

# Budget Participatif: Implémentation de Rule X

Alban Guerbois

Université Paris-Dauphine  
M2 MODO. Atelier IA/Décision. Référent: Jérôme Lang  
albanguerbois@gmail.com

## Abstract

Nous proposons une implémentation de l'algorithme Rule X [Peters et al. 2020] en Python sur Jupyter Notebook. Nous détaillons le pseudo-code de l'algorithme Rule X dans la section 2.4 et présentons dans la section 3 les résultats d'un cas fictif de budget participatif mené auprès de nos camarades de master.

## 1 Budget participatif

Un **budget participatif** est un système démocratique d'allocations de ressources par lequel une organisation disposant d'un budget, finance des projets sélectionnés à l'issue d'un processus de votes.

Cette sélection étant d'habitude réservée aux dirigeants en place, l'avantage principal du budget participatif est d'inclure l'ensemble des membres de l'organisation dans le processus décisionnel.

Ce mécanisme a été inventé dans les années 1990 par le maire de Porto Alegre au Brésil. Il a depuis été adopté par un grand nombre d'états dans le monde, notamment la ville de Paris.

En 2019, la mairie de Paris disposait de 100M€ de fonds pour son budget participatif. Ainsi, à l'issue du vote de près de 230 000 personnes, 200 lauréats ont été sélectionnés parmi plus de 2000 projets candidats.

L'algorithme de sélection des projets utilisé dans la plupart des cas fonctionne de la manière suivante. Tant qu'il reste suffisamment de budget dans l'enveloppe, on sélectionne le projet avec le plus de votes.

Bien qu'intuitif, un tel algorithme présente de nombreux défauts. Ces limitations ont poussé Peters, Skowron et al. à développer une nouvelle méthode de sélection nommée Rule X.

## 2 Rule X

### 2.1 Fonctionnement

Chaque votant dispose d'un ensemble de projets pour lesquels il a voté. Chaque projet est associé à un cout. Par soucis de simplicité on normalise le budget à 1, le cout d'un projet est ainsi obtenu en le divisant par le budget.

Supposons que chaque votant dispose d'une fraction du budget, un projet est dit  **$\rho$ -abordable** si il peut-être financé intégralement (et exactement) par des membres ayant voté pour lui, de telle sorte que chacun de ces membres dépense un montant identique  $\rho$  (ou à défaut son budget restant).

L'idée de l'algorithme est de construire séquentiellement l'ensemble des projets retenus pour le budget participatif. Au départ, on donne à chaque votant une fraction égale du budget. On ajoute tant que possible le projet (non encore sélectionné)  $\rho$ -abordable pour  $\rho$  minimum en mettant à jour les budget restants de chaque votant.

La procédure s'arrête lorsque plus aucun projet n'est  $\rho$ -abordable pour aucun  $\rho$  (qui ne peut excéder la fraction initiale du budget).

### 2.2 Préliminaires

Soit  $N = [n]$  l'ensemble des votants et  $C = (C_i)_{1 \leq i \leq n}$  l'ensemble de projets candidats.

La fonction  $\text{cout} : c \rightarrow \mathbb{Q}^+$  associe à tout projet  $c \in C$ , le coût lié à la réalisation de  $c$ .

Pour tout votant  $i \in N$ , la fonction  $u_i : C \rightarrow \{0, 1\}$  associe à tout projet  $c \in C$ , la valeur 1 si le votant  $i$  a voté pour  $c$  et 0 sinon.

### 2.3 Calcul du $\rho$ pour un projet

Au début de l'algorithme Rule X, l'ensemble des projets sélectionnées noté  $W$  est vide. A chaque itération, un projet va être sélectionné, diminuant le budget de chaque votant.

Soit  $c \in C$ , on note  $p_i(c)$ , le prix payé par le votant  $i$  pour financer le projet  $c$ . Par extension  $p_i(W) = \sum_{c \in W} p_i(c)$ . Ainsi

le budget restant du votant  $i$  après avoir financé les projets de  $W$  est  $\text{budget}_i(W) = 1/n - p_i(W)$ .

Soit  $c \notin W$ , calculer  $\rho$  de telle sorte que le projet  $c$  soit  $\rho$ -abordable revient à résoudre:

$$\sum_{i \in N} \min(\text{budget}_i(W), \rho * u_i(c)) = \text{cout}(c)$$

Il suffit donc de résoudre une équation sur une fonction affine par morceaux.

### 2.4 Algorithme

L'algorithme est décrit ci-dessous:

**Algorithm 1**  $RuleX(C, N)$ 


---

**input** :  $C$ , ensemble de projets candidats  
 $N$ , ensemble de votants  
**output**:  $W$ , ensemble des projets sélectionnés

```

1  $W = \emptyset$ 
2 while  $C \neq \emptyset$ 
3    $D = \emptyset$  // Ensemble de projets
   abordables
4   for  $c \in C$  do
5     Calculer  $\rho_c$  de telle sorte que  $c$  soit  $\rho_c$ -abordable
6     if  $\rho_c \leq 1/n$  then
7        $D \rightarrow D \cup c$  // On garde seulement
       les projets abordables
8    $c^* = \min_{c \in D} \{\rho_c\}$  // Sélection du meilleur
   projet parmi les projets abordables
9    $W = W \cup c^*$ 
10   $C = D$ 
11  Mettre à jour les budgets des votants de  $c^*$ 

```

---

## 2.5 Implémentation

L'algorithme est implémenté en Python en orienté objet.  
 Nous décrivons ci-dessous les attributs des classes utilisées.  
 Projet:

- nom
- cout

Votant:

- budget dépensé
- projets choisis

Budget Participatif:

- projets
- votants

Il serait également possible de représenter une matrice binaire des votes

$$A = (a_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$$

de telle sorte que  $a_{ij} = 1$  si et seulement si le projet  $i$  a été sélectionné par le votant  $j$ . Si  $a_{ij} \in [0, 1]$ , alors on peut voir cela comme la matrice des utilités.

## 3 Exemple fictif

### 3.1 Description des données

Afin de tester l'algorithme sur des données "réelles", nous avons organisé au sein de notre master un petit budget participatif fictif.

Disposant d'une enveloppe toute aussi fictive de 10.000 €, nous avons invités nos camarades à choisir au maximum trois projets parmi les projets candidats (Table 1). Nous avons récupéré les votes de 10 d'entre eux (Table 2) à l'aide d'un sondage sur Google Forms.

Projet	cout (€)
Végétalisation de la façade	3000
Achat Piano en libre service	3500
4 tables pique nique dans la cour	2000
Installation boîtes à livres	1000
Matériel loisir en locations	1200
Achat billard/baby foot	1000
Encarts affichage libre couloirs	800

Table 1: Projets et coûts associés

Projet	1	2	3	4	5	6	7	8	9	10
Végétalisation	x	x	x	x	x			x		
Piano	x	x	x							
tables pique nique							x	x	x	
boîtes à livres				x		x		x		
Matériel loisir	x				x		x		x	
billard/baby foot					x	x	x		x	x
Encarts affichage				x						x

Table 2: Matrice des votes

### 3.2 Résultats

La Table 3 donne les projets sélectionnés par Rule X et par l'algo. naïf. Les détails des itérations sont accessibles depuis le notebook. On peut y observer que le projet "Achat Piano en libre service" est écarté dès la première. En effet, seuls les votants 1, 2 et 3 ont choisis ce projet. Ensemble, ils détiennent au plus 30% du budget total tandis que le projet nécessite 35% du budget. Ce projet est pourtant sélectionné par l'algorithme naïf, donnant ainsi plus de "pouvoir d'achat" à ces votants que ce qu'ils ne méritent. On peut aussi remarquer que l'algorithme Rule X s'intéresse d'abord à des projets de moindre couts apportant un maximum d'utilité. De plus, la totalité du budget n'est pas utilisée, il restait assez d'argent pour financer "4 Tables". Cela montre que Rule X favorise les "bons" projets, quitte à ne pas utiliser toute l'enveloppe.

Projet	Algo. naïf	Rule X (€)
Végétalisation de la façade	x	x
Achat Piano en libre service	x	
4 tables pique nique dans la cour		
Installation boîtes à livres		x
Matériel loisir en locations	x	x
Achat billard/baby foot	x	x
Encarts affichage libre couloirs	x	x

Table 3: Projets et coûts associés

## Références

D. Peters, G.Pierczyński and P.Skowron. 2020. Proportionnal Participatory Budgeting with Cardinal Utilities.