Exhaustiveness [1] The rule does not leave unused budget, if a project can fit in the remaining budget, we have to select it

Is it a good rule then ?

The limit of utility (and approval) maximization

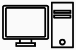
Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	*Score*
	6000	0	0	1	0	1	0	2
	6600	0	1	0	1	0	1	3
	8000	1	1	0	1	0	1	4
	500	1	0	1	1	1	1	5
	18000	0	0	1	0	1	0	2

Table 4: Approval maximization

The limit of utility (and approval) maximization

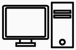
Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	*Score*
	6000	0	0	1	0	1	0	2
	6600	0	1	0	1	0	1	3
	8000	1	1	0	1	0	1	4
	500	1	0	1	1	1	1	5
	18000	0	0	1	0	1	0	2

Table 4: Approval maximization

The limit of utility (and approval) maximization

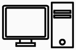
Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	*Score*
	6000	0	0	1	0	1	0	2
	6600	0	1	0	1	0	1	3
	8000	1	1	0	1	0	1	4
	500	1	0	1	1	1	1	5
	18000	0	0	1	0	1	0	2

Table 4: Approval maximization

The limit of utility (and approval) maximization

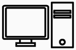
Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	*Score*
	6000	0	0	1	0	1	0	2
	6600	0	1	0	1	0	1	3
	8000	1	1	0	1	0	1	4
	500	1	0	1	1	1	1	5
	18000	0	0	1	0	1	0	2
		8500	14600	500	15100	500	15100	

Table 4: Approval maximization

Participatory Budgeting and proportionality

- Several axioms describing proportionality [1]
- The idea is that if $x\%$ of voters agree on a subset of projects taking $x\%$ of the budget, then they should be satisfied at least as much as if this subset was chosen.

Example

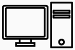
Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	$Score$
	6000	0	0	1	0	1	0	2
	500	1	0	1	1	1	1	4
	18000	0	0	1	0	1	0	2

Voters 3 and 5 represent 1/3 of the voters and agree on selecting the computers, which cost 6000€ ($< 1/3$ of the budget). The items they approve and that are selected should represent at least 1/3 of the budget.

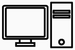
A proportional rule - Phragmen's rule

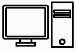
1. We start with an empty set $S = \emptyset$
2. For each project, we split its cost among all the voters that approved it (e.g. if an item costs 5000€ and 10 voters approved it, the cost for each voter would be 500€)
3. For each available project, we measure the effect of adding it to S by computing the maximum cost among the voters after its addition
4. We add the project with the lower maximum cost among the voters
5. Go back to step 3 until there is no more project fitting in the remaining budget.

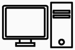
Phragmen's rule - Example : Iteration 1

Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	Cost/a
	6000	0	0	1	0	1	0	3000
	6600	0	1	0	1	0	1	2200
	8000	1	1	0	1	0	1	2000
	500	1	0	1	1	1	1	100
	18000	0	0	1	0	1	0	9000
		0	0	0	0	0	0	

$S = \{\}$

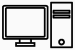
Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	Cost/a
	6000	0	0	1	0	1	0	3000
	6600	0	1	0	1	0	1	2200
	8000	1	1	0	1	0	1	2000
	500	1	0	1	1	1	1	100
	18000	0	0	1	0	1	0	9000
		0	0	3000	0	3000	0	

$S = \{$  $\}$

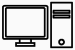
Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	Cost/a
	6000	0	0	1	0	1	0	3000
	6600	0	1	0	1	0	1	2200
	8000	1	1	0	1	0	1	2000
	500	1	0	1	1	1	1	100
	18000	0	0	1	0	1	0	9000
		0	2200	0	2200	0	2200	

$S = \{$  $\}$

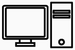
Phragmen's rule - Example : Iteration 1

Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	Cost/a
	6000	0	0	1	0	1	0	3000
	6600	0	1	0	1	0	1	2200
	8000	1	1	0	1	0	1	2000
	500	1	0	1	1	1	1	100
	18000	0	0	1	0	1	0	9000
		2000	2000	0	2000	0	2000	

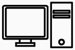
$S = \{$  $\}$

Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	Cost/a
	6000	0	0	1	0	1	0	3000
	6600	0	1	0	1	0	1	2200
	8000	1	1	0	1	0	1	2000
	500	1	0	1	1	1	1	100
	18000	0	0	1	0	1	0	9000
		100	0	100	100	100	100	

$S = \{$  $\}$

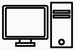
Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	Cost/a
	6000	0	0	1	0	1	0	3000
	6600	0	1	0	1	0	1	2200
	8000	1	1	0	1	0	1	2000
	500	1	0	1	1	1	1	100
	18000	0	0	1	0	1	0	9000
		0	0	9000	0	9000	0	

$S = \{$  $\}$

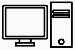
Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	Cost/a
	6000	0	0	1	0	1	0	3000
	6600	0	1	0	1	0	1	2200
	8000	1	1	0	1	0	1	2000
	500	1	0	1	1	1	1	100
	18000	0	0	1	0	1	0	9000
		100	0	100	100	100	100	

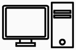
$S = \{$  $\}$

Phragmen's rule - Example : Iteration 2

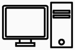
Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	Cost/a
	6000	0	0	1	0	1	0	3000
	6600	0	1	0	1	0	1	2200
	8000	1	1	0	1	0	1	2000
	500	1	0	1	1	1	1	100
	18000	0	0	1	0	1	0	9000
		100	0	100	100	100	100	

$S = \{$  $\}$

Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	Cost/a
	6000	0	0	1	0	1	0	3000
	6600	0	1	0	1	0	1	2200
	8000	1	1	0	1	0	1	2000
	500	1	0	1	1	1	1	100
	18000	0	0	1	0	1	0	9000
		100	0	3100	100	3100	100	

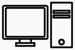
$S = \{$  $+$  $\}$

Phragmen's rule - Example : Iteration 2

Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	Cost/a
	6000	0	0	1	0	1	0	3000
	6600	0	1	0	1	0	1	2200
	8000	1	1	0	1	0	1	2000
	500	1	0	1	1	1	1	100
	18000	0	0	1	0	1	0	9000
		100	2200	100	2300	100	2300	

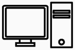
$S = \{$  $+$  $\}$

Phragmen's rule - Example : Iteration 2

Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	Cost/a
	6000	0	0	1	0	1	0	3000
	6600	0	1	0	1	0	1	2200
	8000	1	1	0	1	0	1	2000
	500	1	0	1	1	1	1	100
	18000	0	0	1	0	1	0	9000
		2100	2000	100	2100	100	2100	

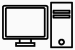
$S = \{$  $+$  $\}$

Phragmen's rule - Example : Iteration 2

Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	Cost/a
	6000	0	0	1	0	1	0	3000
	6600	0	1	0	1	0	1	2200
	8000	1	1	0	1	0	1	2000
	500	1	0	1	1	1	1	100
	18000	0	0	1	0	1	0	9000
		100	0	9100	100	9100	100	

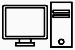
$S = \{$  $+$  $\}$

Phragmen's rule - Example : Iteration 2

Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	Cost/a
	6000	0	0	1	0	1	0	3000
	6600	0	1	0	1	0	1	2200
	8000	1	1	0	1	0	1	2000
	500	1	0	1	1	1	1	100
	18000	0	0	1	0	1	0	9000
		2100	2000	100	2100	100	2100	

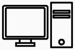
$S = \{$  $+$  $\}$

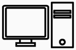
Phragmen's rule - Example : Iteration 3

Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	Cost/a
	6000	0	0	1	0	1	0	3000
	6600	0	1	0	1	0	1	2200
	8000	1	1	0	1	0	1	2000
	500	1	0	1	1	1	1	100
	18000	0	0	1	0	1	0	9000
		2100	2000	100	2100	100	2100	

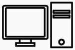
$S = \{$  $+$  $\}$

Phragmen's rule - Example : Iteration 3

Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	Cost/a
	6000	0	0	1	0	1	0	3000
	6600	0	1	0	1	0	1	2200
	8000	1	1	0	1	0	1	2000
	500	1	0	1	1	1	1	100
	18000	0	0	1	0	1	0	9000
		2100	2000	3100	2100	3100	2100	

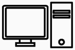
$S = \{$  $+$  $+$  $\}$

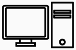
Phragmen's rule - Example : Iteration 3

Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	Cost/a
	6000	0	0	1	0	1	0	3000
	6600	0	1	0	1	0	1	2200
	8000	1	1	0	1	0	1	2000
	500	1	0	1	1	1	1	100
	18000	0	0	1	0	1	0	9000
		4300	4200	100	4300	100	4300	

$S = \{$  $+$  $+$  $\}$

Phragmen's rule - Example : Iteration 3

Project	Cost	V_1	V_2	V_3	V_4	V_5	V_6	Cost/a
	6000	0	0	1	0	1	0	3000
	6600	0	1	0	1	0	1	2200
	8000	1	1	0	1	0	1	2000
	500	1	0	1	1	1	1	100
	18000	0	0	1	0	1	0	9000
		2100	2000	3100	2100	3100	2100	

$S = \{$  $+$  $+$  $\}$

Approbation maximization / Phragmen's rule

Axiom	App. Maximization	Phragmen's rule
Anonimity	✓	✓
Discount monotonicity	✓	✓
Preference monotonicity	✓	✓
Exhaustiveness	✓	✓
Proportionality	✗	✓

Approbation maximization / Phragmen's rule

Axiom	App. Maximization	Phragmen's rule
Anonimity	✓	✓
Discount monotonicity	✓	✓
Preference monotonicity	✓	✓
Exhaustiveness	✓	✓
Proportionality	✗	✓

Does this mean that Phragmen's rule is better than the approbation maximization rule ?

Conclusion

- Participatory budgeting : direct democracy to determine how a budget is used
- Gathering the preferences : rankings, score, approval votes
- Selecting the solution : optimization functions, axioms

What's interesting with participatory budgeting ?

- Used in real-life and large-scale situations

Conclusion

- Participatory budgeting : direct democracy to determine how a budget is used
- Gathering the preferences : rankings, score, approval votes
- Selecting the solution : optimization functions, axioms

What's interesting with participatory budgeting ?

- Used in real-life and large-scale situations
- Projects have different costs
- The solution has a more complex structure than traditional collective decision problems

Perspectives

An issue with current methods

We used additive utilities during the whole presentation. We supposed that the utility of a subset $\{A, B\}$ was equal to the utility of A plus the utility of B, which can be true in some cases but not all.

Example

For a city, two similar, yet distinct, projects can emerge from the citizens' suggestions. For example, the construction of two pools in two close districts. These two project could be chosen together, they are enot mutually exclusive, however building the second pool, once the first is built, is not as useful.

An issue with current methods

We used additive utilities during the whole presentation. We supposed that the utility of a subset $\{A, B\}$ was equal to the utility of A plus the utility of B, which can be true in some cases but not all.

Example

For a city, two similar, yet distinct, projects can emerge from the citizens' suggestions. For example, the construction of two pools in two close districts. These two project could be chosen together, they are enot mutually exclusive, however building the second pool, once the first is built, is not as useful.

- Most of the papers on Participatory Budgeting use additive utilities directly or as an underlying assumption.
- Some projects have synergies between them or antisynergies, a recent article [6] tackle the issue but it is still an open question : how do we handle these synergies ?

What do you do ?

I am currently focusing on :

- Determining aggregation rules that take the project interactions into account
- Infering these interactions from the votes
- Finding new axioms relative to these interactions
- A new(ish) way of expressing preferences: "Knapsack voting" [5] (each voter gives his favorite feasible solution)

Thank you for your attention !

H. Aziz, B. Lee, and N. Talmon.
Proportionally representative participatory budgeting: Axioms and algorithms.
arXiv preprint arXiv:1711.08226, 2017.

F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia.
Handbook of computational social choice.
Cambridge University Press, 2016.

P. Faliszewski, P. Skowron, A. Slinko, and N. Talmon.
Multiwinner voting: A new challenge for social choice theory.
Trends in computational social choice, 74(2017):27–47, 2017.

M. R. Garey and D. S. Johnson.
Computers and intractability, volume 174.
freeman San Francisco, 1979.

- A. Goel, A. K. Krishnaswamy, S. Sakshuwong, and T. Aitamurto.
Knapsack voting for participatory budgeting.
ACM Transactions on Economics and Computation (TEAC), 7(2):1–27, 2019.

- P. Jain, K. Sornat, and N. Talmon.
Participatory budgeting with project interactions.
In *IJCAI*, pages 386–392, 2020.

- N. Talmon and P. Faliszewski.
A framework for approval-based budgeting methods.
In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2181–2188, 2019.