Les enchères et les mécanismes véraces

March 27, 2006

Enchère de Vickrey

Données

- n joueurs veulent un objet $\mathcal{O}$
- Un objet a une valeur réelle (privée) t_i pour chaque joueur i
- Chaque joueur i donne son prix d_i dans une enveloppe scéllée.

Déroulement

1. Phase d'affectation: déterminer Gagnant celui qui a donné la meilleure offre
2. Phase de déterminer le prix :fixer Prix: la deuxième meilleure offre.

Illustration

	valeur qu'il est prêt à mettre	valeur qu'il déclare
joueur 1	10	5
joueur 2	8	8
joueur 3	4	6

Qui remporte? le joueur 2 au prix de 6.

Enchère de Vickrey

Données:

- n joueurs voulant acquerir un object $\mathcal{O}$
- vecteur de type $\vec{t}$ et de déclarations $\vec{d}$

Objectif global : déterminer un prix le plus haut possible.

Objectif local pour i: $u_i = \begin{cases} \text{maximiser } t_i - prix & \text{si } i \text{ achète } \mathcal{O} \\ 0 & \text{sinon} \end{cases}$

Valeur du *prix*: $\max\{d_j : j \neq \arg\max\{d_j\}\}$

Illustration

	valeur qu'il est prêt à mettre	valeur qu'il déclare	p_i	$u_i = t_i - p_i$	*prix*
joueur 1	10	5	0	0	6
joueur 2	8	8	6	2	6
joueur 3	4	6	0	0	6
Stratégie "*dire la vérité*"					
joueur 1	10	10	8	2	8
joueur 2	8	8	0	0	8
joueur 3	4	4	0	0	8

Illustration

	valeur qu'il est prêt à mettre	valeur qu'il déclare	p_i	$u_i = t_i - p_i$	*prix*
joueur 1	10	10	0	0	10
joueur 2	8	11	10	-2	10
joueur 3	4	6	0	0	10
Stratégie "*dire la vérité*"					
joueur 1	10	10	8	2	8
joueur 2	8	8	0	0	8
joueur 3	4	6	0	0	8

Le mécanisme

Données:

- Soit G un ensemble d'objects
- Soit P un ensemble de n acheteurs (vecteurs de type $\vec{t}$ et de déclarations $\vec{d}$)

$$\text{utilité du joueur } i\text{: } u_i = t(s) - paiement$$

Un mécanisme est caractérisé par

1. un algorithme d'allocation *alloc*: $alloc_i(\overrightarrow{d})$ le lot affecté au joueur i en fct. des déclarations $\overrightarrow{d} = \{d_1, \ldots, d_n\}$
2. une fonction de paiement *prix*: $prix_i(\overrightarrow{d})$ le prix que le joueur i doit honnorer en fct. des déclarations $\overrightarrow{d} = \{d_1, \ldots, d_n\}$

L'utilité pour i: $u_i = t(alloc_i(\overrightarrow{d}) - prix_i(\overrightarrow{d})$ si $alloc_i(\overrightarrow{d}) \neq \emptyset$
sinon $u_i = 0$

Mécanisme vérace

Un mécanisme $(alloc, prix)$ est vérace si $\forall$ *agent* i, “dire la vérité” est une stratégie dominante, i.e:

$$\forall \overrightarrow{d} \text{ vecteur de déclarations}, u_i(t_i, d_{-i}) \geq u_i(d_i, d_{-i})$$

En d’autre terme, $\forall \overrightarrow{d}$ le vecteur de déclarations,

$$t(alloc_i(t_i, d_{-i})) - prix_i(t_i, d_{-i}) \geq t(alloc_i(d)) - prix_i(d)$$

Théorème: L’enchères de Vickrey est un mécanisme vérace.

Enchères de Vickrey généralisées (GVA).

Soit $\vec{d}$ le vecteur des déclarations.

- La fonction d'allocation est de maximiser la somme des valeurs déclarées pour les acheteurs.

$$alloc(\vec{d}) = argmax_{a \in Allocation} \sum_{i=1}^{n} d_i(a_i(\vec{d}))$$

Rappel: $a_i(\vec{d})$ correspond à lot affecté à i par allocation a

- Chaque acheteur recoit la prime correspondant à la somme correspondant au gain si il participe.

$prix_j(\vec{d})$ = valeur déclarée -(Valeur de l'ensemble avec sa déclation - Valeur de l'ensemble SANS sa déclation)

Formellement,

$$prix_j(\vec{d}) = - \sum_{i=1, j \neq i}^{n} d_i(g_i(\vec{d})) + \sum_{i=1, j \neq i}^{n} d_i(g_i(\vec{z}))$$

où $z_i = d_i$ pour $j \neq i$ et $z_j(s) = 0 \ \forall s \in G$

1ière Illustration

- Un objet R en 4 copies.
- 3 joueurs `Alice`, `Bob`, `Sam`:
  1. `Alice` et `Bob` : veulent 2 copies aux prix de 8
  2. `Sam` veut 3 copies au prix de 15
- affectation optimale: 2 copies pour `Alice`, `Bob`
- Sans `Alice`, `ou Bob` : 3 copies pour `Sam` (15)
- payement pour `Alice`, `ou Bob` $= 7 = 8 + (16 - 15) =$

2ième Illustration: GVA non robuste aux joueurs fictifs

- 2 joueurs `Alice, Bob`:
  - `Alice` : 2 au prix de 20
  - `Bob` : 2 au prix de 30
- affectation : `Bob`
- prix$=30-(30-20)=20$

- 3 j. `Alice, Bob, Sam`:
  - `Alice` : 2 au prix de 20
  - `Bob` : 1 au prix de 15
  - `Sam` : 1 au prix de 15
- affectation : `Bob`, `Sam`
- prix$=15-(30-20)=5$

Enchères de Vickrey généralisées (GVA).

Théorème: GVA est un mécanisme vérace.

Problème du plus court chemin

Données: un graphe $G = (V, E)$

- Un sommet w est un agent w
- Chaque sommet w a un type t_w (un poids)
- deux sommets de x et y distincts

Objectif global:

Trouver un chemin C de $+$ court chemin entre x et y.

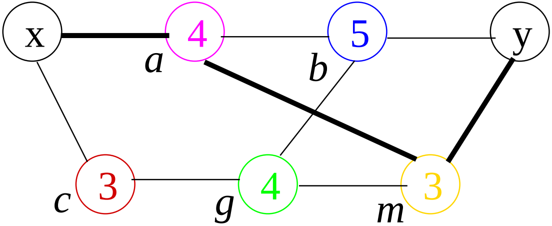
Objectif local de w:

minimiser le coût de w avec

$$cout(w) = \begin{cases} 0 & \text{si } C \text{ ne traverse pas } w \\ t_w & \text{si } C \text{ traverse } w \end{cases}$$

C'est-à-dire

Données:



Objectif global: le $\oplus$ court chemin $C = \{x, a, m, y\}$ de poids 7

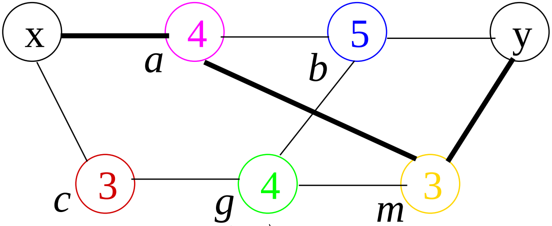
Objectif local:

- $cout(b) = cout(c) = cout(g) = 0$
- $cout(a) = 4$ et $cout(m) = 3$

Constatation:

Si les agents déclarent eux mêmes leur types, alors

- ils auront intérêt à tout faire pour ne pas être sur le plus court chemin
- ils vont déclarer des poids très élevés,



type	$\overrightarrow{t} = \{4, 5, 3, 4, 3\}$
déclaration	$\overrightarrow{d} = \{8, 5, 3, 4, 3\}$

Problème: l'interêt global diverge de l'interêt "local".

Comment imposer aux agents de dire la vérité ?

en donnant des primes!!!

Constatation:

Si les agents déclarent eux mêmes leur types, alors

- ils auront intérêt à tout faire pour ne pas être sur le plus court chemin
- ils vont déclarer des poids très élévés,

Problème: interêt global diverge de l'interêt "local".

Comment imposer aux agents de dire la vérité ?

en donnant des primes!!!

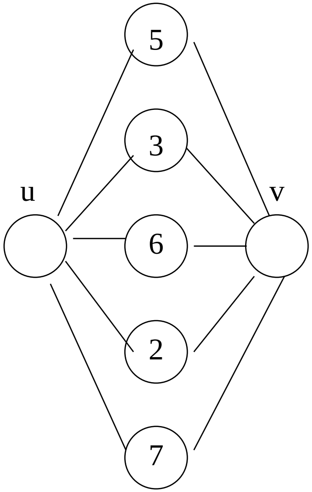
Enchère de Vickrey : Analogie

Données: vecteur de type $\vec{t}$ et de déclarations $\vec{d}$ voulant un objet $\mathcal{O}$
Objectif global : déterminer un prix. (trouver le plus court chemin)
Fct d'utilité pour i: maximiser

$$u_i = \begin{cases} t_i - p_i & \text{si } i \text{ achète } \mathcal{O} \\ 0 & \text{sinon} \end{cases}$$

Prime pour i:

$$p_i = \begin{cases} \max\{d_j : j \neq i\} & \text{si } i \text{ achète } \mathcal{O} \\ 0 & \text{sinon} \end{cases}$$



Formalisation:

Données: un graphe $G = (V, E)$ de n sommets

- Chaque sommet (agent) w a
$$\begin{cases} t_w & \text{un type privé} \\ d_w & \text{un type déclaré} \\ d_w(t_w) & \text{une fonction d'utilité déclaré} \end{cases}$$
- deux sommets de x et y distincts

Objectif global: Trouver un algorithme $A(\overrightarrow{d})$ qui retourne

1. un chemin C en fct des déclarations $\overrightarrow{d} = \{d_1, \dots, d_n\}$
2. et des primes p_w par agent w dépendant de $\overrightarrow{d}$

Objectif local de w: $v_w(t_w) = \begin{cases} 0 & \text{si } w \notin C \\ -t_w & \text{si } w \in C \end{cases}$

Définition: mécanisme

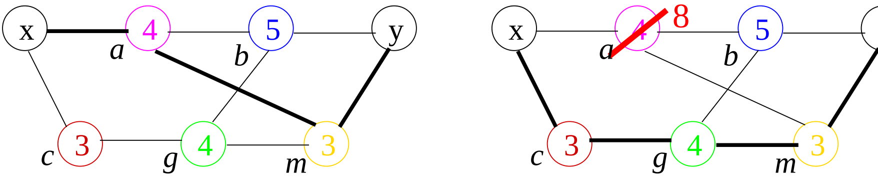
Données: Chaque agent w possède des donnés privées t_w et publique d_w, et une fonction d'utilité v_w.
Un mécanisme est caractérisé par

1. un algorithme $A(\overrightarrow{d})$ qui retourne un résultat à partir des déclarations $\overrightarrow{d} = \{d_1, \ldots, d_n\}$
2. des primes p_w par agent w dépendant de $\overrightarrow{d}$

Déroulement

1. n agents ayant un type t_i et déclaration d_i pour i
2. mécanisme renvoie $A(\overrightarrow{d})$
3. w reçoit la prime p_w dépendant de $\overrightarrow{d}$ et de $A(\overrightarrow{d})$ et w calcule $v_w(t_w, A(\overrightarrow{d}))$
4. Objectif de w: maximiser $u_w = p_w + v_w$

Illustration



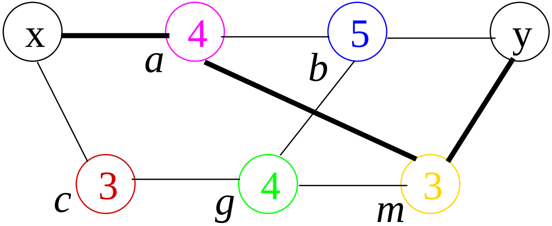
$\overrightarrow{t} = \{4, 5, 3, 4, 3\}$ $\overrightarrow{d} = \{8, 5, 3, 4, 3\}$ $A(\overrightarrow{d}) = \{x, c, g, m, y\}$

Le mécanisme sur le plus court chemin.

- $A(\overrightarrow{d})$: algo. du $\oplus$ court chemin avec $\overrightarrow{d}$ comme poids.
- Fixons i et d_{-i}. $L_{d_i=x}$ = la longueur du *pcc* si $d_i = x$.
- Posons $C_i = -L_{d_i=0} + L_{d_i=+\infty}$ et
- définission la prime
  $$p_i = \begin{cases} C_i & \text{si i est dans le plus court chemin} \\ 0 & \text{sinon} \end{cases}$$
- Chaque agent veut maximiser $u_i = p_i + v_i$.

Illustration: max $u_i = p_i + v_i$

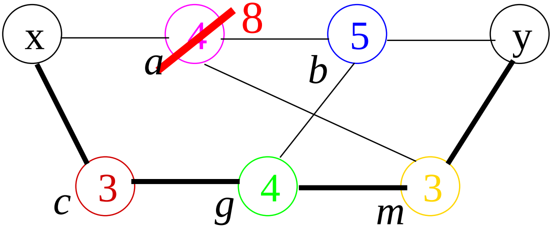
Si tout le monde choisit de dire la vérité,



- pour les agents c, g, m ($\notin pcc$): $p_i = 0$ et $u_i = 0$
- pour l'agent a ($\in pcc$): $p_a = 7$ et $u_a = 3$
  - $p_a = -L_{d_a=0} + L_{d_a=+\infty}$
  - $L_{d_a=0} = 3$ et $L_{d_a=+\infty} = d_{c-g-m} = 10$
  - $v_a = -t_a = -4$ et $u_a = 3 \geq 0$
- pour l'agent m ($\in pcc$): $p_m = 5$ et $u_m = 2$
  - $L_{d_m=0} = d_a = 4$ et $L_{d_m=+\infty} = d_{a-b} = 9$
  - $p_m = -4 + 9 = 5$ et $u_m = -t_m + 5 \geq -3 + 5 \geq 0$

Illustration: max $u_i = p_i + v_i$

Si l'agent a ne dit pas la vérité et déclare 8,

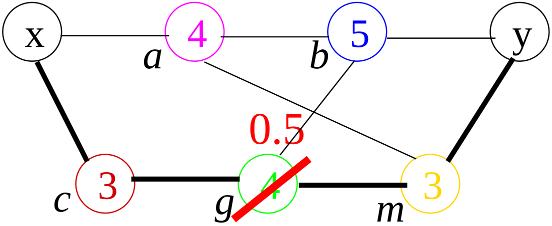


- pour l'agent a ($\notin pcc$): $p_a = 0$ et $u_a = 0$.

  L'agent a ne tire pas profit de son mensonge

Illustration: max $u_i = p_i + v_i$

Si l'agent 4 ne dit pas la vérité et déclare 0.5,



- pour l'agent 4 ($\in pcc$): $p_4 = 1$ et $u_4 = -3$.
  - $p_4 = -L_{d_4=0} + L_{d_4=+\infty} = -6 + 7 = 1$
  - $u_4 = -4 + 1 \leq 0$

L'agent 4 ne tire pas profit de son mensonge

Les mécanismes véraces

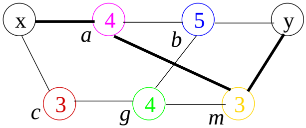
Un mécanisme est vérace si $\forall\ i$, "dire la vérité" est une stratégie dominante, i.e:

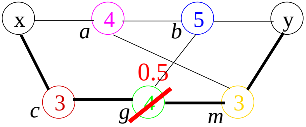
$$\forall d_{-i}\ , \forall d_i\ ,\quad u_i(t_i, d_{-i}) \geq u_i(d_i, d_{-i})$$

Un mécanisme est strictement vérace si il est vérace et s'il n'existe pas d'autre stratégie qui soit toujours aussi bonne que dire la vérité:

$$\forall d_i \neq t_i\ ,\ \exists d_{-i}\ ,\ u_i(t_i, d_{-i}) > u_i(d_i, d_{-i})$$

Signification de la stratégie dominante





Stratégie "dire la verité" "déclarer 0.5 au lieu de 2"

- stratégie dominante s_i: $\forall d_{-i}\ ,\forall d_i\ , u_i(s_i, d_{-i}) \geq u_i(d_i, d_{-i})$
- stratégie que 4 a d'annoncer 0.5 n'est pas dominante car

$$\underbrace{u_i(s_i, d_{-i})}_{-4+1} \quad \leq \quad \underbrace{u_i(t_i, d_{-i})}_{0}$$
$$-4+1 \quad \leq \quad 0$$

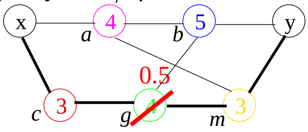
mécanisme vérace ?

Lemme 1: $C_i = -L_{d_i=0} + L_{d_i=+\infty}$ est une valeur seuil :

1. $d_i < C_i \Rightarrow$ l'agent i est dans le plus court chemin.
2. $d_i > C_i \Rightarrow$ l'agent i n'est pas dans le plus court chemin.
3. $t_i = C_i \Rightarrow \forall \vec{d}, u_i(\vec{d}) = 0$.

Preuve **du point 1 : par contradiction : supp. que** $i \notin pcc$**.**

- **longueur du** $pcc = L_{d_i=+\infty}$
- **Donc** $L_{d_i=+\infty} \leq L_{d_i=0} + d_i$
- **Donc** $C_i \leq d_i$

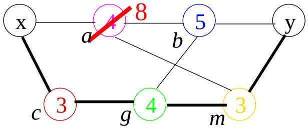


$d_i = 0.5, C_i = -6 + 7$

Preuve du Lemme.

du point 2 : par contradiction : supp. que $i \in pcc$.

- longueur du $pcc = d_i + L_{d_i=0}$
- Donc $d_i + L_{d_i=0} \leq L_{d_i=+\infty}$
- Donc $d_i \leq C_i$



$d_i = 8$, $C_i = -5 + 10$

du point 3 :

- Si $i \notin pcc$, alors $u_i = 0$
- Si $i \in pcc$, $u_i = -t_i + C_i = 0$.

mécanisme vérace ?

Proposition 1: le mécanisme est vérace.

Preuve :

- $u_i(x, d_{-i})$ = utilité de l'agent i quand i annonce x.
- Si $t_i \leq C_i$.
  - i déclare t_i : $i \in pcc$ et $u_i(t_i) = v_i + p_i = -t_i + C_i \geq 0$.
  - i déclare d_i :
    - $d_i \leq C_i$: $i \in pcc$ et $u_i(d_i) = -t_i + C_i = u_i(t_i)$

      $i \in pcc$ dans les deux cas (le gain = la prime)
    - $d_i > C_i$: $i \notin pcc$ et $u_i(d_i) = 0$

      $u_i(t_i) \geq u_i(d_i)$

Preuve de la Proposition

- Si $t_i > C_i$: $\forall d_i$, $u_i(t_i) \geq u_i(d_i)$
  - i déclare t_i : $u_i(t_i) = 0$ car $i \notin pcc$.
  - i déclare d_i : $u_i(t_i) \geq u_i(d_i)$
    - $d_i \leq C_i$; $i \in pcc$. Donc $u_i(d_i) = -t_i + C_i \leq 0 = u_i(t_i)$
    - $d_i > C_i$; $i \notin pcc$ et $0 = u_i(t_i) = u_i(d_i)$

Problème du `single-minded bidder`

Ennoncé

1. un ensemble d'objets G et un ensemble d'acheteurs P
2. un acheteur b est caratérisé par un couple $\langle s, a \rangle$ où s est un ensemble d'objects $\mathcal{O}$ et a le prix du lot.

   fonction $s : P \rightarrow G \;\; s(b) = s$
   fonction $a : P \rightarrow prix \;\; a(b) = a$
3. hypothèse du principe du tout ou rien: un acheteur veut acheter tout le lot ou rien.

Algorithme glouton sur l'affectation

1. $L \leftarrow$ liste triée des acheteurs suivant l'ordre $\frac{a}{|s|^{1/2}}$
2. Examiner dans l'ordre chaque acheteur b de L

   Si b peut encore acquerir tous ces objets qu'il demande alors, ces objects lui seront affectés.

   Sinon aucun object lui est affecté.

Rapport d'approximation

Théorème: L'algorithme glouton AG sur l'affectation approxime l'allocation optimale OPT avec un facteur de $|k|^{1/2}$ avec $k = |G|$.

$$OPT \leq AG|k|^{1/2}$$

Preuve:

- Supposons que les 2 solutions n'ont aucun d'acheteurs en commun. Notons $r_i = \frac{a_i}{|s_i|^{1/2}}$
- $AG = \sum_{i \in AG} a_i \geq (\sum_{i \in AG} a_i^2)^{1/2} \geq (\sum_{i \in AG} r_i^2 |s_i|)^{1/2}$
- $OPT = \sum_{i \in OPT} r_i |s_i|^{1/2} \leq (\sum_{i \in OPT} r_i^2)^{1/2} \times (\sum_{i \in OPT} |s_i|)^{1/2}$
- $OPT \leq (\sum_{i \in OPT} r_i^2)^{1/2} \times (k)^{1/2}$

suite de la preuve

- Prouvons que $(\sum_{i \in OPT} r_i^2) \leq (\sum_{i \in AG} r_i^2 |s_i|)$
- Considérons un acheteur i dans l'affectation de OPT et il n'appartient pas à AG.
- Ceci implique que $\ell \in s_i$ est déjà alloué à un acheteur $j \in AG$ avec $r_j \leq r_i$
- Comme par définition, deux acheteurs dans OPT n'ont pas d'object en commun,

  l'acheteur j a au plus empéché $|s_j|$ acheteurs d'acquerir

$$(\sum_{i \in OPT_j} r_i^2) \leq r_j^2 |s_j|)$$

1ère Illustration

- 2 objects a et b, 3 joueurs $Alice, Bob, Sam$
- $Alice = \langle\{a\}, 10\rangle$, $Bob = \langle\{a, b\}, 19\rangle$, $Sam = \langle\{b\}, 8\rangle$,
- $L = \{Alice, Sam, Bob\}$
- affectation non optimale par l'algo. glouton $= \{Alice, Sam\}$
- affectation optimale $= \{Bob\}$

Adaption de la fct des paiments VCG pour l'algo glouton?

- 2 objects a et b
- 3 joueurs `Alice, Bob, Sam` avec
  $$Alice = \langle \{a\}, 10\rangle,\ Bob = \langle \{a, b\}, 19\rangle,\ Sam = \langle \{b\}, 8\rangle,$$
- affectation par l'algo. glouton $= \{Alice, Sam\}$
  - prix pour `Alice` $:= 11$ (> 10!!!!!!!!!!)
    - Sans `Alice`: affectation par l'algo. glouton $= \{Bob\}$
    - "prix"$= 10 - (18 - 19)$
  - prix pour `Sam` $:= 0$
    - San `Sam`: affectation par l'algo. glouton $= \{Alice\}$
    - "prix"$= 8 - (18 - 10) = 0$

$\boxed{\text{inadaptation du paiement de VCG pour l'algo glouton pour l'affection.}}$

mécanisme vérace avec l'affectation gloutonne

1. Soit L la liste triée (par ordre décroissant).
2. $n(j)$: l'acheteur après j recalé à cause de j mais satisfait si j n'était pas là. Formellement
$$n(j) = \min\{i : j > i, s(j) \cap s(i) \neq \emptyset, \forall \ell, \ell < i, \ell \neq j,\ \ell \textit{ alloue} \Rightarrow s(\ell) \cup s(i) = \emptyset\}$$

L'algorithme de distribution des primes est le suivant:

- j recoit 0 si il est non satisfait ou si $n(j)$ n'existe pas.
- si il existe un $n(j)$ et si j est servi, alors il paie $|s| \times c(n(j))$.

Résultat principal

Theorem

Le mécanisme composé de l'algo affectation et du schéma de paiement est vérace pour le problème single-minded bidders*.*

Preuve:

1. Principe du tout ou rien : oui.
2. Propriété de monotonocité : $s \subseteq g_j$, $s' \subseteq s$, $v' \geq v$, $s' \subseteq g'_j$.
3. principe de la valeur critique:
4. principe de la participation : oui.

$\square$

1ère Illustration

- 2 objets a et b et 3 acheteurs.
- $R = \langle\{a\}, 10\rangle$, $G = \langle\{a, b\}, 19\rangle$, $B = \langle\{b\}, 8\rangle$,
- liste $L = \{R, G, B\}$: R et B acceptés et G refusé.
- Si R ne participait pas, alors G accepté:R paie 9.5.
- G et B payent 0.

2ième Illustration

- 2 objets a et b et 3 acheteurs.
- $R = \langle \{a\}, 20 \rangle$, $G = \langle \{b\}, 15 \rangle$, $B = \langle \{a, b\}, 20 \rangle$,
- liste $L = \{R, G, B\}$: R et G acceptés et B refusé.
- Si R ne participait pas, alors B toujours refuséR paie 0. Idem pour pour G.
- G paie 0!!!