Approches computationnelles de la théorie des jeux - Mémoire de DEA

Mikael Cozic

3 janvier 2004

Introduction : théorie des jeux et computation

Un secteur de plus en plus actif de recherches s'intéresse aux interactions entre informatique théorique et théorie des jeux (et, plus largement, microéconomie) ; on trouve les présentations les plus récentes dans [P 01], [N 99] et [F 02]. Il y a essentiellement deux composantes à ce genre de rapprochement :

1/ approche computationnelle (algorithmique) des notions-clés de la théorie des jeux (et de la microéconomie) :

La plupart des recherches se concentrent sur

- les équilibres dans les jeux noncoopératifs (Nash, parfait en sous jeux, etc... ; cf. [vS 02])

- les solutions des jeux coopératifs (coeur, valeur de Shapley, etc...)

- la théorie de l'implémentation (ou "mechanism design" ; cf. [NR 01])

- les équilibres compétitifs ([DPS 02])

Cette première approche a de nombreux antécédents et de nombreux résultats classiques ; son développement est stimulé par le fait que la plupart des avancées de la théorie des jeux ont lieu indépendamment du contenu calculatoire des concepts et des théorèmes ; il existe donc naturellement de nombreux espaces à combler.

Il s'agit dans cette approche de se demander, par exemple, quel genre d'algorithme peut fournir les solutions prescrites par un certain équilibre, analyser sa complexité, en envisager des approximations, etc...L'exemple de prédilection est le problème (UNEQn) où il s'agit de trouver un équilibre de Nash dans un jeu à n joueurs.

Cette approche trouve une motivation *structurelle* dans le programme de recherches impulsé par H.Simon sur la rationalité limitée ; elle trouve une motivation plus *conjoncturelle* dans le développement du commerce électronique où il est véritablement question d'implémenter (au sens informatique) des algorithmes résolvant des problèmes de décision.

L'approche computationnelle a pour problématique générale une analyse de la complexité dans la théorie des jeux ; mais derrière ceci se cachent des phénomènes et des enjeux très différents. Par souci de clarté, aussi bien pour les discussions que pour les développements techniques qui suivront, il est bon de distinguer différents *points de vue* et différents *niveaux* auxquels cette analyse peut être pertinente.

Les trois principaux points de vue sont

- le point de vue du modélisateur : le modélisateur est celui qui représente une situation donnée par un certain jeu et prédit ou recommande une issue selon une certaine notion d'équilibre

- le point de vue du joueur : le joueur est celui qui participe au jeu, avec ses désirs, ses croyances et les actions qu'il peut entreprendre (ses opportunités)

- le point de vue de l'organisateur : l'organisateur celui qui instaure les règles d'un jeu et établit une connexion entre les décision possibles des joueurs et leurs conséquences

On peut aussi s'intéresser à la complexité d'un jeu à différents niveaux, les différents niveaux présentant plus ou moins d'affinités avec certains points de vue ; il est particulièrement important de bien les distinguer. Les niveaux de complexité suivants sont ceux qui seront traités dans le mémoire :

- la complexité des stratégies : quelle est la complexité des règles de décision que peuvent adopter les joueurs ?

- la complexités des solutions : dans un jeu, quelle est la complexité du calcul des équilibres ?

- la complexité des conséquences : quand, dans un jeu, les issues sont associées aux actions des joueurs par une fonction de conséquence, quelle est la complexité du calcul de cette fonction ?

2/ approche décisionnelle (ou économique) des réseaux informatiques : application de la théorie des jeux à des situations qui émergent dans les réseaux informatiques.

Liée au développement d' Internet, c'est la ligne de recherches la plus récente ; elle est notamment investiguée par C.Papadimitriou et N.Nisan. L'idée de base est relativement simple : il s'agit de prendre en compte dans la construction des protocoles et dans leur gestion le fait que les différents systèmes qui y participent ont leur intérêt propre. Ce qui signifie que chacun a des motivations qui peuvent éventuellement diverger avec les objectifs du réseau, et que chacun attend une juste rétribution pour l'exécution des tâches qu'il est amené à faire dans le réseau.

Cete problématique va donc mobiliser essentiellement la théorie de l'implémentation qui s'intéresse à la manière dont on peut motiver les agents à agir pour un objectif général, et la théorie des jeux coopératifs qui s'intéresse aux répartitions équitables. L'exemple de prédilection (cf. [FPSS 02]) est le problème ROUTAGE où l'on se demande comment inciter les éléments d'un réseau comme Internet à reporter véracement leurs coûts de transmission des données de telle sorte que les données soient acheminées à moindre coût.

Il est évident que ces deux approches sont appelés à être étroitement liées : pour élaborer un protocole sur la base de la théorie de l'implémentation, il faut encore que ce soit algorithmiquement faisable.

De manière générale, un des intérêt essentiels de ce rapprochement tient

- d'une part, à la superposition de compétences, de contraintes et de difficultés issues de chacune des deux disciplines ; par exemple, le recours à la théorie de l'approximation (compétence de l'informatique) pour les enchères combinatoires n'est pas nécessairement compatible avec le maintien de la véracité du mécanisme (contrainte de la théorie des jeux), comme on le verra dans le Chapitre III

- d'autre part, à l'émergence de problèmes inédits ou de contraintes nouvelles ; un bon exemple de ce phénomène est le problème de l'EMPLOI DU TEMPS étudié par [NR 01] et qui pose des difficultés inédites à la théorie de l'implémentation dont les solutions classiques ne fonctionnent pas ici.

Le mémoire est basé sur une série d'exposés et de recherches que j'ai pu faire avec M. de Rougemont au cours de mon stage dans l'équipe Algorithmique et Complexité du LRI (Orsay, avril-juin 2002). Il se consacre avant tout à la première approche qui vient d'être évoquée, l'approche computationnelle de la théorie des jeux.

Il est organisé selon les trois grands types de complexités que j'ai distinguées précédemment : dans le Chapitre I, on s'intéresse à la complexité des solutions ; dans le Chapitre II, à la complexité des stratégies ; et dans le Chapitre III, à la complexité des conséquences.

Références

[DPS 02] Deng, Papadimitriou et Safra, "On the Complexity of Equilibria", STOC 02

[F 02] Feigenbaum, "Agent Privacy in Distributed Algorithmic Mechanisms", Position Paper, http ://www.cs.yale.edu/ jf

[FPSS 02] Feigenbaum, Papadimitriou, Sami et Schenker, "A BGP-based Mechanism for Lowest-Cost Routing", POC 2002

[N 99] Nisan, "Algorithms for Selfish Agents", STACS 99

[NR 01] Nisan et Rosen, "Algorithmic Mechanism Design", ACEM 2001

[P 01] Papadimitriou, "Algorithms, Games and the Internet", STOC 01

[vS 95] von Stengel, "Computing Equilibria for two-person games", Technical Report n°253, Département d'Informatique,ETH Zürich, 1995

[vS 02] von Stengel, "Computing Equilibria for two-person games" in Aumann and Hart (eds) *Handbook of Game Theory*, vol.III, North-Holland, Amsterdam, 2002

Table des matières

Chapitre 1

Le calcul des équilibres de Nash

L'article séminal [N 51] introduit la notion-clé de la théorie des jeux non-coopératifs, l'**équilibre de Nash** ; il montre aussi un résultat fondamental, à savoir que si l'on prend en compte les stratégies aléatoires (dites *stratégies mixtes*), dans tout jeu il existe un équilibre de Nash. La preuve fait appel au Théorème de Kakutani grâce à une formulation en termes de point fixe de l'équilibre de Nash ; de ce fait, elle n'est pas constructive et ne donne donc pas de méthode pour exhiber un équilibre dans un jeu quelconque. C'est un phénomène que l'on retrouve dans la théorie des équilibres généraux puisque le Théorème d'existence de Arrow-Debreu (1954) fait appel à un cas particulier du Théorème de Kakutani, le Théorème de Brouwer.

Une question qui est donc laissée ouverte par ces théorèmes d'existence est celle de savoir comment trouver de tels équilibres ; dans ce chapitre, nous allons exposer l'algorithme classique qui permet de calculer un équilibre de Nash dans un jeu à deux joueurs, l'algorithme de Lemke-Howson.

1.1 Notions élémentaires

1.1.1 Les jeux finis à N joueurs

Dans un jeu non-coopératif, plusieurs joueurs entrent en interaction ; chaque joueur dispose d'un certain nombre d'actions et peut tirer différents bénéfices du jeu selon les actions que lui et les autres entreprennent (selon le *profil d'actions* choisi, pour utiliser la terminologie standard).

Par exemple, dans la Bataille des Sexes, deux joueurs ont chacun deux actions possibles à disposition pour la soirée à venir : aller au concert ou aller au match de football. Les deux joueurs ont avant tout envie de passer la soirée ensemble et donc l'option la pire pour chacun est de se retrouver seul ; par contre, l'un des joueurs préfère aller au concert, l'autre aller au football. On peut représenter une telle situation par la matrice suivante :

	concert	football
concert	(2,1)	(0, 0)
football	(0, 0)	(1, 2)

La prédiction canonique de la théorie des jeux, appelée *équilibre de Nash*, se base sur l'idée qu'un profil d'actions est stable si aucun des joueur n'a intérêt à dévier unilatéralement, c'est-à-dire si aucun des joueurs n'a intérêt à entreprendre une autre action si celles des autres restent les mêmes.

Par exemple, dans la Bataille des Sexes, les profils d'actions (concert, concert) et (football, football) sont des équilibres de Nash.

Définitions

- un **jeu** fini à N joueurs sous forme normale est un n-uplet $< N, S = (S_i)_{i\in N}, u = (u_i)_{i\in N} >$ où
  - N est l'ensemble des joueurs
  - S_i est l'ensemble des stratégies du joueur i
  - $u_i : S \longrightarrow \mathcal{R}$ est la fonction de paiement du joueur i
- une **stratégie mixte** p_i pour le joueur i est une distribution de probabilité sur S_i ; soit $\Delta(S_i)$ l'ensemble des distributions de probabilité sur S_i
- un n-uplet $p^* \in \Delta(S) = \prod_{i\in N}(\Delta(S_i))_{i\in N}$ de stratégies mixtes est un **équilibre de Nash** si pour tout $i \in N$ et tout $p_i \in \Delta(S_i)$, $u_i(p_i, p^*_{-i}) \leq u_i(p^*)$

1.1.2 La programmation linéaire

Définition (Le problème LP Primal)

Soit une matrice $A \in \mathbf{R}^{M\times N}, I \subseteq M, J \subseteq N, b \in \mathbf{R}^M, c \in \mathbf{R}^N$ (c est appelé la **fonction objectif**) ;

l'ensemble $P = \{x \in \mathbf{R}^N /$

- $\sum_{j\in N} a_{ij}x_j = b_i, i \in M - I$
- $\sum_{j\in N} a_{ij}x_j \leq b_i, i \in I$
- $x_j \geq 0, j \in J\}$

est appelé l'**ensemble faisable**.

Le problème LP primal est le problème :

$$\boxed{\text{maximiser } c^\top x \text{ sujet à } x \in P}$$

Exemple :

maximiser $4x_1 + x_2 + 5x_3 + 3x_4$

sujet à

$$
\begin{aligned}
x_1 - x_2 - x_3 + 3x_4 &\leq 1 \\
5x_1 + x_2 + 3x_3 + 8x_4 &\leq 55 \\
-x_1 + 2x_2 + 3x_3 - 5x_4 &\leq 3 \\
x_1, x_2, x_3, x_4 &\geq 0
\end{aligned}
$$

Ici, nous avons : $n = 4$, $I = M = 3$ et, par exemple, $b_1 = 1, a_{11} = 1, a_{12} = -1$, etc... $\diamondsuit$

Etant donné un problème LP Primal, on peut chercher une stratégie générale pour trouver des bornes supérieures à la valeur optimale de la fonction objectif ; la fonction objectif du problème LP Dual fournit de telles bornes.

Définition (Le problème LP Dual)

Soit un problème LP Primal défini comme ci-dessus ; le problème LP Dual correspondant a pour ensemble faisable $D = \{y \in \mathbf{R}^M /$

- $\sum_{i \in M} y_i a_{ij} = c_j, j \in N - J$
- $\sum_{i \in M} y_i a_{ij} \geq c_j, j \in J$
- $y_i \geq 0, i \in I\}$

et est défini comme le problème

$\boxed{\text{minimiser } y^\top b \text{ sujet à } y \in D}$

Exemple :

Reprenons l'exemple précédent et regardons comme l'introduction de variables auxiliaire nous conduit naturellement au problème dual. Si nous multiplions chaque contrainte b_i par une nouvelle variable y_i et les additionnons, nous obtenons :

$(y_1 + 5y_2 - y_3)x_1 + (-y_1 + y_2 + 2y_3)x_2 + (-y_1 + 3y_2 + 3y_3)x_3 + (3y_1 + 8y_2 - 5y_3)x_4 \leq y_1 + 55y_2 + 3y_3$

Soit

$$
\begin{aligned}
\alpha^1 &= y_1 + 5y_2 - y_3 \\
\alpha^2 &= -y_1 + y_2 + 2y_3 \\
\alpha^3 &= -y_1 + 3y_2 + 3y_3 \\
\alpha^4 &= 3y_1 + 8y_2 - 5y_3
\end{aligned}
$$

S'il est vrai que

$$
\begin{aligned}
\alpha^1 &\geq 4 \\
\alpha^2 &\geq 1 \\
\alpha^3 &\geq 5 \\
\alpha^4 &\geq 3
\end{aligned}
$$

alors $4x_1 + x_2 + 5x_3 + 3x_4 \leq y_1 + 55y_2 + 3y_3$
Nous sommes alors conduits au problème LP suivant :
minimiser $y_1 + 55y_2 + 3y_3$
sujet à

$$\begin{aligned} y_1 + 5y_2 - y_3 &\geq 4 \\ -y_1 + y_2 + 2y_3 &\geq 1 \\ -y_1 + 3y_2 + 3y_3 &\geq 5 \\ 3y_1 + 8y_2 - 5y_3 &\geq 3 \end{aligned}$$

Cela correspond exactement à la définition donnée précédemment.

Le résultat élémentaire dont on a besoin pour les applications à la théorie des jeux, et que nous ne démontrerons pas, est le théorème de Dualité :

Théorème (Théorème de Dualité)
Soit une paire Primal-Dual définie comme ci-dessus ;
– (a) Dualité faible : pour tout $x \in P$ et $y \in D$, $c^\top x \leq y^\top b$
– (b) Dualité forte : si $P, D \neq \emptyset$, alors il existe $x \in P, y \in D$ tels que $c^\top x = y^\top b$

1.2 Les équilibres de Nash dans les jeux à 2 joueurs

Soit un jeu à deux joueurs ; nous avons alors les données suivantes :
• deux ensembles de stratégies pures, S_1 et S_2 dont on note les cardinalités comme suit : $|S_1| = m_1$ et $|S_2| = m_2$
• deux fonctions de paiement u_1 et u_2

Le problème que l'on va traiter, UNEQ2, est celui de savoir comment, à partir de ces données, trouver *un* équilibre de Nash dans le jeu - c'est donc un problème différent de celui de trouver *tous* les équilbres de Nash, que l'on peut appeler TOUSEQ2.

1.2.1 L'équilibre de Nash comme problème de programmation

Pour utiliser la programmation dans la résolution de UNEQ2, nous allons d'abord commencer par donner une traduction des concepts de la théorie des jeux dans langage matriciel qui est celui de la programmation.

Définition

U_i est la matrice des paiements pour le joueur i c'est-à-dire la matrice

– dont les lignes sont indexées par les m_i stratégies de i

– dont les colonnes sont indexées par les m_j stratégies de $i \neq j$

Notation

On dénote par

– u_{1i} les lignes de la matrice U_1 pour $i \in S_1$
– u_{2j} les lignes de la matrice U_2 pour $j \in S_2$

Exemple :

Voici un jeu où le premier joueur a trois actions possibles, le second deux ; dans la forme quasi-matricielle ordinaire, on aurait présenté les paiements de la manière suivante :

(0,1)	(6,0)
(2,0)	(5,2)
(3,4)	(3,3)

Les matrices induites U_1 et U_2 sont donc les suivantes :

0	6
2	5
3	3

U_1

1	0	4
0	2	3

U_2

$\diamondsuit$

Définition
– l'ensemble des stratégies mixtes pour le joueur 1 est $X = \{x \in \mathbf{R}^{m^1 \times 1} / Ex = e, x \geq \mathbf{0}\}$ où $E = [1, ..., 1] \in \mathbf{R}^{1 \times m^1}$ et $e = 1$
– l'ensemble des stratégies mixtes pour le joueur 2 est $X = \{x \in \mathbf{R}^{m^2 \times 2} / Fx = f, x \geq \mathbf{0}\}$ où $F = [1, ..., 1] \in \mathbf{R}^{1 \times m^2}$ et $f = 1$

Definition
– le paiement espéré du joueur 1 pour la paire de stratégies mixtes $(x, y), x \in X, y \in Y$ est $x^\top(U_1 y)$
– le paiement espéré du joueur 2 pour la paire de stratégies mixtes $(x, y), x \in X, y \in Y$ est $(x^\top {U_2}^\top)y$

Exemple :
$x = (0, 1/3, 2/3), y = (2/3, 1/3)$

Le paiement espéré du joueur 1 est $x^\top U_1 y$ ie
$(0.2) + (1/3.(2/3.2 + 1/3.5)) + (2/3.(2/3.3 + 1/3.3)) = 3$

Définition
une meilleure réponse du joueur 1 à un $y \in Y$ est un vecteur x tel que x est une solution au problème LP suivant :

$$\boxed{\text{maximiser } x^\top (U_1 y) \text{ sujet à } Ex = e, x \geq \mathbf{0}}$$

Nous pouvons en effet reconnaître un problème de la forme

$$\boxed{\text{maximiser } c^\top x \text{ sujet à } x \in P}$$

avec
$P = \{x \in \mathbf{R}^N /$
– $\sum_{j \in N} a_{ij} x_j = b_i, i \in M - I$
– $\sum_{j \in N} a_{ij} x_j \leq b_i, i \in I$
– $x_j \geq 0, j \in J\}$
où $A \in \mathbf{R}^{\mathbf{m} \times \mathbf{n}}, I \subseteq M, J \subseteq N, b \in \mathbf{R}^M, c \in \mathbf{R}^N$
Ici :
$A = E$ est une $1 \times m_1$ matrice
x est un m_1-vecteur colonne
$J = M_1$
$I = \emptyset$
$c = U_1 y$ est un m_1-vecteur
$b = e$ est un 1-vecteur

Aussi, le problème Dual est :

$$\boxed{\text{minimiser } b^\top u \text{ sujet à } u \in DX}$$

où DX est l'ensemble faisable dual de X, c'est-à-dire, *mutatis mutandis*,

$$\boxed{\text{minimiser } e^\top u \text{ sujet to } E^\top u \geq U_1 y}$$

Par la Dualité forte, $x \in X$ est une solution optimale ssi il existe $u \in DX$ tel que

$$x^\top (U_1 y) = e^\top u$$

ce qui équivaut à

$$x^\top (U_1 y) = (Ex)^\top u$$
$$x^\top (U_1 y) = (x^\top E^\top) u$$
$$\underline{x^\top (E^\top u - U_1 y) = 0}$$

Les vecteurs x et $E^\top u - U_1 y$ sont, en raison des contraintes, nonnegatifs ; ils ne peuvent donc avoir de composantes positives en même position, c'est pourquoi on dit qu'ils sont **complémentaires**.

De la même façon,

une meilleure réponse du joueur 2 à un vecteur $x \in X$ est un vecteur y tel que y est une solution du problème LP suivant :

$\boxed{\text{maximiser } (x^\top U_2^\top)y \text{ sujet à } Fy = f, y \geq \mathbf{0}}$

Aussi, le problème Dual est :

$\boxed{\text{minimiser } v^\top f \text{ sujet à } F^\top v \geq U_2 x}$

Par la Dualité forte, $y \in Y$ est une solution optimale ssi il existe $v \in DY$ tel que

$$\begin{array}{c} (x^\top U_2^\top)y = v^\top f \\ y^\top (x^\top U_2{}^\top)^\top = (Fy)^\top v \\ y^\top (U_2 x^\top) = (y^\top F^\top)v \\ \underline{y^\top (F^\top v - U_2 x) = 0} \end{array}$$

Proposition
Soit $< U_1, U_2 >$ un jeu à 2 joueurs ; la paire (x^*, y^*) est une équilibre de Nash ssi pour u, v adéquats

$$\boxed{\begin{array}{l} (1)\ Ex^* = e \\ (2)\ Fy^* = f \\ (3)\ E^\top u \geq U_1 y^* \\ (4)\ F^\top v \geq U_2 x^* \\ (5)\ {x^*}^\top (E^\top u - U_1 y^*) = 0 \\ (6)\ {y^*}^\top (F^\top v - U_2 x^*) = 0 \\ (7)\ x^*, y^* \geq \mathbf{0} \end{array}}$$

(CLP1)

Remarque : en fait, si l'on se reporte à [CPS 92], la forme générale d'un problème linéaire complémentaire est : étant donné un vecteur $q \in \mathbf{R}^n$ et une $n \times n$-matrice M, trouver un vecteur $s \in \mathbf{R}^n$ t.q.

$$\boxed{\begin{array}{l} z \geq \mathbf{0} \\ b + Mz \geq \mathbf{0} \\ z^\top (b + Mz) = 0 \end{array}}$$

(CLP)

[KMS 96] montrent précisément comment **(CLP1)** est un cas particulier de **(CLP)**.

1.2.2 Approche de l'algorithme de Lemke-Howson en termes de graphes (Shapley 1974)

L'algorithme présenté dans [LH 64] est exposé d'une manière particulièrement abstraite ; comme [vS 02], nous allons donc commencer par détailler une approche très intuitive quand la cardinalité des ensembles de stratégies ne dépasse pas 3, l'approche en termes de graphes de [S 74]. Cette approche exhibe un équilibre de Nash en construisant un chemin sur le simplexe de chacun des ensembles de stratégies

L'étape cruciale dans la construction de tels chemins est l'étiquetage des stratégies ; chaque stratégie mixte s_i du joueur i est en effet étiquetée par
- les stratégies pures de i qui ne sont pas dans le support de s_i
- les stratégies pures de j qui sont des meilleures réponses à s_j

Grâce à cet étiquetage, nous allons pouvoir donner une définition équivalente de l'équilibre de Nash en termes d'étiquettes.

Formellement, on va construire l'étiquetage en définissant pour toute stratégie pure i du joueur 1 deux ensembles dont les éléments sont étiquetés par i : d'une part l'ensemble $X(i)$ de stratégies mixtes de 1 et d'autre part l'ensemble $Y(i)$ de stratégies mixtes de 2. On procède de la même façon pour les stratégies pures j du joueur 2.

Définition

Pour les stratégies $i \in S_1, j \in S_2$, soient
- $X(i) = \{x \in X \quad / \quad x_i = 0\}$
- $Y(j) = \{y \in Y \quad / \quad y_j = 0\}$
- $X(j) = \{x \in X \quad / \quad u_{2j}x \geq u_{2k}x, \forall k \in S_2\}$
- $Y(i) = \{y \in Y \quad / \quad u_{1i}y \geq u_{1k}y, \forall k \in S_1\}$

L'ensemble des étiquettes de la stratégie mixte x (resp. y) est alors $L(x) = \{k \in S_1 \bigcup S_2 \quad / \quad x \in X(k)\}$ (resp. $L(y) = \{k \in S_1 \bigcup S_2 \quad / \quad x \in Y(k)\}$).

Théorème

Une paire (x, y) de stratégies mixtes dans le jeu $< U_1, U_2 >$ est un équilibre de Nash ssi $k \in S_1 \bigcup S_2$, $x \in X(k)$ ou $y \in Y(k)$ (ou les deux).

Preuve

C'est une conséquence immédiate d'un résultat classique : une paire (x, y) de stratégies mixtes dans le jeu $< U_1, U_2 >$ est un équilibre de Nash ssi pour tout $i \in S_1, j \in S_2$
- si $x_i > 0$ alors $u_{1i}y \geq u_{1k}y, \forall k \in S_1$
- si $y_j > 0$ alors $u_{2j}x \geq u_{2k}x, \forall k \in S_2$

($\Rightarrow$) soit $k \in S_1 \bigcup S_2$; si $k \in S_1$, ou $x_k = 0$ et $x \in X(k)$; ou $x_k > 0$ et par hypothèse $y \in Y(k)$. Par le même argument, si $k \in S_2$, alors ou $y \in Y(k)$ ou $x \in X(k)$.

($\Leftarrow$) soit (x, y) une paire de stratégies et $i \in S_1$ = une stratégie pure du joueur 1 t.q. $x_i > 0$. IL faut montrer que $u_i y \geq u_k y, \forall k \in S_1$. Supposons que ce n'est pas le cas : $x \notin X(i)$ parce que $x_i > 0$ et $y \notin Y(i)$ par hypothèse donc il existe un $k \in S_1 \bigcup S_2$ t.q. $x \notin X(k)$ et $y \notin Y(k)$. Contradiction. ♠

A partir de cet étiquetage, on va définir un graphe pour chaque ensemble de stratégies.

Définition

soit G_1 le graphe
- noeuds = $\{x \in X / |\{k / x \in X(k)\}| = m_1\} \bigcup \{\mathbf{0}\}$
- arêtes = $\{(x, x')/$ x et x' diffèrent par une étiquette $\}$

De la même façon,
G_2 est le graphe
- noeuds = $\{y \in Y / |\{k / y \in Y(k)\}| = m_2\} \bigcup \{\mathbf{0}\}$
- arêtes = $\{(y, y')/$ y et y' diffèrent par une étiquette $\}$

Enfin,
soit $G_1 \times G_2$ le graphe
- noeuds = $\{(x, y)/x \in G_1, y \in G2\}$
- arêtes = $\{((x, y), (x', y'))/(x, x')$ est une arête de G_1 et $y = y'$ ou (y, y') est une arête de G_2 et $x = x'\}$

Définition

soit $k \in S_1 \bigcup S_2$ et $(x, y) \in G_1 \times G_2$; (x, y) est une paire **presque-k complètement étiquetée** si $\forall l \in S_1 \bigcup S_2 - \{k\}$, l étiquette x ou y.

Lemme
- tout équilibre de Nash (x, y) (et le pseudo-équilibre $\mathbf{0}, \mathbf{0}$) est, dans $G_1 \times G_2$, adjacent à exactement un noeud (x', y') qui est presque-k complètement étiqueté.
- soit (x, y) une paire presque-k complètement étiqueté ; il y a exactement deux noeuds adjacents qui sont presque-k complètement étiquetés dans $G_1 \times G_2$

Preuve (sketch)

Considérons le graphe G_1 (le cas de G_2 est symétrique). Soit $a = (a_1, ..., a_{m_1})$ un noeud de G_1, a_i étant une étiquette de a ; nous avons à montrer que pour chaque k, $1 \leq k \leq m_1$, il existe exactement un noeud a' t.q. $a_i = a'_i$ pour tout $1 \leq i \leq m_1$ sauf pour k où $a_k \neq a'_k$.

Ceci revient à dire que dans le graphe G_1 chaque noeud touche exactement m_1 autres noeuds, chacun différant du noeud initial par une étiquette différente.

Ce n'est pas trivial, car pour une étiquette donnée a_k, il existe *a priori* plusieurs étiquettes possibles qui pourraient la remplacer. On peut cependant démontrer cela en admettant l'hypothèse (HND) qui stipule que la dimension des stratégies mixtes étiquetées par un ensemble d'étiquettes décroît avec la taille de cet ensemble.

Précisons avant de l'énoncer en détails que l'on se situe dans l'espace euclidien, et que l'on considère l'enveloppe convexe des stratégies pures ; la *dimension* d'un ensemble convexe dans ce contexte est d ssi si l'ensemble a $d+1$ (et pas plus) points affinement indépendants.

Hypothèse de Non Dégénérescence (HND)

pour toute stratégie mixte $x \in X$ étiquetée par K (resp. $y \in Y$ étiquetée l'ensemble L), $\bigcap_{k \in K} X(k)$ est de dimension $m_1 - |K|$ (et $\bigcap_{k \in L} Y(k)$ est de dimension $m_2 - |L|$)

Corollaire : tout point de G_1 (resp. G_2) est étiqueté différemment.

soit K^* un ensemble d'étiquettes de cardinal $m_1 - 1$ et supposons qu'il existe une stratégie mixte du joueur 1 étiqueté par K^* ; soit s l'ensemble des stratégies qui sont dans $X(K^*)$. D'abord, par (HND), s est de dimension 1 ; ensuite, s est dans le simplexe (dimension $m_1 - 1$) donc s est un segment ; enfin, à nouveau par (NHD), aucun point du simplexe n'a plus de m_1 étiquettes.

Nous allons montrer que les seuls points de s avec m_1 étiquettes sont les points extrêmes - si C est un ensemble conxexe, alors $x \in C$ est un point extrême ssi pour tout $x_i, x_j \in C$ et pour tout $\lambda \in [0,1]$, si $x = \lambda x_i + (1-\lambda)x_j, x_1 \neq x_2$, alors $\lambda \in \{0,1\}$. Soit x_1 et x_2 ces points extrêmes.

D'abord il est clair que, puisque nous sommes dans le simplexe des stratégies pures, les deux points extrêmes de s ont exactement m_1 étiquettes.

Il faut montrer qu'il n'existe pas de point de s, distincts des points extrêmes, porteur de m_1 étiquettes. Soit x un tel point, k son étiquette additionnelle et $[k]$ l'ensemble des stratégies mixtes étiquetées par k.

Ou bien $k \in S_1$: cela signifie que k n'est pas dans le support de x mais est dans le support de x_1 comme de x_2 ; c'est clairement impossible puisque xx est une combinason convexe de ces points.

Ou bien $k \in S_2$: remarquons d'abord que $[k]$ peut seulement toucher s en un point, autrement $L(x)$ aurait une dimension supérieure à 0, ce qui contredit (HND). Intuitivement, x ne peut être qu'un point "au milieu" du segment s. Supposons que ce soit le cas ; alors $[k]$ doit traverser l'un des $[l]$ pour $l \in K^*$ et $[k] \bigcap [l]$ sera de dimension $m_1 - 1$ au lieu d'être de dimension $m_1 - 2$. Par (NHD), c'est impossible. ♠

Théorème

Soit un jeu non dégénéré à deux joueurs $< U_1, U_2 >$ and $k \in U_1 \bigcup U_2$; alors l'ensemble des noeuds (et arêtes correspondantes) presque-k complètement étiquettés dans $G_1 \times G_2$ consiste en des chemins et des cycles disjoints. Les points terminaux des chemins sont les équilibres de Nash et le pseudo-équilibre $(\mathbf{0}, \mathbf{0})$. Le nombre des équilibres de Nash est impair.

A partir du Lemme précédent, on peut proposer un algorithme qui trace un chemin sur le graphe $G_1 \times G_2$ partant du pseudo-équilibre $(\mathbf{0}, \mathbf{0})$ et arrivant à un (véritable) équilibre de Nash.

algorithme de Shapley
(a) se placer au point $(\mathbf{0}, \mathbf{0})$
(b) choisir une étiquette $k \in S_1 \bigcup S_2$
(c) aller au (seul) point adjacent à $(\mathbf{0}, \mathbf{0})$ presque-k complètement étiqueté, noté (x_1, y_1). Il existe alors une étiquette l_1 dupliquée
(d) aller au (seul) point adjacent à (x_1, y_1) presque-l_1 complètement étiqueté et différent du point de provenance (ici $(\mathbf{0}, \mathbf{0})$)
(e) itérer l'étape (d) jusqu'à trouver un point (x_n, y_n) sans étiquette dupliquée ; ce point est un équilibre de Nash

Question : l'algorithme de Shapley résoud UNEQ2 ; mais est-ce que l'itération de l'algorithme permet de résoudre également TOUSEQ2 (c'est-à-dire : trouver tous les équilibres de Nash du jeu) ?

Réponse : non ; [S 74] donne un contre-exemple.

1.2.3 Approche algébrique de l'agorithme de Lemke-Howson

Dans cette section, on va voir la contrepartie de l'algorithme de Shapley en termes de programmation, en quoi consiste à proprement parler l'algorithme de Lemke-Howson. A chaque point du graphe produit $G_1 \times G_2$ correspondra la solution basique d'un système d'équations linéaires ; au parcours le long des arêtes du graphe, un changement de solution (appelé "pivot") ; et à l'atteinte d'un point complètement étiqueté, l'atteinte d'une solution qui satisfait la fonction objectif.

Notation

Nous dénotons par I_i la $i \times i$-matrice identité, ie la matrice t.q. si A est une $i \times k$-matrice, $I_i\times$ A = A

Le pivot complémentaire

Rappelons que l'ensemble des contraintes matricielles suivantes définissent un équilibre de Nash :

$$\boxed{\begin{array}{l}(1)\ Ex^* = e\\(2)\ Fy^* = f\\(3)\ E^\top u \geq U_1 y^*\\(4)\ F^\top v \geq U_2 x^*\\(5)\ {x^*}^\top (E^\top u - U_1 y^*) = 0\\(6)\ {y^*}^\top (F^\top v - U_2 x^*) = 0\\(7)\ x^*, y^* \geq \mathbf{0}\end{array}}$$

(CLP1)

Pour présenter l'algorithme, nous allons procéder à un certain nombre de manipulations sur cette formulation initiale. Nous allons d'abord, à la suite de [KL 96] et [vS 02], procéder à un changement de variables qui simplifiera la description exhaustive de l'algorithme :

- x' est associée à x, v : $x' = x/v$
- y' est associée à y, u : $y' = y/u$

La condition (4) de **(CLP1)** est

$$F^\top v \geq U_2 x$$

donc quand on change les variables

$$F^\top \geq U_2 x'$$

De la même façon, pour la condition (3),

$$E^\top \geq U_1 y'$$

Maintenant, comme dans la méthode du simplexe, on introduit des *slack* vecteurs z' et w', de manière à transformer ces inégalités en équations : alors on peut reformuler **(CLP1)** comme le problème de trouver $x', y', z', w' \geq \mathbf{0}$ avec $z' \in \mathbf{R}^{m_2}$ et $w' \in \mathbf{R}^{m_1}$ t.q.

(3') $U_1 y' + w' = E^\top$
(4') $U_2 x' + z' = F^\top$

Ce sont les contraintes élémentaires ; nous avons encore à rendre compte des contraintes de complémentarité (5) and (6) :

$$\begin{gathered}(5)\ x^\top (E^\top u - U_1 y) = 0\\ x'.v^\top (E^\top u - U_1 (y'.u)) = 0\\ x'^\top (u.(E^\top - U_1 y') = 0\\ x'^\top (E^\top - U_1 y') = 0\end{gathered}$$

Comme

$$\begin{gathered}w' = E^\top - U_1 y'\\ (5')\ x'^\top w' = 0\end{gathered}$$

De la même façon, on trouve

(6') $y'^{\top}z' = 0$

(3') $U_1y' + w' = E^{\top}$
(4') $U_2x' + z' = F^{\top}$
(5') $x'^{\top}w' = 0$
(6') $y'^{\top}z' = 0$
(7') $x', y', z', w' \geq \mathbf{0}$

(CLP3)

Posons $m = m_1 + m_2$; soient les matrices

$U = \begin{bmatrix} 0 & U_1 \\ U_2 & 0 \end{bmatrix}$, $q = \begin{bmatrix} E^{\top} \\ F^{\top} \end{bmatrix}$, $s = \begin{bmatrix} x' \\ y' \end{bmatrix}$, $r = \begin{bmatrix} w' \\ z' \end{bmatrix}$

Exemple

$$U = \begin{bmatrix} 0 & 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 2 & 5 \\ 0 & 0 & 0 & 3 & 3 \\ 1 & 0 & 4 & 0 & 0 \\ 0 & 2 & 3 & 0 & 0 \end{bmatrix}$$

$\diamondsuit$

Ce qui permet de rendre la forme du problème plus compact encore : $Us+r = q$ est équivalent à (5')+ (6') si $s^{\top}r = \mathbf{0}$.

Soit les matrices $C = \begin{bmatrix} U & I_{m_1+m_2} \end{bmatrix}$ et $t = \begin{bmatrix} s \\ r \end{bmatrix}$

Alors $Us + r = q$ est à son tour équivalent à $Ct = q$ (EQ).

Exemple

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 6 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 5 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 3 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Définition

soit U la matrice carrée de taille m définie comme précédemment ; et $\beta = \{b_1, ..., b_m\}$ une collection de m indices et C_β la matrice carrée dont la i-ème colonne est la colonne b_i de C. Si C_β est invertible, β est appelée une **base**.

Exemple

$$\beta = \{w'_1, w'_2, w'_3, z'_4, z'_5\}; C_\beta = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Définition

étant donné une base β, le vecteur t est appelé une **solution basique** si
– pour $t_\beta = (t_i)_{i\in\beta}$, $C_\beta t_\beta = q$
– pour $j \notin \beta, t_j = 0$
Si $t \geq \mathbf{0}$, cette solution basique est appelée **faisable**.

Notation

si $i \in \beta$, t_i est appelée une variable basique ; si $j \notin \beta$, t_j est appelée une variable non basique.

Etant donnée une base β, on peut changer la formulation (EQ) en un tableau de manière à ce que l'éventuelle solution basique correspondante puisse être lue sur le tableau. Puisque C_β est invertible, on peut substituer à (EQ)

$$C_\beta{}^{-1}Ct = C_\beta{}^{-1}q$$

$$C_\beta{}^{-1}\textstyle\sum_{1\leq j\leq 2m} C_j t_j = C_\beta{}^{-1}q$$

$$C_\beta{}^{-1}\textstyle\sum_{i\in\beta} C_i t_i = C_\beta{}^{-1}q - \sum_{j\notin\beta} C_\beta{}^{-1}C_j t_j$$

$$t_\beta = C_\beta{}^{-1}q - \textstyle\sum_{j\notin\beta} C_\beta{}^{-1}C_j t_j$$

Exemple
Dans notre jeu bimatriciel favori, on peut prendre pour base $\beta = \{w'_1, w'_2, w'_3, z'_4, z'_5\}$. Dans ce cas, le tableau est aisé à formuler :

$$w'_1 = 1 - 6y'_5$$

$$w'_2 = 1 - 2y'_4 - 5y'_5$$

$$w'_3 = 1 - 3y'_4 - 3y'_5$$

$$z'_4 = 1 - x'_1 - 4x'_3$$

$$z'_5 = 1 - 2x'_2 - 3x'_3$$

Une solution basique faisable est donc équivalente aux conditions (3')-(4') et (7') ; qu'en est-il des conditions de complémentarité, à savoir (5') et (6') ?

Définition

Une base β est dite **complémentaire** si pour toute paire (x'_i, w'_i) (resp. (y'_j, z'_j)) exactement l'une des composante est dans la base.

A présent, on peut expliquer les idées principales de ce que l'on appelle l'algorithme du Pivot Complémentaire : on part d'une solution basique faisable à **(CLP3)** et l'on change la base jusqu'à ce que l'on trouve une base complémentaire, tout en maintenant la faisabilité. Cet algorithme présente donc de nombreuses affinités avec l'algorithme du Simplexe ; la différence principale est que l'on a plus de véritable fonction objectif à améliorer progressivement, mais une paire de vecteurs complémentaires à trouver.

L'algorithme consiste essentiellement en la description de la bonne façon de pivoter, c'est-à-dire de la bonne façon de changer de base ; quand une variable basique t_j pour $j \notin \beta$ entre dans la base, une variable basique t_i pour $i \in \beta$ la quitte et l'on note la nouvelle base $\beta \bigcup \{j\} - \{i\}$.

Il existe deux règles principales dans l'algorithme :

- pivot complémentaire : quand une variable quitte la base, la prochaine variable qui entre dans la base est sa variable complémentaire
- le *minimum ratio test* : soit $c = (c_i)_{i\in\beta} = {C_\beta}^{-1}q$ et $d = (d_i)_{i\in\beta} = {C_\beta}^{-1}C_j r_j$ Alors le test sélectionne la variable r_i t.q. c_i/d_i est le minimum (et bien sûr $d_i > 0$). Cela permet de préserver la faisabilité de la solution ie $z \geq \mathbf{0}$

Désormais, on peut donner la description exhaustive de l'algorithme :

algorithme du Pivot Complémentaire
(a) initialiser avec $z' = F^\top$ et $w' = E^\top$- ie toutes les "vraies" variables ont pour valeur initiale 0
(b) choisir un $k \in S_1 \bigcup S_2$ et entrer dans la base la variable correspondante
(c) déterminer la variable sortante selon le *minimum ratio test*
(d) si la variable s_i qui a juste quitté la base est indexée par k, aller à (e) ; autrement, la prochaine variable à entrer est le complément de s_i et aller à (c)
(e) à partir des valeurs résultantes (x', y'), calculer les valeurs (x, y)

Exemple

On va donner une exécution de l'algorithme pour notre exemple ; on mettra en regard l'exécution de l'algorithme de Shapley.

$w'_1 = 1 - 6y'_5$
$w'_2 = 1 - 2y'_4 - 5y'_5$
$w'_3 = 1 - 3y'_4 - 3y'_5$

$z_4' = 1 - x_1' - 4x_3'$
$z_5' = 1_2 x_2' - 3x_3'$

✓supposons que l'on choisisse comme étiquette 2 [même chose pour le graphe] ;

✓la première variable à entrer est x_2' [2 n'étiquette plus la stratégie correspondant au point du graphe] ;

✓ici nous n'avons plus le choix : la variable sortante est z_5' [5 devient une étiquette pour la stratégie du joueur 1]

✓par la règle du pivot complémentaire, la variable à entrer est y_5' [5 n'étiquette plus la stratégie du joueur 2] ;

✓par le *minimum ratio test*, w_1' doit quitter la base [1 devient une étiquette pour le joueur 2]

$y_5' = 1/6 - 1/6w_1'$
$w_2' = 1/6 - 2y_4' + 5/6w_1'$
$w_3' = 1/2 - 3y_4' + 1/2w_1'$

✓et x_1' entre dans la base [1 n'étiquette plus la stratégie du joueur 1]

✓la variable sortante est alors z_4' [4 n'étiquette plus la stratégie du joueur 2]

$$\boxed{\begin{array}{l} x_1' = 1 - 4x_3' - z_4' \\ x_2' = 1/2 - 3/2x_3' - 1/2z_5' \end{array}}$$

✓enfin w_2' doit quitter la base ; donc l'algorithme arrête l'itération [nous arrivons à une paire complètement étiquettée]

$$\boxed{\begin{array}{l} y_5' = 1/6 - 1/6w_1' \\ y_4' = 1/12 + 5/12w_1' - 1/2w_2' \\ w_3' = 1/4 - 3/4w_1' - 3/2w_2' \end{array}}$$

✓il reste alors l'étape (e) ie la renormalisation des variables : puisque $x_1' = 1, x_2' = 1/2, x_3' = 0, y_4' = 1/12$ et $y_5' = 1/6$, et on obtient la paire de stratégies mixtes bien connue désormais $((2/3, 1/3, 0), (1/3, 2/3))$. $\diamondsuit$

1.3 Discussion et perspectives

1.3.1 Quelques questions laissées ouvertes par la description de l'algorithme

(1) La première question qui se pose est celle de savoir comment se comporte l'algorithme que l'on vient de décrire.

• Les deux règles que nous avons mises en avant, la règle du pivot complémentaire et le *minimum ratio test* sont elles univoques ?

La réponse est positive pour la première, mais ce n'est pas le cas pour la seconde : il est tout à fait possible que plusieurs variables satisfassent le test. Supposons que l'on en choisisse une pour quitter la base ; alors celle qui reste dans la base aura une valeur nulle. C'est pourquoi l'Hypothèse des Variables Basiques Nonnulles (HVBN) est une condition suffisante pour que l'algorithme tel que nous l'avons exposé fonctionne. Le parallélisme avec l'algorithme de Shapley vaut toujours, puisque l'on montre que (HBVN) et (HND) sont équivalentes ([vS 95]).

Si (HVBN) n'est pas satisfaite, on utilise une variante de l'algorithme inspirée de remèdes que l'on a proposé à semblables maux dont souffre l'algorithme du Simplexe (il utilise aussi le *minimum ratio test*, cf [C 83], pp. 35 and sq.)

• Quelle est la performance de l'algorithme de Lemke-Howson ?

Il a été montré que l'algorithme termine en temps exponentiel sur certaines instances (Murty, 1978)

(2) La seconde question qui se pose est celle de savoir quand on ne se restreint plus au cas particulier de l'équilibre de Nash dans un jeu à 2 joueurs. Les trois grands types d'extension cherchent à savoir comment traiter

• les jeux sous forme extensive

• les jeux à N joueurs

• les autres notions d'équilibre

On n'abordera pas ces questions ici ; on trouve dans [vS 02] et surtout [vS 95] une exposition des difficultés et des progrès propres à chaque type d'extension.

1.3.2 Interprétation du calcul des équilibres : qui calcule les équilibres ?

De quel point de vue le calcul des équilibres est-il pertinent ? Le point de vue du modélisateur, le point de vue du joueur, les deux, ou ni l'un ni l'autre ?

La question entretient un rapport étroit avec la divergence interprétative devenue classique en théorie des jeux qui oppose une interprétation dite *éductive* et une interprétation *évolutive* des équilibres.

Dans la première intérprétation, "les joueurs trouvent leur chemin vers l'équilibre grâce à un raisonnement méthodique" ([B 92], p.475) ; à l'opposé, l'interprétation évolutive considère "l'équilibre atteint, non pas comme le résultat d'une réflexion préalable des joueurs, mais comme une conséquence de l'ajustement de comportement par essai-et-erreur de joueurs myopes lorsqu'ils jouent un jeu répété éventuellement sur de longues périodes" (*ibid.*).

Bien sûr, ce ne sont pas les deux seules interprétations possibles mais plutôt les deux extrémités d'un spectre d'attitudes cognitives possibles. Le point important est que selon l'interprétation qui prévaut, la pertinence du calcul des équilibres est très différente.

Si l'on accepte l'interprétation éductive, alors les questions essentielles sont celles de savoir quel genre de procédure délibérative les agents suivent pour mettre en oeuvre leur raisonnement pratique et de quelles informations ils ont besoin pour faire aboutir leur délibération. L'analyse algorithmique des équilibres est pertinente pour la première de ces deux questions puisqu'elle donne précisément des renseignements sur les méthodes qui peuvent sous-tendre les procédures délibératives.

Si l'on rejette l'interprétation éductive, alors il est probable que l'on ne conçoive pas les algorithmes permettant de trouver des équilibres comme des descriptions possibles du raisonnement qui conduit les joueurs à atteindre l'équilibre. Dans ce cas, les algorithmes nous renseignent plutôt sur les méthodes qu'un modélisateur pourraient employer pour faire une prédiction sur le jeu. Dans cette interprétation, la situation serait analogue à celle d'un physicien qui veut se renseigner sur les algorithmes qui permettent de calculer effectivement des prédictions à partir du modèle mathématique qu'il a élaboré.

1.4 Références

[B 92] K.Binmore, Fun and Games, D.C.Heat and Co, 1992 ; trad.fr. Jeux et théorie des jeux, De Boeck, 1997

[CSP 92] Cottle, Pang et Stone, The linear complementarity problem, Academic Press 1992

[C 83] Chvatal, Linear Programming, Fremman and Company, New York, 1983

[KL 96] Mc Kelvey et Mc Lennan, "Computation of Equilibria in Finite Games", in Amman and ali (eds) *Handbook of Computational Economics*, vol.I, pp.87-142, Elsevier, Amsterdam 1996

[LH 64] Lemke et Howson, "Equilibrium points in bimatrix games", in*Journal of the Society for Industrial and Applied Mathematics*Vol.12, N°2, June 1964

[S 74] L.Shapley, "A Note on the Lemke-Howson Algorithm", *Mathematical Programming Study I*, 1974, pp. 175-189

[vS 95] von Stengel, "Computing Equilibria for two-person games", Technical Report n°253, Département d'Informatique,ETH Zürich, 1995 [vS 02] von Stengel, "Computing Equilibria for two-person games" in Aumann and Hart (eds) *Handbook of Game Theory*, vol.III, North-Holland, Amsterdam, 2002

Chapitre 2

Automates finis et jeux répétés

2.1 Introduction

2.1.1 Les jeux répétés

L'une des limitations majeures d'un jeu tel que nous l'avons défini jusqu'à présent, c'est-à-dire sous forme normale, réside dans son incapacité à prendre en compte les interactions à long terme qui sont liées à la répétition d'une même situation ; et donc son incapacité à capturer des notions comme celle de menace, de sanction ou de coopération. Parmi les motivations essentielles de la théorie des jeux répétés, il y a une intuition qui cherche à rendre compte de la stabilité de normes socialement désirables, mais individuellement apparemment inexplicables en termes de comportement rationnel, par la prévoyance et la réciprocité.

On va essayer d'expliciter cette intuition et de voir comment elle conduit à la théorie des jeux répétés en analysant le fameux Dilemme du Prisonnier.

Sous sa forme standard, le Dilemme du Prisonnier est un jeu à deux joueurs où chaque joueur peut choisir entre deux actions ; il peut ou bien trahir l'autre joueur (action D), ou bien coopérer avec lui (action C). Les paiements sont les suivants :

I / II	C	D
C	$(3,3)$	$(0,4)$
D	$(4,0)$	$(1,1)$

Dans ce jeu, il est clair qu'il est irrationnel pour un joueur de coopérer : jouer D est une stratégie dominante puisque, quoique fasse l'adversaire, D est préférable à C . (D,D) est un équilibre en stratégies dominantes ; c'est également un équilibre de Nash.

Il est clair également que (C,C) est une issue "socialement" meilleure que l'équilibre (D,D), au moins dans le sens où chacun des joueurs préfère la première issue à la seconde. On dit que (C,C) Pareto-domine (D,D) ; et cela constitue une raison assez claire pour que les choses *ne se passent pas toujours* comme l'équilibre le prescrit. C'est la raison pour laquelle on a vu dans le Dilemme du Prisonnier une illustration extrêmement simple de la façon dont l'intérêt d'agents en interaction pouvait paralyser les possibilités de coopération ; un phénomène que l'on peut modéliser ainsi est, par exemple, la course à l'armement entre deux nations.

Est-ce qu'il existe des conditions dans lesquelles la coopération peut émerger ? C'est-à-dire des conditions où des issues socialement préférables peuvent être atteintes à l'équilibre.

Une des tentatives de résolution les plus fructueuses du Dilemme part de l'idée extrêmement simple, et manifeste dans beaucoup de nos comportements quotidiens, selon laquelle un agent peut avoir intérêt à coopérer avec un autre agent quand il pense qu'il aura par la suite à nouveau l'occasion d'interagir avec lui. Typiquement, une bonne raison pour ne pas m'enfuir devant mon boulanger avec mes pains au chocolat alors que je ne l'ai pas encore payé, c'est que demain matin, si je lui en demande de nouveau, j'ai bien peu de chances d'être servi. L'argument général est donc le suivant : dans bien des situations, quand un agent semble agir contre son intérêt, c'est en fait seulement contre son intérêt *immédiat*, mais conformément à son intérêt à plus long terme, qu'il agit.

Pour vérifier si cette intuition est la bonne, on définit formellement à partir d'un jeu de départ G la répétition de G ; on distingue le cas où G est finiment répété (horizon fini, section 2.2) du cas où il ne l'est pas (horizon infini, section 2.3). Dans la section 2.4, on présentera le modèle des restrictions computationnelles dans les jeux à horizon fini, avec ses résultats élémentaires ; dans la section 2.5 on verra comment ce modèle permet de retrouver en horizon fini certains résultats attractifs qui sont normalement l'apanage des jeux à horizon infini. Enfin, dans la section 2.6, on discutera de la portée de ces restrictions computationnelles.

2.2 Jeux répétés à horizon fini

Si G est un jeu sous forme stratégique, un jeu répété G^t est un jeu à t étapes où, à chaque étape, chaque joueur i a l'opportunité de jouer ce qu'il peut jouer au jeu G soit S_i. En outre, dans le modèle le plus élémentaire, au cours de la partie, les joueurs sont parfaitement au courant de ce que chacun a joué au cours des actions précédentes.

Dans le jeu G, actions et stratégies sont indistinctes ; ce n'est plus le cas dans le jeu G^t.

Par exemple, dans le Dilemme du Prisonnier répété, une stratégie, appelons la MAE ("mise à l'épreuve") peut consister à coopérer spontanément et ensuite à coopérer tant que l'autre joueur coopère ; et à trahir tant que l'adversaire ne coopère pas trois fois de suite. Sur un tel exemple, on voit qu'une stratégie est une certaine fonction qui à chaque étape associe une action en fonction du déroulement passé du jeu. Pour définir les stratégies dans un jeu répété, on capture donc d'abord cette notion de déroulement passé d'une partie.

Les paiements des joueurs sont ensuite définis comme la moyenne arithmétique des paiements reçus à chaque étape.

Définitions

Soit G un jeu sous forme normale ;

• une **histoire à l'étape** k de la répétition de G, notée H^k est un élément de S^{k-1} ; on pose $S^0 = \emptyset$.

• l'ensemble des **stratégies pures du joueur** i **dans le jeu** G^t est l'ensemble des fontions $\sigma_i : \bigcup_{k=1}^{t} H^k \to S_i$

• chaque profil $\sigma = (\sigma_1, ..., \sigma_N)$ de stratégies pures pour G^t induit une partie à t étapes $\omega(\sigma) = (\omega_1(\sigma), ..., \omega_t(\sigma))$ définie comme suit :

$\omega_1(\sigma) = \sigma(\emptyset) = (\sigma_1(\emptyset), ..., \sigma_N(\emptyset))$ et

$\omega_k(\sigma) = \sigma(\omega_1(\sigma), ..., \omega_{k-1}(\sigma)) = (\sigma_1(\omega_1(\sigma), ..., \omega_{k-1}(\sigma)), ..., \sigma_N(\omega_1(\sigma), ..., \omega_{k-1}(\sigma))$

• soit σ un profil de stratégies pures pour G^t ; on définit alors les paiments du joueur i quand σ est joué comme

$u_i^t(\sigma) = (\sum_{k=1}^{\infty} u_i(\omega_k(\sigma)))/t$

2.3 Jeux répétés à horizon infini

La définition des stratégies pures de la répétition infinie d'un jeu G s'effectue d'une manière tout à fait analogue au cas fini. Par contre, la définition des paiements est plus délicate, ne serait-ce que parce que rien ne garantit que la somme des paiements à chaque étape ne converge ; en outre, on souhaite souvent introduire un facteur d'escompte. On notera G^∞ la répétition de G sans facteur d'escompte et G^λ la répétition avec un facteur d'escompte $\lambda \in]0, 1]$.

Définitions

Soit G un jeu sous forme normale ;

• l'ensemble des **stratégies pures du joueur** i **dans les jeux** G^∞ **et** G^λ est l'ensemble des fontions $\sigma_i : \bigcup_{k=1}^{\infty} H^k \to S_i$

• soit σ un profil de stratégies pures pour G^∞ (resp. G^λ) ; quand σ est joué, on définit alors **les paiements du joueur** i **dans** G^∞ **(resp.** G^λ**)** de la manière suivante :

$u_i^\infty(\sigma) = \lim_{t\to\infty} \sum_{k=1}^{\infty} u_i(\omega_k(\sigma))/t$

$u_i^\lambda(\sigma) = \sum_{k=1}^{\infty} \lambda.(1-\lambda)^{k-1}.u_i(\omega_k(\sigma))$

2.3.1 Le *Folk Theorem*

Revenons maintenant aux motivations de départ qui commandent la théorie des jeux répétés, c'est-à-dire la compréhension de l'attrait que peuvent exercer certaines issues du jeu de base ((C, C) dans le Dillemme) sans qu'elles soient des équilibres de ce jeu. La théorie atteint-elle son objectif ? Est-ce que, par exemple, dans le Dilemme du Prisonnier, la coopération mutuelle devient un équilibre quand on répète le Dilemme ?

Dans le cas des jeux à horizon infini, la réponse est positive : c'est le contenu de ce qu'on appelle le *Folk Theorem*.

Définitions

Soit G un jeu et i un joueur ;

• w est un **paiement de G** s'il existe un profil de stratégies s tel que pour tout i, $w_i = u_i(s)$.

• w est un **paiement faisable de G** si w est la combinaison convexe de paiements de G soit $w \in co(u(S))$; on note $w \in F(G)$.

• le **paiement minimax** du joueur i dans G, noté v_i est le plus petit paiement que les autres joueurs peuvent forcer i à recevoir soit

$v_i = \min_{s_-i \in S_-i} \max_{s_i \in S_i} u_i(s_{-i}, s_i)$.

On note p_{-i} un des profils d'actions des autres joueurs capables de produire v_i.

• soit w paiement faisable de G ; w est **individuellement rationnel** si pour tout i, $w_i \geq v_i$.

Théorème (*Folk Theorem*)

Tout profil de paiement individuellement rationnel d'un jeu G est le profil de paiement d'un équilibre de Nash de G^∞.

Preuve

Pour tout paiement individuellement rationnel, on donne un méthode pour construire un type de stratégie qui constitue un équilibre de Nash dans G^∞ ; on appelera cette stratégie "Impitoyable" (notée IMPIT).

IMPIT consiste à joueur spontanément ce qu'il faut jouer pour produire le paiement et à le faire tant que les autres joueurs le font également ; si un joueur dévie, il est puni par (une action correspondant à) son minimax, et cela irrémédiablement. Dans le Dilemme du Prisonnier, si le paiement visé est celui de la coopération, IMPIT consistera à coopérer tant que l'adversaire coopère lui aussi, et à faire défaut irrévocablement dès que l'adversaire ne coopère plus.

Dans le cas général, la démonstration s'appuie sur le lemme suivant ([S 92]) :

Lemme

Quand t tend vers l'infini, $F(G^t)$ converge vers $F(G)$

On se restreindra ici, comme dans [OR 94], au cas des combinaisons convexes *rationnelles*. Soit $w \in F(G)$; alors il existe des rationnels α_s tels que $w = \sum_{s \in S} \alpha_s u(s)$. Il existe γ et des entiers β_s tels que $\alpha_s = \beta_s/\gamma$ et $\gamma = \sum_{s \in S} \beta_s$; clairement, une partie de G^∞ composée de cycles de longueur γ où les joueurs jouent β_s fois le profil d'actions s induit un paiement moyen de w.

La stratégie IMPIT dans le jeu G^∞ consiste à jouer les actions du cycle précédemment défini tant que les autres joueurs font de même ; dès qu'un joueur j dévie de ce cycle, chaque joueur i joue la stratégie de punition soit $(p_{-j})_i$. On vérifie facilement que si tous les joueurs jouent IMPIT, on obtient un équilibre de Nash dont le paiement est égal à w : aucun joueur n'a intérêt à dévier car il obtiendrait un paiement moyen proche de son minimax. ♠

Le principal fait qui motive les restrictions computationnelles des stratégies dans les jeux répétés à horizon fini est qu'*il n'y a pas d'équivalent exact du Folk Theorem quand on passe en horizon fini.* Dans certaines situations, la menace d'une punition infinie est effectivement nécessaire pour faire émerger un équilibre : sans elle, les joueurs peuvent avoir intérêt à dévier lors du dernier coup, puisqu'ils agiront alors impunément.

Exemple

Considérons le paiement $(3,3)$ dans le Dilemme du Prisonnier. L'argument du *Folk Theorem* ne marche plus ici : face à la stratégie IMPIT, il existe une meilleure réponse que la stratégie IMPIT ; on va la nommer ANTI-IMPIT. Dans G^n, joueur IMPIT contre IMPIT rapporte un paiement de 3 ; par contre, jouer C $n-1$ fois puis D au dernier coup (ANTI-IMPIT) rapporte $(3(n-1)+4)/n$, ce qui est supérieur. IMPIT n'est donc pas une meilleure réponse. ♢

2.4 Restrictions computationnelles des stratégies

Les jeux répétés à horizon infini permettent une multiplication spectaculaire des équilibres ; existe-t-il un moyen *dûment motivé* d'obtenir la même chose dans des répétitions à horizon fini ?

L'étude des restrictions computationnelles des stratégies tente de donner une réponse positive à cette question en prenant en compte la complexité des règles de décision que les agents peuvent mettre en oeuvre (ou complexité stratégique).

2.4.1 Les stratégies comme automates finis

Dans cette sous-section, on expose la mesure la plus communément acceptée de complexité stratégique.

L'idée qui est à la base de cette approche est comparable à celle que l'on trouve dans la complexité de Kolmogorov en théorie de l'information : de même que la complexité informationnelle d'une suite de caractères est la taille du plus petit programme capable de l'engendrer, la complexité d'une stratégie est la taille du plus petit automate capable de l'implémenter.

Définitions

• une **machine pour le joueur** i est un 4-uplet $m_i =< Q, q_0, f, t >$ où

Q est l'ensemble des états

$q_0 \in Q$ est l'état initial

$f : Q \to S_i$ est la fonction de sortie

$\tau : Q \times S_{-i} \to Q$ est la fonction de transition

On note M_i l'ensemble des machines pour le joueur i.

• une **machine aléatoire pour le joueur** i est une distribution de probabilités sur l'ensemble des machines pour le joueur i soit un élément de $\Delta(M_i)$

Exemples

⋆ les machines les plus simples sont évidemment celles qui jouent inconditionnellement une stratégie pure du jeu G ; elles peuvent n'avoir qu'un seul état, lequel est associé à la stratégie pure par la fonction de sortie.

⋆ la stratégie IMPIT est elle aussi facilement implémentable ; dans le cas du Dilemme du Prisonnier, quand le paiement visé est celui de la coopération, IMPIT est implémentable par un automate à deux états :
- un premier état, qui est l'état initial, joue C tant que l'adversaire le joue également ; dès que celui-ci joue D, l'automate passe au
- second état, qui joue D inconditionnellement

⋆ la stratégie ANTI-IMPIT requiert elle plus d'états des machines qui l'implémentent parce qu'elle doit *compter* les coups ; dans le cas du Dilemme, l'automate minimal l'implémentant est fonction de la durée du jeu : pour un jeu à t étapes, il comporte
- $t - 1$ états qui jouent la coopération
- 1 état (le dernier atteint contre IMPIT) qui joue la trahison

$\star$ une stratégie rendue célèbre par les expérimentations de Axelrod ([A 84]) est OPO (Oeil Pour Oeil) dans le Dilemme du Prisonnier : elle consiste à coopérer tant que l'autre coopère, trahir quand il trahit et coopérer à nouveau dès qu'il cesse de trahir.

On remarque que face à une suite de t actions des autres joueurs $s_{-i} =< s^1_{-i}, ..., s^t_{-i} >$,

(1) le comportement d'une machine pour i est déterminé inductivement :

$\tau(q_0, s^1_{-i}, ..., s^t_{-i}) = \tau(\tau(q_0, s^1_{-i}, ..., s^{t-1}_{-i}), s^t_{-i})$

(2) ce comportement induit une stratégie pour le joueur i,

$\sigma_i^{M_i}(s^1, ..., s^{t-1}) = f(\tau(q_0, s^1_{-i}, ..., s^t_{-i}))$.

On dit qu'une stratégie σ_i est **implémentée** par une machine M_i si $\sigma_i = \sigma_i^{M_i}$; par abus de langage, on désignera souvent $\sigma_i^{M_i}$ simplement par M_i.

La mesure de complexité des stratégies commmunément choisie est celle de la taille des automates ; **on note $c(M_i)$ la taille de l'automate** M_i.

Si l'on revient sur la taille des automates implémentant IMPIT et ANTI-IMPIT, on se rend compte que la déviation au dernier coup, caractéristique de ANTI-IMPIT, a un coût en termes de complexité : pour compter, ANTI-IMPIT doit comporter t états.

Maintenant, si l'on restreint l'ensemble des stratégies pures des joueurs à des stratégies pouvant être implémentées par des machines de taille strictement inférieure à t, ANTI-IMPIT est disqualifiée et ne peut plus concurrencer IMPIT contre IMPIT. Cette observation suggère donc un rapport possible entre la taille des automates et les meilleures réponses ; comme l'ensemble des meilleures réponses à un profil donné est la base de la notion d'équilibre, on entrevoit la possibilité de contrôler les équilibres en jouant sur la taille des automates.

Peut-on généraliser ce phénomène ? Dans le cas du Dilemme du Prisonnier par exemple, est-il possible que, quand on se restreint aux stratégies les plus simples, on puisse récupérer la coopération à l'équilibre ?

L'attrait d'une telle démarche, c'est que l'on peut motiver ce genre de restriction en termes de "rationalité limitée" (H.Simon) : il n'est pas raisonnable de s'attendre à ce que des agents dépensent leurs ressources calculatoires immodérément pour améliorer très légèrement leur paiement [PY 94]. Remarquons toutefois que, même dans la perspective très particulière d'une implémentation des stratégies par des automates finis, la restriction de la taille des machines n'est pas la seule interprétation possible de cette idée : on pourrait également *endogénéiser* le coût calculatoire et intégrer dans le modèle sa mise en balance avec le gain attendu. Le *machine game* de [AR 88] (cf [R 98]) procède de cette façon.

Si l'on replace la récupération de la coopération dans le Dilemme du Prisonnier dans le contexte plus large du *Folk Theorem*, voici donc la question qui commmande les recherches sur les restrictions des stratégies, et sa généralisation :

1. quelles bornes sur l'espace des stratégies possibles des joueurs permettent de faire emerger la coopération comme équilibre de Nash dans le dilemme du prisonnier finiment répété ?
2. quelles bornes sur l'espace des stratégies possibles des joueurs permettent, dans un jeu quelconque G, pour un profil de paiement individuellement rationnel quelconque w, d'obtenir dans une répétition finie de G un équilibre dont le paiement est égal à w ?

Notations

- dans une répétition finie de G on note $r_i(t)$ la restriction imposée au joueur i sur la taille des machines implémentant ses stratégies dans un jeu à t répétitions
- soit G^t un jeu répété et s_{-i} un profil de stratégies de tous les joueurs sauf i ; on note $br(r_i, s_{-i})$ l'ensemble des stratégies de i qui sont (i)implémentables par un automate de taille inférieure ou égale à $r_i(t)$ et (ii) les meilleures réponses de i à s_{-i} parmi ses stratégies de taille inférieures ou égale à r_i
- si G est un jeu, on note $Eq(G)$ l'ensemble des équilibres de Nash de G et $pEq(G)$ l'ensemble des profils de paiements des équilibres de G
- si G^t est un jeu répété, on note $G^t(r_1, ..., r_N)$ le jeu répété dont (i) les stratégies sont celles de G^t qui satisfont aux restrictions r_i et (ii) les paiements sont la trace des paiements de G^t par les restrictions

Il est aisé de donner une taille à partir de laquelle les restrictions n'ont plus d'effet :

Proposition([N 98])

Toute stratégie pure d'un joueur i dans G^t peut être implémentée par un automate de taille $\sum_{k=1}^{k=t} |S_{-i}|^{k-1}$

Preuve

Relativement à un joueur i, on peut définir la notion d'**histoire des autres** qui désigne les différents comportements possibles des autres joueurs au cours de la partie : $HA(i) = \bigcup_{k=1}^{k=t} HA^k(i)$ où $HA^0 = \{\emptyset\}$ et $HA^k = S_{-i}^{k-1}$. La Proposition repose sur le fait qu'une machine dont les états sont les histoires des autres peut reproduire n'importe quelle stratégie, pourvu que la fonction f soit bien choisie.

Soit σ_i une stratégie du joueur i ; soit la machine M_i où pour tout $h \in HA^k(i)$, $1 \leq k \leq t$,

$Q = HA(i)$

$q_0 = \emptyset$

$\tau(h, s_{-i}) = (h, s_{-i})$

$f(h) = \sigma_i(h')$

où h' est l'unique histoire à l'étape k qui soit compatible avec h et σ. Clairement, M_i implémente σ_i. ♠

Commentaire : de tels automates peuvent être représentés par des arbres $|S_{-i}|$-aires complets où les noeuds sont les états et les branches les transitions.

Exemple

Dans un Dilemme du Prisonnier itéré t fois, toute stratégie pure peut être implémentée par un automate de taille $1 + 2 + 2^2 + ... + 2^{t-1}$ ♢

Corollaire

Pour tout jeu G il existe une constante c telle que si pour tout i, $r_i \geq \exp(c.t)$, alors $pEq(G^t(r_1, ..., r_N)) \subseteq pEq(G^t)$.

Preuve

Soit $s^* \in Eq(G^t(r_1, ..., r_N))$; est-ce que $u(s^*) \in pEq(G^t)$?

Pour que cela soit le cas, il suffit que pour tout joueur i, $s_i^* \in br(s_{-i}^*) = \{s_i :$ s_i est une meilleure réponse à $s_{-i}^*\}$. Pour que cela soit vrai, il suffit que toutes les stratégies pures dans le support de s_i^* appartiennent à $br(s_{-i}^*)$; par hypothèse, on sait qu'elles appartiennent à $br(r_i, s_{-i}^*) = \{s_i : s_i$ est implémentable par une machine de taille r_i et est une meilleure réponse à $s_{-i}^*\}$. Par la Proposition précédente, si $r_i \geq \sum_{k=1}^{k=t} |S_{-i}|^{k-1}$, $br(s_{-i}^*) = br(r_i, s_{-i}^*)$.

Donc si pour tout i, $r_i \geq \max_{i \in N} \sum_{k=1}^{k=t} |S_{-i}|^{k-1}$, alors $pEq(G^t(r_1, ..., r_N)) \subseteq pEq(G^t)$. La constante c doit satisfaire l'inégalité suivante :

$c \geq (\log(\max_{i \in N} \sum_{k=1}^{k=t} |S_{-i}|^{k-1}))/t$.

♠

A l'opposé, si l'on impose des restrictions très fortes sur les stratégies, on peut par exemple obtenir la coopération dans le Dilemme du Prisonnier.

Proposition

Si $r_1, r_2 < t$, alors (OPO, OPO) est un équilibre de Nash et induit la partie $t * (C, C)$ donc le paiement 3 pour chaque joueur.

Preuve

Une réponse à OPO meilleure qu'OPO doit dévier (jouer D) au dernier coup mais pas avant ; il faut donc une partie de type $(t-1) * (C, C) + (C, D)$. Mais cela exige du joueur qui dévie de compter les coups jusqu'à la déviation et donc de ne pouvoir être implémenté par une machine de moins de t états. ♠

Remarque : le même raisonnement vaut pour le profil (IMPIT, IMPIT).

2.5 Résultats principaux

2.5.1 Introduction

Dans cette section, on expose les résultats principaux de l'étude des restrictions computationnelles des stratégies ; commençons par en donner les objectifs et la méthode.

Revenons au Dilemme du Prisonnier et supposons que l'on ne cherche plus exactement la paiement coopératif $(3,3)$ mais que l'on se contente d'une *approximation* du paiement ; faut-il encore imposer des restrictions aussi sévères que celles de la Proposition précédente ?
De manière générale, étant donné un paiement w individuellement rationnel d'un jeu G, à combien peut s'élever la complexité r_i des stratégies mises en oeuvres par les joueurs pour que dans le jeu $G^t(r_1, ..., r_N)$ il existe un équilibre s^* tel que $|u^t(s^*) - w| < \varepsilon$?

Pour répondre à cette question, la méthode générale consiste à construire un profil d'automates (éventuellement aléatoires) $M = (M_1, ..., M_N)$ et à montrer que pour tout i

- $c(M_i) \leq r_i$ - les automates satisfont les restrictions computationelles
- $M_i \in br(r_i, M_{-i})$ - la stratégie M_i est une meilleure réponse dans le jeu $G^t(r_1, ..., r_N)$
- $|u^t(M) - w| < \varepsilon$ - l'équilibre produit une bonne approximation du paiement recherché

2.5.2 Le Dilemme du Prisonnier

Théorème ([PY 1994])

soit G^t un Dilemme du Prisonnier répété t fois ; pour tout $\epsilon > 0$, si $r_1(t)$ ou $r_2(t) \leq 2^{c_\epsilon t}$ avec $c_\epsilon = \frac{\epsilon}{12(1+\epsilon)}$, alors pour un t assez grand, il existe dans $G^t(r_1, r_2)$ un équilibre de Nash de paiement moyen au moins égal à $3 - \epsilon$

Preuve

L'idée de la preuve repose sur l'observation de la manière dont IMPIT peut être contré par ANTI-IMPIT, c'est-à-dire en faisant défection au dernier coup. Pour faire défection, il faut cependant compter, ce qui exige des états ; les automates que l'on va construire vont être tels que, pour faire défection, ils vont avoir besoin de tant d'états que leur taille excèdera la borne.

On peut supposer que $r_1(t), r_2(t) \geq t-1$ et sans perte de généralité que $r_1(t) \leq 2^{c_\epsilon t}$. On va alors poser :

- d est le plus grand entier tel que $2^d d(2+\frac{2}{\epsilon}) \leq r_1(t)$

- l est le plus grand entier tel que $2^d(d+l+2) \leq r_1(t)$

- $R = r_1(t) - 2^d(d+l+2)$

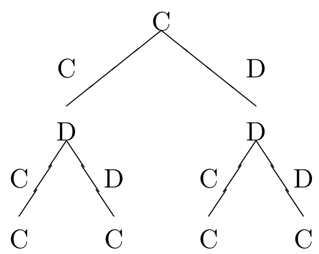
La description de l'équilibre est la suivante ; on va d'abord construire deux machines aléatoires différentes pour chacun des joueurs.

- la stratégie mixte m_1 attribue une probabilité 1/2 aux deux automates A_1 et A_2
- la stratégie mixte m_2 attribue une probabilité égale 2^{-d} aux 2^d automates B_y, $y \in \{C,D\}^d$

(i) Description de la partie attendue

<u>Phase 1</u> : d étapes
dans la construction des automates, une étape caractéristique est l'échange de "cartes de visites" [PY 94] ou "phase de communication" [N 98] pendant laquelle les adversaires s'échangent une série d'actions C et D, chaque série pouvant être considérée comme un message. L'automate est alors un arbre binaire complet où chaque feuille est un état étiquetté conformément à la fonction de transition, et chaque branche correspond à une action possible de l'adversaire.

Voici par exemple le début d'un automate dont la carte de visite est de type $(C,D,C) \in \{C,D\}^3$:



A l'issue de la carte de visite, chaque action possible de l'adversaire conduira à un noeud différent soit 2^d transitions qui conduisent à 2^d noeuds ; par abus de langage, on appellera ces transitions issues de la carte de visite les feuilles de la carte de visite.

A_1 et A_2 commencent par une carte de visite (arbre binaire complet) de profondeur d et de types respectifs C^d et (C^{d-1}, D) ; chaque carte B_y commence par une chaîne de longueur d reliant par des transitions de type C la suite d'états de type y.

Remarque : la structure d'arbre binaire complet fait que les automates A_j enregistrent le type y de l'automate adverse qui a été tiré au sort.

Phase 2 : $d+1$ étapes

Cette phase sert à compenser les différences de paiement induites par le choix entre les automates ; l'étape $d+1$ le fait pour le choix entre A_I et $A_I I$, les étapes $d+2$ à $2d+1$, font de même pour les différences induites par les B_y.

A l'étape $d+1$, les joueurs jouent

⋆ (C,C) s'ils ont joué (C,C) à l'étape d,
⋆ (D,D) sinon.

Les étapes restantes vont dépendre essentiellement du type y de l'automate B_y choisi ;

(1) si y contient $2a$ D dans ses $d-1$ premières lettres, alors les automates jouent
$a*(D,D)+(d-a-1)*(C,C)+(D,D)$ (cas (1))

(2) si y contient $(2a{+}1)$ D dans ses $d{-}1$ premières lettres, alors les automates jouent

⋆ $(C,C)+a*(D,D)+(d-a-2)*(C,C)+(D,D)$ (cas (2a))
si les deux automates s'accordent à l'étape d ((C,C) ou (D,D))

⋆ $(a+1)*(D,D)+(d-a-2)*(C,C)+(D,D)$ (cas (2b))
si les deux automates ne s'accordent pas à l'étape d

Montrons d'abord que de telles parties permettent de gommer les différences entre automates : à l'issue des phases 1 et 2, le joueur 1 n'a pas intérêt à préférer un de ses deux automates sur l'autre, et le joueur 2 n'a pas non plus intérêt à préférer un de ses automates sur les 2^d-1 autres.

• Joueur 1 :

- dans le cas (1), la différence entre les automates A_1 et A_2 ne peut avoir d'effet qu'aux étapes d et $d+1$; l'automate A_1 remporte à ses étapes $(3+3)$ avec probabilité $1/2$ (quand la dernière lettre de y est C) et $(0{+}1)$ avec la même probabilité (quand la dernière lettre de y est D) donc l'espérance de gain est de $7/2$; l'automate A_2 remporte à ses étapes $(4+1)$ avec probabilité $1/2$ (quand la dernière lettre de y est C) et $(1+1)$ avec probabilité $1/2$ donc l'espérance de gain est également de $7/2$.

- dans le cas (2), la différence a un effet sur les étapes d, $d+1$ et $d+2$. L'automate A_1 remporte $(3+3+3)$ avec probabilité $1/2$ (quand la dernière lettre de y est C) et $(0+1+1)$ avec probabilité $1/2$ (quand la dernière lettre de y est D) soit $11/2$ d'espérance de gain. L'automate A_2 remporte $(4+1+1)$ avec probabilité $1/2$ (cas (2a)) et $(1+1+3)$ avec probabilité $1/2$ (cas (2b)) soit $11/2$ d'espérance de gain.

• Joueur 2 :

Soit un automate B_y

- dans le cas (1), y comporte $2a$ D. B_y va emporter $(2a).4 + (d - 1 - 2a).3$ dans les $d - 1$ premières étapes ; $7/2$ en moyenne pour les étapes d et $d + 1$; et enfin $(a + 1).1 + (d - a - 1).3$ pour les étapes $d + 2$ à $2d + 1$. Soit un total de $6d - 3/2$.

- dans le cas (2), y comporte $2a+1$ D. B_y va emporter $(2a+1).4+(d-2-2a).3$ dans les $d - 1$ premières étapes ; $7/2$ en moyenne pour les étapes d et $d + 1$; et enfin $(a+1).1+(d-a-1).3$ avec probabilité $1/2$ (cas (2a)) et $(a+2).1+(d-a-2).3$ avec probabilité $1/2$ (cas (2b)) ; soit un total de $6d - 3/2$.

Le paiement est donc le même en moyenne pour tous les B_y, quel que soit y, durant les phases 1 et 2.

Phase 3 : étapes $2d + 2$ à t

Les automates entrent dans un cycle de longueur l qui contient une alternance de

$\star$ $l - \hat{d} - 2$ étapes de coopération soit $l * (C, C)$ puis

$\star$ une séquence $D + f(y) + C$ avec $f : \{C, D\}^d \rightarrow S$ injective où S est l'ensemble des suites de $\{C, D\}$ contenant un nombre égal de l'un et de l'autre et de longueur $\hat{d} \geq d + \lceil \lg d \rceil$ et pair

Cette séquence joue le rôle d'un rappel de l'identité y de l'automate B_y choisi puisque par définition de la fontion f deux y différents engendrent deux $f(y)$ différents. Par définition également, les paiements sont indifférents au y choisi puisque dans tous les $f(y)$ il y a le même nombre de C et de D.

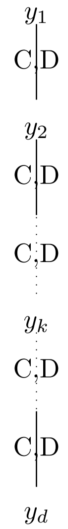
La phase 3 est la répétition du cycle suivant : $(l - \hat{d} - 2) * C + D + f(y) + C$, si ce n'est qu'à la dernière étape (l'étape t), les B_y font défaut.

Clairement, durant la phase 3, aucun des joueurs n'a intérêt à favoriser un de ses automates quand l'autre joue sa machine aléatoire. Ceci est donc établi pour toute la durée de la partie.

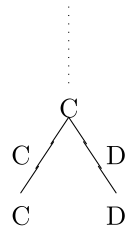
(ii) Description des automates

- Joueur 2 :

Les automates B_y sont de tailles $t + 2$ et de profondeur t ; durant la phase 1, il n'y a qu'une transition possible. Voici l'allure qu'aura cette première partie d'un automate B_y où $y = (y_1, ..., y_d)$:



Aux étapes $d{+}1$, $d{+}2$ cependant, il existe deux états possibles pour permettre les ajustements que l'on a décrit dans la phase 2. Par exemple, voici les étapes d et $d+1$ pour un B_y dont le y se termine par la lettre C.



Dans les étapes $d{+}3$ à $t{-}1$, il y a un noeud par étape ; l'automate y conduit si l'adversaire a joué le coup attendu à l'étape précédente ; sinon, l'automate joue irrévocablement D jusqu'à la fin. Ce noeud de punition est d'ailleurs atteint à la dernière étape, quoique fasse l'adversaire à l'étape $t-1$.

• Joueur 1 :

Les automates A_j sont d'abord constitués d'une carte de visite comme on l'a expliqué précédemment soit un arbre binaire complet de profondeur d, qui compte donc $2^d - 1$ noeuds. Pour les phases ultérieures, la transition attendue conduit au noeud attendu alors qu'une déviation implique une punition irrévocable ; l'automate comporte donc un état punitif duquel il est impossible de sortir. Chaque feuille issue de la carte de visite enregistre la séquence y qui y a conduit ; de chacune de ces feuilles partent les noeuds attendus soit

⋆ une chaîne de $d+1$ noeuds pour la phase 2

⋆ une boucle de l noeuds pour la phase 3

Il y a 2^d feuilles issues de la carte de visite ; les phases 2 et 3 mobilisent donc $2^d(d+l+1)$ noeuds ; à supposer toujours que $R+0$, les automates A_j sont donc de taille $2^d - 1 + 2^d(d+l+1) + 1$ soit $2^d(d+l+2)$. En vertu de la définition de l, les automates sont donc de taille inférieure à $r_1(t)$.

(iii) Vérification de l'équilibre

Il faut ensuite montrer que les deux machines aléatoires que nous venons de décrire constituent un équilibre de Nash dans le jeu $G^t(r_1, r_2)$.

On va d'abord établir les inégalités (Ineq1) et (Ineq2) dont on a besoin pour cette partie de la démonstration (partie (iii)) ainsi que la partie (iv).

Par hypothèse,

$r_1(t) \leq 2^{c_\epsilon t}$

$2^d d(2 + \frac{2}{\epsilon}) \leq r_1(t)$

donc

$2^d d(2 + \frac{2}{\epsilon}) \leq 2^{c_\epsilon t}$

$d + \log d + \log(2 + 2/\epsilon) \leq c_\epsilon t$

$d + 1 \leq c_\epsilon t$

Par ailleurs,

$2^{d+1}(d+1)(2 + \frac{2}{\epsilon}) > r_1(t) \geq n - 1$

$2(d+1)(2 + \frac{2}{\epsilon}) > (r_1(t))/2^d$

Par définition,

$d + l + 2 \leq (r_1(t))/2^d$
donc
$d + l + 2 < 2(d+1)(2 + \frac{2}{\epsilon})$

ce qui équivaut à

$d + l + 2 < (4(d+1)(1+\epsilon))/\epsilon$

$3d + 3l + 6 < (12(d+1)(1+\epsilon))/\epsilon = (d+1)/c_\epsilon$
donc
$3d + 3l + 6 < t$ (Ineq1)

$d + l + 3 > r_1(t)/2^d \geq d.(2 + 2/\epsilon)$
donc
$l.\epsilon > 2d.\epsilon + 2d - d.\epsilon - 3\epsilon$

$2l.\epsilon > 4d + 2\epsilon.(d-3)$

Si $d \geq 6$, alors $d + 2\epsilon.(d-3) \geq 6$

$4d + 2\epsilon.(d-3) \geq 3d + 6$
donc
$2l.\epsilon > 3d + 6$

$2\epsilon/3 > (d+2)/l$

Par ailleurs on a vu que

$d + 1 \leq c_\epsilon t$ soit

$d + 1 \leq \epsilon.n/12(1+\epsilon)$

ce qui implique $d < \epsilon.n/12(1+\epsilon)$

$12d < \epsilon.n/(1+\epsilon)$

$12d/n < \epsilon/(1+\epsilon)$ donc

$12d/n < \epsilon$

$4d/n < \epsilon/3$
donc
$(2+d)/l + 4d/n < \epsilon$ (Ineq 2)

On peut désormais passer à la vérification de l'équilibre proprement dite.

• *m_2 est une meilleure réponse à m_1* ; si ce n'est pas le cas, alors il doit exister une machine (déterministe) qui produit un meilleur résultat contre m_1 que n'importe laquelle des B_y ; la seule façon d'obtenir un meilleur résultat, c'est de dévier de la partie attendue. Aucune déviation n'est profitable au dernier coup puisque le profil d'actions est alors (C, D) ; supposons que l'automate dévie à l'avant-dernier coup ; alors il gagnera 4+1 points contre 3+4 à un B_y quelconque. L'argument vaut *a fortiori* pour les déviations antérieures.

• *m_1 est une meilleure réponse à m_2* : on va en fait montrer que pour tout $A' \in M_1$, si $u_1(A', m_2) > u_1(m_1, m_2)$, alors $c(A') > r_1(t)$.
(*)
Supposons que contre tous les automates B_y, l'automate A' ne dévie jamais pendant les $2d+1+l$ premières étapes ; soit S_y les noeuds de A' qui sont visités pendant les étapes $d+1$ à $2d+2l$ contre l'automate B_y

Lemme
Il existe $d+1+l$ états différents dans S_y et si $y \neq y'$, alors $S_y \bigcap S_{y'} = \emptyset$.

Définition

Soit $s \in S_y$ et supposons que s soit visité à l'étape i ; alors il existe un chemin de longueur $2d + 2l - i$ qui suit les transitions attendues de m_1 ; on appelle **chemin normal partant de** s **à l'étape** i , noté $chn(s, i)$ la suite des actions entreprises par l'automate quand il suit ce chemin. Un tel chemin normal est un mot dans l'alphabet $\{C, D\}$.

Lemme

Soit $s \in S_y$ tel qu'il existe deux étapes $i, i', i \neq i'$, $d + 1 \leq i, i' \leq 2d + 2l$ où s est visité ; alors $chn(s, i')$ est préfixe de $chn(s, i)$ (ou inversement).

Preuve

Supposons qu'il existe s qui visite i et i' sans que les chemins normaux ne soient préfixes l'un de l'autre ; et, sans perte de généralité, que $i < i'$. Soit k la première lettre de $ch(s, i')$ telle que $ch(s, i) \neq ch(s, i')$. Comme nous sommes en présence d'automate déterministe, nécessairement l'adversaire doit avoir joué deux actions différentes. Mais dans ce cas, l'une des deux actions du joueur 2 n'est pas identique à celle du joueur 1 et donc les joueurs ne jouent pas à l'unisson, ce qui est exclu par construction dans les phases 2 et 3. ♠

Preuve

A l'aide de ce dernier Lemme, on peut montrer le premier. On peut en effet vérifier que, en raison de la présence de la période de "rappel" dans la phase 3, les $d + 1 + l$ premiers chemins normaux ne peut être préfixes les uns des autres donc les $d+1+l$ premiers états de S_y doivent être distincts. [PY 94] proposent de distinguer les mots correspondants aux chemins normaux par la première occurence d'une suite de au moins $l - \hat{d} - 2 \quad C$ suivis d'un D.

Soient $d + 1 \leq i, i' \leq d + l + 1$.

- si $d + 1 \leq i, i' \leq 2d + 1$, la suite définie ci-dessus est clairement à une distance différente du début de $chn(i)$ et $chn(i')$.

- si $2d + 1 \leq i, i' \leq 2d + 1 + l$, il en est de même.

- si $d + 1 \leq i \leq 2d + 1$ et $2d + 1 \leq i' \leq (2d + 1 + l) - (\hat{d} + 2)$, il en est de même car dans $chn(i)$ la suite apparaît après au plus $d + 1$ lettre tandis que dans $ch(i')$ elle apparaît après au moins $\hat{d} + 2$ lettres.

- reste le cas où $d+1 \leq i \leq 2d+1$ et $(2d+1+l)-(\hat{d}+2) \leq i'2d+1+l$. Ici, on ne se sert plus de la distance entre le début du mot et le bloc mais du nombre exact de C dans le bloc ; nécessairement, le bloc de $chn(i)$ contient $l - \hat{d} - 2$ C car la phase 2 se termine toujours (et à dessein) par une action D. Mais le bloc de $chn(i')$ contient (au moins) $l - \hat{d} - 1$ C car le cycle de la phase 3 se termine toujours par une action C.

Enfin, l'injectivité de la fonction f assure, par un argument analogue, que pour $y \neq y'$, $S - y \neq S_{y'}$. ♠

Lemme

Apelons états intiaux les états dont on a besoin pour rendre compte des d premières étapes, l'automate A' doit nécessairement avoir $2^d - 1$ états ; en outre ces états sont distincts des S_y.

Preuve

Supposons qu'il y ait moins de $2^d - 1$ états ; alors il existe un état e et deux transitions tels que ces deux transitions mènent en e ; donc il existe deux suites y et y' telles que quand A' joue contre les automates correspondants, il ne les distingue pas. On aura donc $S_y = S'_y$, ce qui est exclu.

Supposons qu'un des $2^d - 1$ premiers états puisse être visité quand A' joue contre un certain B_y et ensuite visité à nouveau pendant après la phase 1 pour un certain $B_{y'}$, distinct ou non de B_y (donc cet état appartiendrait à un $S_{y'}$). Considérons le chemin parcouru dans A' à partir de cette seconde visite.

- ou bien le chemin sort des états initiaux en un $S_(y'')$, ce qui est exclu par le Lemme précédent

- ou bien le chemin sort des états initiaux à l'endroit même où commence $S_{y'}$; pour que l'automate ne boucle pas, il faut alors que l'adversaire fasse emprunter des transitions différentes en un même état, ce qui est exclu dans la phase 2. ♠

Jusqu'à présent, A' doit comporter $2^d - 1 + 2^d.(d+1+l) = 2^d.(d+2+l) - 1$ états.

A' ne peut dévier avec profit par rapport aux actions attendues que s'il trahit au dernier coup soit si son chemin normal de $2d + 2$ à la fin du jeu est :
$k * [(l - \hat{d} - 2) * C + D + f(y) + C] + j * C + D$
avec $k = \lfloor (l - 2d - 1)/t \rfloor$ et $j < l - d - 2$.

Par les mêmes arguments que précédemment, on voit qu'il faut $t - 2d - 2l - 1$ états supplémentaires par rapport à ceux que l'on a déjà comptés pour dévier profitablement contre un adversaire.

Dans l'hypothèse où $R = 0$ soit $2^d(d+2+l) = r_1(t)$, il est clair que, en vertu de (Ineq1), $c(A') \geq 2^d.(d + 2 + l) - 1 + t - 2d - 2l - 1 > r_1(t)$.

(**)

Supposons qu'il existe des y tel que A' dévie contre B_y avant l'étape $2d + 1 + 2l$; soit D l'ensemble des tels B_y. Contre un $B_y \in D$, A' perd au moins $n - 2d - 1 - 2l$ soit en tout au moins $|D|.(t - 2d - 1 - 2l)$. Mais avec ces déviations, A' peut gagner des états et les utiliser pour compter contre d'autres automates et ainsi dévier au dernier coup (ce qui lui fait emporter non plus 0 mais 1 au dernier coup).

Dévier contre un B_y tel que $y \in D$ peut faire économiser à A' au plus $d+1+l$ états pour les phases 2 et 3 et d pour la phase 1 soit un total de $D.(2d+1+l)$.

D'après (*), A' a besoin de $t - 2d - 2l - 1$ états pour dévier profitablement contre un B_y tel que $y \notin D$. Donc l'économie d'états permise par les déviations rapporte à A' au plus $D.(2d + 1 + l)/(t - 2d - 2l - 1)$.

On a vu précédemment que $t > 3d + 3l + 6$ donc $t - 2d - 2l - 1 > d + l + 5$ donc $2d + 1 + l < (t - 2d - 2l - 1)^2$ donc A' ne peut faire de gain de cette façon.

(iv) Vérification des paiements

Il reste enfin à montrer que le paiement approxime à ϵ près le paiement coopératif ; on le montrera pour le Joueur 2.

• Joueur 2 :

Chaque automate B_y emporte, on l'a vu, $6d - 3/2$ pendant les phases 1 et 2, soit $3(2d + 1) - 9/2$; durant la phase 3, il touche 3 sauf pendant la phase de rappel de chaque cycle où il touche 1 au début de la phase de rappel et pendant la moitié de la séquence $f(y)$. Pendant la phase 3, un automate B_y touche donc

$3(t - 2d - 1) - (\hat{d}/2 + 1).\lfloor(t - 2d - 1)/l\rfloor$

donc pendant la partie

$3t - 9/2 - (\hat{d}/2 + 1).\lfloor(t - 2d - 1)/l\rfloor$

Clairement $9/2 < 4d$; il est clair également que l'on peut toujours trouver $\hat{d}$ tel que $(\hat{d}/2 + 1) < 2 + d$ donc tel que

$(\hat{d}/2 + 1).\lfloor(t - 2d - 1)/l\rfloor < (2 + d).n/l$

aussi, en vertu de (Ineq2),

$u_2(m_1, m_2) > |3 - \epsilon|$.

♠

2.5.3 Généralisations

Le résultat précédent est très local au sens où il ne concerne qu'un seul profil de paiement dans un jeu à deux joueurs bien particulier. Le *Folk Theorem* est beaucoup plus général puisqu'il concerne les jeux à N joueurs et n'importe quel paiement pourvu qu'il soit individuellement rationnel. Est-il possible d'avancer vers une telle généralisation ?

[N 98] donne une réponse positive en proposant des résultats très généraux dans le cas des jeux à deux joueurs ; voici le premier de ces théorèmes :

Théorème([N 98])

Soit un jeu $G =< N, (S_i)_{i\in N}, (u_i)_{i\in N} >$; pour tout $\varepsilon > 0$ assez grand il existe deux entiers positifs t_0 et r_0 tels que

si $t \geq t_0$, w est un paiement strictement individuellement rationnel et $r_0 \leq \min(r_1, r_2) \leq \exp(\varepsilon^3.t)$,

alors il existe un paiement $y \in pEq(G^t(r_1, r_2))$ tel que pour tout i, $|y_i - w_i| < \varepsilon$.

2.5.4 Discussion

Si l'on revient aux motivations initiales des restrictions computationnelles, on est en droit de se demander si cette approche permet effectivement de *résoudre* un paradoxe comme celui de la justification de la coopération dans le Dilemme du Prisonnier, objectif avoué de [PY 94] et [N 98].

On savait, par le *Folk Theorem*, que le paiement coopératif peut être atteint dans les jeux répétés à horizon infini ; on sait désormais que, moyennant des restrictions raisonnables sur la taille des automates implémentant les stratégies, on peut approcher le paiement coopératif dans les jeux répétés à horizon fini.

Si l'on ne se contente pas du *Folk Theorem*, c'est que l'on pense trouver dans les jeux à horizon fini un formalisme dont l'interprétation est plus satisfaisante. La question qui se pose est donc celle de savoir si les restrictions computationnelles fournissent bien un formalisme plus satisfaisant que celui des jeux à horizon infini. Cette question dépend évidemment très étroitement de l'interprétation que peuvent recevoir les jeux à horizon infini.

L'interprétation *objective*, naïve, de l'horizon infini représente G^∞ comme la répétition infinie du jeu G ; l'interprétation *subjective* conçoit G^∞ comme l'itération d'une interaction dont aucun joueur ne connaît exactement quelle est l'étape finale ([R 98]). C'est quand on considère la première interprétation que le modèle de l'horizon infini est manifestement irréaliste ; si l'on se place dans la seconde interprétation, alors le fait que les répétitions réelles de nos interactions sont finies ne discrédite pas pour autant le modèle de l'horizon infini.

Il est cependant délicat de préciser exactement ce que signifie "une connaissance exacte de l'étape finale" et, corrélativement, de pouvoir juger de la correction du modèle à horizon infini sous l'interprétation subjective. Pour une discussion plus détaillée, cf [OR 94], préliminaires au Chap.8.

Sous une interprétation subjective charitable, on peut dire que le modèle de l'horizon infini suggère que l'entrée dans une interaction longue et dont les acteurs ne connaissent pas l'échéance constitue des conditions favorables à l'émergence de la coopération. D'une manière générale, ce que l'on attend d'un modèle d'interaction répétée est un certain *pouvoir explicatif* vis-à-vis du phénomène qui nous intéresse ; on peut donc se demander si les restrictions computationnelles engendrent un modèle qui nous informent sur la façon dont la coopération peut émerger quand on limite les capacités stratégiques des agents.

En d'autres termes, que signifient les stratégies qui correspondent aux automates que l'on a construits ? Sont-elles purement *ad hoc* ? Cette est question est d'autant plus importante que la motivation qui commmandent les restrictions computationnelles est précisément de rendre les stratégies plus réalistes.

D'abord, le point le plus massif est probablement qu'une stratégie comme les A_j est basée sur la preuve du *Folk Theorem*, c'est-à-dire sur IMPIT : exception faite de la phase de communication, une partie normale est instituée et toute déviation d'un joueur est irrévocablement punie par le minimax. C'est toujours la menace d'une sanction irrévocable qui soutient la coopération.

Maintenant, ce qu'il est intéressant de voir, c'est l'originalité des stratégies computationnellement restreintes par rapport à IMPIT. Clairement, l'envoi de messages pendant la phase de communication (Phase 1) et surtout leur usage dans la période de rappel de la Phase 3 constituent l'originalité des stratégies que l'on a décrites dans les sections précédentes. Malheureusement, il est assez difficile de voir quelle pourrait être l'interprétation que l'on pourrait donner de ces messages. Par rapport à IMPIT, les stratégies computationnellement restreintes représentent donc un surcroît de difficultés interprétatives ; surtout, il semble que l'on arrive au résultat inverse de l'objectif initial qui était de rendre les stratégies utilisables par les agents plus réalistes.

Cependant, si l'on ne prend pas les stratégies à l'équilibre au pied de la lettre et si l'on regarde le modèle avec un peu plus de recul (et d'indulgence), on peut dire que les restrictions computationnelles suggèrent l'interprétation suivante : l'entrée dans une interaction suffisamment longue d'agents qui, tout en connaissant l'échéance de leur interaction, ne sont pas nécessairement capables de se situer relativement à elle, permet l'émergence d'équilibres coopératifs.

Je pense que cette interprétation du modèle est relativement fidèle aux résultats et aux outils que l'on a utilisés pour y parvenir ; elle permet également d'en cerner une limitation importante.

Les automates finis peuvent ne pas savoir où ils en sont dans l'interaction si on les empêche de compter ; mais manifestement, il n'est pas toujours nécessaire d'estimer le temps écoulé pour pouvoir se situer par rapport à une échéance. Il est des cas où ça peut l'être : imaginons un photocopieur dont on veut estimer quand ses cartouches d'encre seront vides ; la meilleure méthode est probablement de compter le nombre de photocopies qu'il fait. Mais il est des cas où ça ne l'est pas : imaginons les relations avec un élu dont on connaît l'échéance du mandat et dont on sait qu'il ne pourra le renouveler. Pourtant, dans ce genre de situations également, il semble bien que des situations de coopération peuvent émerger, bien qu'elles tendent probablement à s'amenuiser lorsque l'échéance approche.

En bref, les restrictions computationnelles miment, dans les jeux à horizon fini, le ressort des jeux à horizon infini, soit le déficit de positionnement temporel relativement à l'échéance ; elles n'apportent donc pas véritablement de pouvoir explicatif supplémentaire, ne rendant notamment pas compte de la façon dont la coopération pourrait se développer entre des agents qui connaissent leur position relativement à l'échéance et que cette connaissance n'accapare pas cognitivement. Il semble donc pour le moins prématuré de considérer que les restrictions computationnelles nous fournissent une "solution" aux paradoxes suggérés par le Dilemme du Prisonnier.

2.6 Références

[A 84] R.Axelrod, The Evolution of Cooperation, Basic Books, 1984 ; trad.fr Comment réussir dans un monde d'égoïstes, O.Jacob, 1996

[AR 88] D.Abreu et A.Rubinstein, "The Structure of Nash Equilibrium in Repeated Games with Finite Automata", *Econometrica*, Vol.56, Nov.1988, pp. 1259-1281

[N 98] A.Neyman, "Finitely Repeated Games with Automata", *Mathematics of Operations Research*, Vol. 23, N°3, 1998, pp.513-552

[PY 94] C.Papadimitriou et M.Yannakakis, "On Bounded Rationality and Computational Complexity", STOC 94

[R 98] A. Rubinstein, Modeling Bounded Rationality, MIT Press, 1998

[S 92] S.Sorin, "Repeated Games with Complete Information", dans R.J.Aumann et S.Hart, Handbook of Game Theory, vol.I, Elsevier 1992

Chapitre 3

Théorie computationnelle du mécanisme

3.1 Introduction

Le problème dit du mécanisme (ou de l'implémentation) peut être considéré comme *l'inverse du problème typique de théorie des jeux* [OR 94] : au lieu de fixer les règles d'un jeu et d'en chercher les solutions possibles (les équilibres), on se donne des issues et on cherche les règles d'un jeu dont les solutions seraient précisément ces issues. On appelle *organisateur* celui qui doit résoudre un tel problème ; l'archétype de l'organisateur est Salomon qui, d'après l'Ancien Testament, invente un "mécanisme" pour arriver à attribuer le nouveau-né disputé à sa vraie mère. On trouve dans cet exemple les ingrédients essentiels de la théorie du mécanisme : un organisateur doit faire face à un problème d'allocation ; pour résoudre ce problème d'allocation, il ne peut s'appuyer que sur certaines informations que les agents peuvent lui donner mais dont rien ne garantit qu'elles soient véraces.

Avant de présenter les définitions formelles nécessaires, on va donner quelques intuitions sur cette perspective un peu particulière et sur la terminologie assez lourde qui lui est associée. On notera en gras les notions centrales qui seront formellement définies par la suite.

On se place du point de vue d'un planificateur qui dispose comme données de départ d'un **arrière-plan**, c'est-à-dire d'un ensemble d'agents, avec leurs goûts possibles, et d'un ensemble de biens qu'il peut leur distribuer. Supposons par exemple que le organisateur dispose d'un seul bien b et considère trois agents, I, II et III. Chaque agent associe une valeur au bien, *valeur qui est connue de lui seul* ; la configuration informationnelle est capitale. Cette valeur que l'agent i associe au bien, on l'appelle le **type** de i.

L'organisateur désire distribuer le bien b en fonction de ces types ; en général, on suppose que l'organisateur désire maximiser les valeurs que les agents accordent aux biens, ici donc allouer b à l'agent qui lui attribue la plus grande valeur. On appelle cet objectif général de l'organisateur la **fonction de choix social**. Pour la mettre en place, il est libre d'instaurer des **paiements** ; par exemple, l'agent qui obtient le bien doit payer un certain prix fixé ou non à l'avance, et les autres en général ont un paiement nul. Le plan que prépare le organisateur va donc se terminer par une allocation du bien et un paiement (positif ou nul) pour chaque joueur ; on peut résumer cela sous la forme d'un quadruplet $o =< i, p_I, p_{II}, p_{III} >$ où i est le joueur qui obtient b et les p_i sont les paiements. On appelle **issue** un tel quadruplet ; une issue est donc un état résultant possible.

Pour représenter la valeur que les agents accordent aux issues, les types ne suffisent pas : l'agent i peut accorder peu de valeur à une issue où il obtient l'objet b (auquel, en soi, il peut accorder une valeur forte) mais où il doit payer beaucoup. Cette valeur globale, qui dépend de l'allocation, des paiements, et du type de l'agent, on l'appelle l'**utilité** de l'agent. Dans notre exemple, pour $o =< i, p_I, p_{II}, p_{III} >$ l'utilité de l'agent i, étant donnés o et son type, est représentée comme la valeur qu'il accorde à b (qu'il obtient donc ici) moins son paiement p_i.

L'hypothèse centrale est que les agents cherchent à maximiser leur utilité ; pour aboutir à sa fonction de choix social, l'organisateur ne doit pas compter sur la bonne volonté des agents, mais seulement sur la poursuite de leur intérêt propre. En particulier, rien ne garantit que, s'il demande aux agents de lui révéler leurs types, les agents le fassent véracement puisque cela n'est peut-être pas de leur intérêt.

Pour poursuivre son objectif, l'organisateur va donc mettre sur pied un jeu où les agents auront certaines actions offertes et où, en fonction de ces actions, le bien b sera alloué et les paiements établis. Par exemple, l'organisateur peut proposer une enchère du genre suivant :

- chaque agent a pour action possible une offre pour le bien b ; cette offre peut correpondre ou non à la valeur réelle que son type lui fait attribuer à b
- b est attribué à celui qui fait la meilleure offre
- celui qui obtient b le paie au prix de la seconde meilleure offre ; les autres ont un paiement nul

Quand on regarde attentivement ces règles, on se rend compte que l'organisateur spécifie deux composantes, et qu'ensuite un jeu est automatiquement induit étant donné l'arrière-plan. D'une part, l'organisateur offre des actions possibles à chaque agents ; d'autre part il relie ces actions à des issues au sens où on l'a défini plus haut. Les utilités des agents sur les issues induisent des utilités sur les actions, et l'on a un jeu au sens technique du terme.

Ces règles du jeu ou **mécanisme**, on ne veut pas qu'elles soient arbitraires ; on veut que, lorsque les agents jouent de manière rationnelle (de manière à maximiser leur utilité), l'issue associée à leurs actions soit précisément celle qui satisfait l'objectif du organisateur (la fonction de choix social).

Par exemple, on veut que les règles de l'enchère décrite précédemment soient bien telles que le joueur qui obtienne b soit celui qui lui accorde la plus grande valeur. Plus précisément, puisque l'on a décidé d'accorder b à la plus forte offre, on veut que celui qui attribue la plus forte valeur à b soit incité, par la poursuite de la maximisation de son utilité, à faire la plus haute offre. Les paiements doivent donc être conçus pour motiver les joueurs à procéder ainsi.

Dans la section 3.2, on expose les notions de base de la théorie classique du mécanisme.

Dans la section 3.3, la théorie "computationnelle" du mécanisme sera abordée par le biais des enchères combinatoires ; cet exemple relève du premier genre de convergence entre théorie des jeux et théorie de la complexité, soit une approche computationnelle de questions traditionnelles de théorie des jeux.

Dans la section 3.4, on donnera un bref exemple du second genre de convergence, soit une approche décisionnelle des réseaux informatiques ; il s'agira en effet de se demander quel système instituer pour motiver des relais à effectuer du routage de données.

3.2 Théorie du mécanisme

3.2.1 Définitions de base

Définition

On appelle **arrière-plan** un triplet $< N, O, (\Theta_i)_{i \in N} >$ où
- O est un ensemble d'issues
- N est un ensemble d'agents
- Θ_i est l'ensemble des types de l'agent i c'est-à-dire l'ensemble de ses déterminants de préférence

Notation
On dénote par $\theta = (\theta_1, ..., \theta_n) \in \Theta = \Theta_1 \times ... \times \Theta_n$ un profil de types

Définition

Etant donné un ensemble de types Θ_i et un ensemble d'issues O, l'**utilité** de l'issue o pour le joueur i s'il est dans le type $\theta_i \in \Theta_i$ est notée $u_i(o, \theta_i)$

Exemple

Soit un unique bien b que l'on cherche à vendre par enchère à deux joueurs - $N = \{1, 2\}$. O est alors l'ensemble des triplets (i, p_1, p_2) où $i \in \{1, 2\}$ est le joueur qui obtient le bien et p_j le prix qui est demandé au joueur j. L'ensemble des type Θ_i détermine les différentes valeurs possibles v_i accordées au bien par le joueur i, et si l'on accepte l'hypothèse selon laquelle les paiments exigés et les valeurs accordées au bien sont homogènes, l'utilité peut se calculer comme la valeur estimée du bien moins le prix qui en est (éventuellement) demandé :
$u_i((j, p_i, p_j), \theta_i) = v_i(j, \theta_i) - p_i$

Définition

Une **fonction de choix social** est une fonction $f : \Theta_1 \times ... \times \Theta_n \longrightarrow O$ qui associe à chaque profil de types une issue

Exemple

la fonction la plus communément retenue est dite fonction efficace (ou "de bien-être social" ou encore fonction utilitariste) ; c'est celle qui maximise la valeur que les agents attribuent aux issues :

$$f(\theta) = \max_{o \in O} \sum_{i=1}^{n} v_i(o, \theta_i)$$

♢

Notation

On appelle contexte la donnée d'un arrière-plan et d'une fonction de choix.

Pour construire un jeu à partir de l'arrière-plan $< N, O, \Theta >$, il faut spécifier d'une part les actions offertes à chaque agent et d'autre part les liens qui existent entre ces actions et les éléments de O, c'est-à-dire la fonction de conséquence

Définition

Un **mécanisme** est un couple $M =< A, g >$ où
– $A = (A_i)_{i \in N}$ est la famille des actions offertes à chaque agent
– $g : A \rightarrow O$ est une fonction de conséquence

Exemple : l'enchère au second prix

Comme on l'a vu précédemment, dans une enchère au second prix d'un seul bien,

- les actions offertes aux agents sont l'annonce d'une offre pour l'objet en jeu
- la fonction de conséquence attribue l'objet à l'agent qui a fait l'offre la plus généreuse, et au prix de la seconde offre la plus généreuse

♢

Remarque : pour un arrière-plan donné, un mécanisme induit un jeu si on étend la fonction d'utilité des joueurs conformément à g : pour $a \in A$, $u_i(a, \theta_i) = u_i(g(a), \theta_i)$

Définitions

Soient S un concept de solution et un contexte $C =< AP, f >$ (composé d'un arrière-plan AP et d'une fonction de choix social f) ; on dit que

– si M est un mécanisme, M **S-implémente** la fonction f si, pour tout $\theta \in \Theta$, $f(\theta) = g(a_1{}^*, ..., a_n{}^*)$ où $(a_1{}^*, ..., a_n{}^*)$ est un S-équilibre dans le jeu induit par M.

– un **problème de S-implémentation** à trouver un couple M qui permette de S-implémenter f.

Notation Si S est un équilibre de Nash, on parle de N-implémentation ; si c'est un équilibre de stratégies dominantes de DS-implémentation ; enfin si c'est un équilibre bayésien, on parle de BN-implémentation.

3.2.2 Différentes notions d'équilibre

On a dit qu'un problème de mécanisme consistait à chercher un jeu dont les équilibre correspondraient à des issues désirables *a priori* ; il existe cependant de nombreuses notions d'équilibre. Dans un problème de mécanisme, l'équilibre est donc un paramètre à spécifier ; dans cette section, on passe en revue les trois principaux équilibres retenus en théorie du mécanisme, même si, dans la suite, on se concentrera surtout sur l'équilibre en stratégies dominantes.

Pour définir plus précisément les différents concepts d'équilibre, on va parler de *stratégies* plutôt que d'actions. On note S_i l'ensemble des stratégies $s_i : \Theta_i \rightarrow A_i$ de l'agent i ; les stratégies sont donc paramétrées par les types.

Exemple([MWG 95], p.254.)

L'exemple suivant montre comment le paramétrage des actions par les types enrichit l'espaces des stratégies possibles. On considère un Dilemme du Prisonnier où le joueur 2 a deux types possibles, le type I et le type II ; il a alors 4 stratégies au sens où l'on vient de définit la notion :
– (avouer si type I, trahir si type II)
– (avouer si type II, trahir si type I)
– (avouer dans tous les cas)
– (trahir dans tous les cas) $\diamondsuit$

Définition

Supposons que les agents aient le profil de types $\theta = (\theta_1, ..., \theta_n)$

– un profil de stratégies $s^* = (s_1^*, ..., s_n^*)$ est un **équilibre de Nash** si pour tout $i \in N$ et pour tout $s'_i \neq s_i^*, s'_i \in S_i$
$u_i(g(s_i^*(\theta_i), s_{-i}^*(\theta_{-i})), \theta_i) \geq u_i(g(s'_i(\theta_i), s_{-i}^*(\theta_{-i})), \theta_i)$

– un profil de stratégies $s^* = (s_1^*, ..., s_n^*)$ est un **équilibre de stratégies dominantes** si pour tout $i \in N$ et pour tout $s'_i \neq s_i^*, s'_i \in S_i$ et pour tout $s_{-i} \in S_{-i}$,
$u_i(g(s_i^*(\theta_i), s_{-i}(\theta_{-i})), \theta_i) \geq u_i(g(s'_i(\theta_i), s_{-i}(\theta_{-i})), \theta_i)$

Commentaire : nous avons déjà abordé au Chapitre I la notion d'équilibre de Nash. Une stratégie dominante est une meilleure réponse non seulement au profil de stratégies adverses à l'équilibre, mais à *tous* les profils de stratégies des adversaires. C'est donc un concept de solution bien plus fort que l'équilibre de Nash.

Exemple : dans le Dilemme du Prisonnier, le couple de stratégies (D, D) est un équilibre de stratégies dominantes.

Une première conséquence de la force de la notion d'équilibre en stratégies dominantes, c'est qu'à la différence de l'équilibre de Nash, il n'existe pas toujours d'équilibre dans un jeu donné - même quand on s'autorise des stratégies mixtes.

Une seconde conséquence, plus heureuse, c'est que l'équilibre en stratégies dominantes est aussi un concept plus robuste : si un tel équilibre existe, il y a fort à parier qu'il sera joué car les agents n'ont pas à anticiper les stratégies des adversaires à l'équilibre. En effet, une stratégie dominante domine toutes les stratégies possibles des adversaires. En particulier, l'agent n'a pas à connaître les préférences de leurs adversaires, ni même à postuler leur rationalité.

Définitions

Soient un ensemble N de joueurs et une famille de types $(\Theta_i)_{i \in N}$ pour ces joueurs ;

• on définit l'***a priori* commun** ("*common prior*") comme une distribution de probabilités sur l'ensemble des profils (de types) possibles, ie un élément de $\Delta(\Theta = \times_i \Theta_i)$

• supposons que les joueurs aient pour profil de types θ ; un profil de stratégies $s^*(.) = (s_1^*(.), ..., s_n^*(.))$ est un **équilibre bayésien** si pour tout $i \in N$, pour tout $\theta_i \in \Theta_i$ et pour tout $s_i' \neq s_i^*, s_i' \in S_i$

$$E_{\theta-i}[u_i(g(s_i^*(\theta_i), s_{-i}^*(\theta_{-i})), \theta_i)/\theta_i] \geq E_{\theta-i}[u_i(g(s_i'(\theta_i), s_{-i}^*(\theta_{-i})), \theta_i)/\theta_i]$$

où $E_{\theta-i}$ est l'espérance déterminée par l'*a priori* commun.

Commentaire : dans l'équilibre bayésien, chaque joueur ne joue pas une meilleure réponse au profil des adversaires mais une meilleure réponse à la distribution de probabilités sur les profils des adversaires.

3.2.3 Mécanismes à révélation directe et principes de révélation

Mécanismes à révélation directe

La classe la plus importante des mécanismes est celle où les actions qui sont offertes aux agents est l'envoi de messages concernant leur type. Le principe de révélation assure qu'en se restreignant à ce genre de mécanisme, on ne perd pas en généralité.

Définition

M est un **mécanisme à révélation directe** si pour tout joueur i, $A_i = \Theta_i$. On note $\hat{\theta}_i$ le type révélé par le joueur i.

Définition

Soit M un mécanisme, Θ un ensemble de profils de préférence, S un concept d'équilibre et f une fonction de choix social ; on dit que M **S-implémente véracement** (ou est compatible avec les motivations) la fonction f si M est un mécanisme de révélation directe et si dire la vérité est un S-équilibre dans le jeu induit par M.

Commentaire : un mécanisme à révélation directe et vérace est donc un mécanisme où l'intérêt de chaque agent l'incite à rapporter ses préférences réelles. Il existe de nombreuses situations où les agents n'ont pas intérêt à être véraces.

Exemple 1, mécanisme non vérace : [MWG 95], p. 860.

Supposons qu'il y ait trois issues x, y, z que l'agent 1 a un seul type possible, θ_1 qui classe les issues comme suit : $x \succ y \succ z$. L'agent 2 pour sa part a deux types possibles : θ_2 dont le classement est $z \succ y \succ x$ et θ_2' dont le classement est $y \succ x \succ z$. Si la fonction de conséquence est $g(\theta_1, \theta_2) = y$ et $g(\theta_1, \theta_2') = x$, clairement quand le vrai type de l'agent 2 est θ_2', il a intérêt à mentir et à annoncer θ_2. $\diamondsuit$

Exemple 2, mécanisme vérace : enchères de Vickrey.

Remarque : il découle immédiatement des définitions que dans un mécanisme vérace, la fonction de conséquence g est identique à la fonction de choix social f.

Les principes de révélation énoncent qu'à certaines conditions si une fonction de choix est implémentable par un mécanisme dans un certain arrière-plan, alors elle l'est également par un mécanisme à révélation directe et vérace.

Principe de révélation

Théorème (Principe de Révélation en équilibre de stratégies dominantes)

Soit un contexte $C =< AP, f >$; s'il existe un mécanisme M qui DS-implémente C, alors il existe un mécanisme M' à révélation directe et vérace qui DS-implémente également C.

Preuve ([MGW 95], p. 871)

Soit un mécanisme $M =< (S_i)_{i \in N}, g >$ tel qu'il existe un profil de stratégies s^* qui soit un DS-équilibre et tel que pour tout θ, $g(s^*(\theta)) = f(\theta)$. On a donc par définition pour tout $i \in N$ et pour tout $s'_i \neq s^*_i, s'_i \in S_i$ et pour tout $s_{-i} \in S_{-i}$,

$$u_i(g(s^*_i(\theta_i), s_{-i}(\theta_{-i})), \theta_i) \geq u_i(g(s'_i(\theta_i), s_{-i}(\theta_{-i})), \theta_i)$$

donc en particulier

$$u_i(g(s^*_i(\theta_i), s^*_{-i}(\theta_{-i})), \theta_i) \geq u_i(g(s^*_i(\theta'_i), s^*_{-i}(\theta_{-i})), \theta_i)$$

pour tout $i \in N$, $\theta_i, \theta'_i \in \Theta_i$. Or $g(s^*(\theta)) = f(\theta)$ donc

$$u_i(f(\theta_i, \theta_{-i}), \theta_i) \geq u_i(f(\theta'_i, \theta_{-i}), \theta_i)$$

Cela signifie donc que dans le jeu induit par un mécanisme à révélation directe et où la fonction de conséquence g est identique à la fonction de choix f, révéler ses véritables préférences est un équilibre en stratégies dominantes. ♠

Hypothèse de quasi-linéarité

Une hypothèse courante porte sur la structure des utilités des agents :

Définition

Supposons qu'à toute issue $o \in O$ on puisse associer le choix d'une allocation d d'un ensemble d'allocations possibles D et une famille de paiements $(p_i)_{i \in N}$; la fonction d'utilité de l'agent i est **quasi-linéaire** si pour tout type θ_i,

$$u_i(o, \theta_i) = v_i(d, \theta_i) - p_i$$

Sous cette hypothèse, on peut décomposer les fonctions de choix social f et de conséquence g en :

$\underline{f(\theta) = (x(\theta), p_1(\theta), ..., p_n(\theta))}$ où

- $x()$ est une fonction de choix
- les $p_i()$ sont des fonctions de paiement

$\underline{g(s) = (k(s), t_1(s), ..., t_n(s))}$ où

- $k : S_1 \times ... \times S_n \to D$ est une fonction d'allocation (ainsi appelée parce que souvent D est un ensemble d'allocations possibles)

– $t_i : S_1 \times ... \times S_n \rightarrow \mathbf{R}$ est fonction de transfert qui détermine le paiement de l'agent i pour un profil de types révélés donné

Exemple : un projet public à coût uniformément réparti

On demande à un ensemble N d'agents s'il sont pour ($d = 1$) ou contre ($d = 0$) la construction d'un ouvrage d'arts ; chaque t_i représente le financement demandé aux agents pour un tel projet. Si l'on suppose que ces fonctions répartissent également le coût $c()$ sur l'ensemble des agents, soit $t_i = (c(d)/N)$, alors l'utilité de chaque agent est donc
$u_i(d, \theta_i) = v_i(d, \theta_i) - (c(d)/N)$
([MGWG 95], p.861) $\diamondsuit$

Il y a de très nombreuses propriétés que l'on peut définir sur la fonction de choix sociale $f()$ ainsi décomposée : on peut exiger que les paiement soient équilibrés, faiblement (leur somme est non négative) ou fortement (leur somme est nulle), que la fonction de choix $x()$ maximise les valeurs des agents, que chaque agent ait intérêt à s'engager dans le jeu (utilité non négative), etc...

La propriété que l'on commence par imposer concerne souvent l'objectif général de l'organisateur ; les deux objectifs les plus courants sont inspirés d'une part des enchères de biens privés où l'on peut faire l'hypothèse que le vendeur cherche à maximiser son revenu (donc la somme des fonctions de transfert), d'autre part par les enchères de biens public où l'on fait souvent l'hypothèse que le vendeur (l'Etat) cherche à maximiser les valeurs que les acheteurs accordent aux biens. Dans le premier cas, on dit que la fonction de choix est optimale ; dans le second, qu'elle est efficace. On se restreindra ici aux fonctions de choix efficaces.

Définition

une fonction de choix social $f(\theta)$ est **efficace** ("allocatively-efficient") si pour tout profil de préférences θ et pour toute autre fonction de choix $x'() \neq x()$,
$\sum_{i=1}^{N} v_i(x(\theta), \theta_i) \geq \sum_{i=1}^{N} v_i(x'(\theta), \theta_i)$

A l'occasion, on pourra également mobiliser l'exigence que l'organisateur n'ait pas un budget déficitaire :

Définitions

• une fonction de choix social $f(\theta)$ est **budgétairement équilibré** si pour tout profil de préférences θ

$\sum_{i \in N} p_i(\theta) \geq 0$

• une fonction de choix social $f(\theta)$ est **fortement budgétairement équilibré** si pour tout profil de préférences θ

$\sum_{i \in N} p_i(\theta) = 0$

3.2.4 Les mécanismes VCG

Un des résultats principaux de la théorie du mécanisme est que l'on peut ramener à une seule famille de mécanismes les mécanismes à révélation directe qui sont DS-véraces et efficaces.

Définition (Groves 1973)

Un **mécanisme VCG** est un mécanisme à révélation directe où pour un n-uplet $\hat{\theta} = (\hat{\theta}_1, ..., \hat{\theta}_n)$ de types rapportés ($\hat{\theta}_i = s(\theta_i)$),

- $k^*(\hat{\theta}) = \max_{d \in D} \sum_{i=1}^{n} v_i(d, \hat{\theta}_i)$
- $t_i^*(\hat{\theta}) = h_i(\hat{\theta}_{-i}) - \sum_{j \neq i} v_j(k^*(\hat{\theta}), \hat{\theta}_j)$

où $h_i : \hat{\Theta}_{-i} \to \mathbf{R}$ est une fonction arbitraire qui prend pour arguments les types rapportés par les adversaires du joueur i.

Proposition

Un mécanisme VCG est DS-vérace

Preuve

Soit M un mécanisme VCG qui ne soit pas DS-vérace ; alors il existe un joueur i et un type $\hat{\theta}_i \neq \theta_i, \hat{\theta}_i \in \Theta_i$ tels que

$$v_i(k^*(\hat{\theta}_i, \theta_{-i}), \theta_i) - t_i(\hat{\theta}_i, \theta_{-i}) > v_i(k^*(\theta_i, \theta_{-i}), \theta_i) - t_i(\theta_i, \theta_{-i})$$

$$v_i(k^*(\hat{\theta}_i, \theta_{-i}), \theta_i) + \sum_{j \neq i} v_j(k^*(\hat{\theta}_i, \theta_{-i}), \theta_j) > v_i(k^*(\theta_i, \theta_{-i}), \theta_i) + \sum_{j \neq i} v_j(k^*(\theta_i, \theta_{-i}), \theta_j)$$

$$\sum_{j \in N} v_j(k^*(\hat{\theta}_i, \theta_{-i}), \theta_j) > \sum_{j \in N} v_j(k^*(\theta_i, \theta_{-i}), \theta_j)$$

Contradiction. ♠

Commentaire :

- dans un mécanisme de Grove, le transfert à destination de l'agent i ne dépend de i que dans la mesure où le type qu'il annonce participe à la détermination de $k^*(\theta)$; en outre, un changement de type annoncé n'influence le transfert que dans la mesure où il induit un choix qui est différemment évalué par les adversaires (et seulement eux) de i. La fonction de transfert a donc pour effet d'internaliser les externalités de chaque agent ie l'effet induit sur les autres de ses reports de type ; le paiement de i va ainsi refléter le système des paiements tout entier, ce qui incite i à être vérace.

- les fonctions $h_i : \hat{\Theta}_{-i} \to \mathbf{R}$ n'interviennent pas dans la preuve ; elles sont introduites car elles permettent aux mécanismes d'avoir éventuellement des propriétés supplémentaires ("budget-balanced", "individual rationality"...)

Exemple : les enchères (simples) de Vickrey

Chaque joueur i soumet une offre b_i ; celui qui propose l'offre la plus élevée remporte l'enchère et obtient le bien en jeu au prix de la seconde meilleure offre. Formellement,

$f(\theta) = (k(b) =< k_1(b), k_2(b) >, t_1(b), t_2(b))$ où
- $k_1(b) = 1$ si $b_1 \geq b_2$; $= 0$ sinon
- $k_2(b) = 1$ si $b_1 < b_2$; $= 0$ sinon
- $t_1(b) = b_2.k_1(b)$
- $t_2(b) = b_1.k_2(b)$

Les enchères simples de Vickrey sont bien des mécanismes de Grove puisque l'on peut exprimer les fonctions de transfert de la manière suivante :

$$t_i(b) = b_j - v_j(k(b), b_j)$$

On peut également vérifier que être vérace est une stratégie dominante dans les enchères de Vickrey : soit b_i une offre de l'agent i et supposons que $b_i \neq \theta_i$ où θ_i est le vrai type du joueur i.

- ou bien annoncer b_i ne change pas l'identité du gagnant, et alors il le peut affecter non plus le paiement de i.

- ou bien i emporte l'enchère avec b_i alors qu'il ne l'aurait pas emporté en étant vérace ; alors $b_i > \theta_i$ et il existe un autre joueur j tel que $b_i > b_j > \theta_i$; mais dans ce cas, le paiement de i va induire une utilité négative, alors qu'elle aurait été nulle si il avait été vérace

- ou bien i perd l'enchère avec b_i alors qu'il l'aurait gagné sinon et donc $b_i < \theta_i$; il ne peut alors faire mieux qu'avec θ_i qui lui rapporte forcément une utilité non négative. $\diamondsuit$

3.3 Enchères combinatoires

3.3.1 Définitions et exemples

Les enchères combinatoires sont des mécanismes où les agents ont la possibilité de lançer des offres non plus simplement sur des objets isolés, mais sur des ensembles d'objets provenant d'une base finie ; elles permettent une expression plus raffinée des préférences des agents et rendent possible une allocation plus efficace des biens.

Définition

Un **arrière-plan d'allocation combinatoire** est un triplet $< N, I, (v_i)_{i \in N} >$ où
- N est un ensemble de joueurs
- I est un ensemble d'items discrets
- le type d'un agent i se ramène à une fonction de valuation $v_i : 2^I \to \mathbf{R}$ qui attribue une valeur à chaque sous-ensemble d'items

Hypothèses

- *normalisation* : $v_i(\emptyset) = 0$
- *monotonie* ("*free disposal*") : pour tout $s, s' \in I$, si $s \subseteq s'$, $v_i(s) \leq v_i(s')$

Définition

Soit un arrière-plan d'allocation combinatoire $< N, I, (v_i)_{i \in N} >$; le **problème d'allocation combinatoire (PAC)** associé est le problème de trouver une allocation des items de I qui soit efficace, c'est-à-dire une allocation $S = (S_1, ..., S_n)$ telle que tout $S_i \subseteq I$ et

– $S = \max_{(S_1,...,S_n)} \sum_{i \in N} v_i S_i$

– pour tout $i \neq j$, $S_i \bigcap S_j = \emptyset$

Les enchères combinatoires ne sont pas simplement un casse-tête algorithmique ; il existe de nombreuses applications potentielles du (PAC) comme par exemple
- les allocations de plages de cours à des étudiants (contraintes typiques : les cours ne se chevauchent pas, le nombre d'U.V est correct, un T.D. pour chaque C.M, etc...)
- les allocations de plages de décollage et d'atterrisage (contraintes typiques : avoir un atterrissage pour tout décollage, etc...)
- les allocations de "*travel packages* " (chambres d'hôtel, billets d'avion, locations de véhicules... ; contraintes typiques : distance hôtel/aéroport, temps de location du véhicule inférieur ou égal à l'intervalle aller/retour,...)
- aux Etats-Unis, l'attribution d'ondes radio aux compagnies de téléphonie mobile par la Commission Fédérale des Communications (FCC)

Si les applications potentielles des enchères combinatoires sont nombreuses, les raisons d'y recourir également : l'intérêt de ces enchères, c'est de pouvoir (partiellement au moins) capturer une structure de préférence complexe des agents. En effet,
- un agent peut être prêt à payer plus pour deux objets s'il les obtient tous les deux que la somme de ce qu'il serait prêt à payer pour chacun pris isolément ; on dit qu'il y a *complémentarité* entre ces biens.
- inversement, un agent peut être disposé à payer moins pour deux objets obtenus conjointement que la somme de ce qu'il serait prêt à payer pour chacun. Pensons par exemple à deux voyages qui se déroulent au même moment ! On dit alors qu'il y a *substituabilité* entre ces biens.

3.3.2 Enchères généralisées de Vickrey et mécanismes de Clarke

On appelle enchère de Vickrey généralisée l'application des mécanismes VCG au (PAC) ; sous hypothèse de quasi-linéarité, il nous faut un ensemble des allocations D. Il est naturellement défini à partir de I :

$$D = \{(S_1, ..., S_n) : S_i \bigcap S_j = \emptyset \ , \ S_i \subseteq I\}$$

Plus précisément, les enchères de Vickrey généralisées relèvent d'une certaine famille de mécanismes VCG, les mécanismes de Clarke, qui spécifient d'une manière bien particulière la fontion $h_i(\hat{\theta}_{-i})$:

Définition

On appelle **mécanisme de Clarke** un mécanisme VCG où

$$h_i(\hat{\theta}_{-i}) = \sum_{j \neq i} v_j(k^*_{-i}(\hat{\theta}_{-i}), \hat{\theta}_j)$$

avec

$$k^*_{-i}(\hat{\theta}_{-i}) = \max_{d \in D} \sum_{j \neq i} v_j(d, \hat{\theta}_j)$$

Commentaires :

- la fonction $k^*_{-i}(\hat{\theta}_{-i})$ sélectionne donc l'allocation qui maximise la valeur totale que les agents lui attribuent sur la base des types révélés, abstraction faite de l'agent i. C'est donc une fonction d'allocation efficace pour $N \backslash I$. La fonction $h_i(\hat{\theta}_{-i})$ est précisément la somme des valeurs révélées pour une telle allocation maximale.

- dans un mécanisme de Clarke, la fonction de transfert est donc

$$t_i(\hat{\theta}) = \sum_{j \neq i} v_j(k^*_{-i}(\hat{\theta}_{-i}), \hat{\theta}_j) - \sum_{j \neq i} v_j(k^*(\hat{\theta}), \hat{\theta}_j)$$

ce qui signifie que

1) si l'annonce d'un type par le joueur i ne modifie pas l'issue qui maximise les types révélés par rapport à celle qui maximise uniquement les types révélés par ses adversaires, alors le transfert est nul.

2) si au contraire l'annonce modifie l'issue qui maximise les types révélés, alors le prix que le joueur i doit payer équivaut au changement qu'il induit dans la somme totale des types.

3) dans le cas particulier où il y a un seul objet, le joueur i^* qui remporte l'objet est celui qui propose l'offre la plus élevée, et il l'emporte au prix $t_{i^*}(\hat{\theta})$ soit le montant de la deuxième meilleure offre.

En d'autres termes, le transfert caractéristique des mécanismes de Clarke fait payer à chaque agent l'influence que son type révélé a sur la satisfaction globale de ses adversaires.

Exemple : l'enchère de Vickrey comme mécanisme de Clarke

On vérifie facilement que la fonction $h_i(\hat{\theta}_{-i})$ de l'enchère (simple) de Vickrey est de ce type puisque par exemple pour $i = 1$ et $j = 2$,

$k^*_{-1}(b_2) = (0, 1)$

donc

$h_1(\hat{\theta}_{-1}) = v_2(k^*_{-1}(b_2), b_2)$

$h_1(b_2) = v_2((0, 1), b_2)$

$h_1(b_2) = b_2$ $\diamondsuit$

On peut maintenant déterminer les enchères généralisées de Vickrey : ce sont de pures applications des mécanismes de Clarke au (PAC), c'est-à-dire que chaque joueur i annonce une fonction de valuation $\hat{v}_i$ sur l'ensemble des parties de I. A cette condition, les enchères généralisées de Vickrey permettent de DS-véracement implémenter n'importe quel (PAC).

3.3.3 Problèmes de complexité dans les enchères généralisées de Vickrey

Les enchères de Vickrey fournissent donc une solution tout à fait générale qui permet de prendre en compte d'une manière particulièrement raffinée les besoins des agents. Pourtant, il y a bien peu d'exemples concrets de pratiques qui s'accordent avec une telle solution. Une raison essentielle réside dans le fait que *les enchères généralisées de Vickrey ne sont pas faisables du point de vue computationnel* :

"*while combinatorial markets have major economic advantages, they can be computationaly complex to clear. There has been a recent surge of interest in developing combinatorial clearing algorithms*" ([SSG 01])

La présente section est consacrée à ce problème.

Pour l'organisateur, les tâches computationnelles se concentrent manifestement aux niveaux

(1) de la fonction de choix $k^*()$ qui doit proposer une allocation $S^* = (S^*_1, ..., S^*_n)$ qui maximise les valuations rapportées ; ce problème est souvent nommé problème de la détermination du gagnant (PDG), par référence à l'enchère où un seul objet est à vendre.

(2) des fonctions de transfert t_i, donc des fonctions h_i, donc des fonctions $k^*_{-i}(\hat{\theta}_{-i})$ qui sont à nouveau des problèmes de maximisation de valuations rapportées

Remarque : il est important de noter que le (PAC) et le (PDG) sont distincts : dans le (PDG) il s'agit de maximiser les valuations annoncées, dans le (PAC) les valuations réelles. C'est bien sûr la manière dont sont définies les transferts qui va permettre à la résolution du (PDG) d'impliquer la résolution du (PAC).

Le fait est que les problèmes (1) et (donc) (2) sont NP-hard ; est-il possible de trouver un mécanisme qui surmonte cet obstacle ?

Le problème de la détermination du gagnant (PDG)

Définition

Etant donnés un ensemble d'items I et un profil de valuations révélées $\hat{v} = (\hat{v}_1, ..., \hat{v}_n)$, le **problème de la détermination du gagnant (PDG)**consiste à trouver une allocation S^* telle que

$S^* = \max_{d \in D} \sum_{i \in N} \hat{v}_i(d_i)$

où $D = \{(S_1, ..., S_n) : S_i \bigcap S_j = \emptyset \ , \ S_i \subseteq I\}$.

Une telle combinaison S^* est nommée une **allocation gagnante**.

Analyse algorithmique du (PDG)

Le (PDG) n'est manifestement pas un problème aisé à résoudre dès que le nombre d'items en jeu n'est pas très petit. La façon naïve de le résoudre consisterait à énumérer les partitions exhaustives (les partitions qui incluent tous les biens) de I et à comparer leurs performances. Mais le nombre de telles partitions est déjà

$\sum_{q=1}^{|I|} Z(|I|, q)$ où

$Z(n, q) = q.Z(n-1, q) + Z(n-1, q-1)$ et $Z(n, n) = Z(n, 1) = 1$.

Proposition ([S 02])

Le nombre de partitions exhaustives est $\mathcal{O}((|I|)^{|I|})$ et $\omega(|I|^{|I|/2})$.

Comme le remarquent [RPH 95] et [S 02], on peut ramener un (PDG) à un problème du type Zero-One Linear Programming (ZOLP), qui est un problème de programmation entière où, dans les solutions faisables, les variables ne prennent pour valeur que 0 ou 1 (pour les définitions concernant la programmation linéaire, cf Chap.I).

On note tout d'abord $m(S)$ pour $S \subseteq I$ la valuation révélée la plus forte à l'égard de l'ensemble de bien(s) S. Pour simplifier, commençons par substituer alors au problème (PDG) celui de trouver :

$S'' = \max_{p \in P} \sum_{S \in p} m(S)$

où P est l'ensemble des partitions quelconques que l'on peut faire à partir de I soit

$P = \{(S_1, ..., S_q) : q \in \mathbf{N}, S_i \bigcap S_j = \emptyset \ , \ S_i \subseteq I\}.$

On peut facilement exprimer S" sous la forme d'un problème de programmation entière 0-1, que l'on nomme (ZOLP1) :

(ZOLP1)

$\max \sum_{S \in \wp(I)} m(S).x_S$

sujet à

pour tout $S \in \wp(I)$, $x_S \in \{0, 1\}$

pour tout $i \in I$, $\sum_{\{S : i \in S\}} x_S \leq 1.$

(ZOLP1) est lui-même équivalent à un problème bien connu, Weigthed SET PACKING (WSP) :

Weighted SET PACKING (WSP)

données :
une famille (C_i) de sous-ensembles d'un ensemble U ; chaque C_i est affecté d'un poids w_i

problème :
trouver une famille $(C'_j) \subseteq (C_i)$ telle que
- $\sum_{j \in J} w_j$ soit maximum et
- les C'_j soient des ensembles disjoints

Il est clair que (ZOLP1) est équivalent à la recherche de S'' ; mais pour arriver à S'', nous avons simplifié notre problème initial en suppposant que pour chaque sous-ensemble de biens, on pouvait se concentrer sur l'offre la plus généreuse. Il est temps de voir si cette simplification nous fait perdre en généralité.

Exemple

Soient $I = \{1, 2\}$, $N = \{Pierre, Jean\}$ et les valuations suivantes :

	$\{1\}$	$\{2\}$	$\{1, 2\}$
v_{Pierre}	4	6	8
v_{Jean}	3	4	8

On peut alors vérifier que S'' alloue $\{1\}$ et $\{2\}$ à Pierre "séparément" alors que Pierre n'est pas prêt à payer autant pour $\{1, 2\}$! Maintenant, si l'on refuse qu'un même joueur reçoive différentes allocations, ce qui paraît légitime, alors S' alloue $\{1, 2\}$ à Pierre, ce qui élève la somme des valeurs rapportées à 8 ; mais alors on ne maximise pas cette somme car l'allocation qui donne $\{1\}$ à Jean et $\{2\}$ à Pierre a pour somme 9. Donc ici $S'' \neq S^*$. $\diamondsuit$

Le problème que soulève ce genre d'exemples, c'est que les valeurs rapportées manifestent de la substituabilité entre des biens ; c'est ceci qui rend la simplification S'' infidèle au problème original S^*.

[dVV 00] notent que l'on peut formuler le (PDG) dans le cas général en termes de programmation entière en renforçant les contraintes. On va enrichir les variables en les paramétrant sur les joueurs : la variable $x_{(S,i)}$ prend la valeur 1 si le sous-ensemble $S \subseteq I$ est alloué au joueur i, 0 sinon.

(ZOLP2)

$\max \sum_{i \in N} \sum_{S \in \wp(I)} \hat{v}_i(S).x_{(S,i)}$

sujet à

pour tout $S \in \wp(I)$, $i \in N$, $x_{(S,i)} \in \{0, 1\}$

pour tout $j \in I$, $\sum_{\{S: j \in S\}} \sum_{i \in N} x_{(S,i)} \leq 1$

pour tout $i \in N$, $\sum_{S \in \wp(I)} x_{(S,i)} \leq 1$

Cette difficulté peut également être ingénieusement contournée par l'introduction de *pseudo-biens* qui permettent de ramener fidèlement tout (PDG) à (ZOLP1) - donc également à (WSP).

Exemple

Si l'on reprend notre exemple, $v_{Pierre}\{1\} = 4$, $v_{Pierre}\{2\} = 6$ et $v_{Pierre}\{1, 2\} = 8$. Pour une telle situation, introduisons un pseudo-bien p qui sera ajouté au biens substituables mais pas à leur réunion : $v_{Pierre}\{1, p\} = 4$, $v_{Pierre}\{2, p\} = 6$ et $v_{Pierre}\{1, 2\} = 8$. On ne pourra donc plus allouer $\{1, p\}$ et $\{2, p\}$ à Pierre et l'on aboutira effectivement à $S^* = S''$. Regardons alors comment le problème peut s'exprimer en un (WSP).

L'ensemble à considérer est l'ensemble $I \bigcup \{p\} = \{1, 2, p\}$; on pondère alors chaque sous-ensemble S par $m(S)$, soit

$m(\{1\}) = 3$

$m(\{2\}) = 4$

$m(\{1,2\}) = 8$
$m(\{1,p\}) = 4$
$m(\{2,p\}) = 6$
$m(\{1,2,p\}) = 8$
et alors clairement les sous-ensembles choisis vont être $\{1\}$ et $\{2,p\}$. $\diamondsuit$

Maintenant, on peut utiliser des résultats classiques de théorie de la complexité qui concernent (WSP).

Proposition (Karp, 1972)

(WSP) est NP-hard

Preuve

Supposons que l'on ait une instance de Weighted Maximum Independent Set (WMAXIS) soit un graphe pondéré $(V, E, (w_v)_{v \in V})$; procédons alors à la conversion suivante :

- à chaque arête e on fait correspondre un élément c_e ; soit $C = \{c_e; e \in E\}$

- à chaque noeud v, on fait correspondre un sous-ensemble C_v qui contient tous les c_e tels que e est incident à v ; on reporte les poids correspondant w_v aux sous-ensembles C_v.

On obtient donc une instance $(C, (C_v)_{v \in V}, w_v)$ de (WSP) ; il est alors clair que $opt(WMAXIS, (V, E, (w_v)_{v \in V}) = opt(WIS, (C, (C_v)_{v \in V}, w_v))$. Or (WMAXIS) est NP-hard.
$\spadesuit$

Cas faisables du (PDG)

[RHP 95], [T 00], [N 00]

Il existe des restrictions sur le (PDG) qui sont calculables en temps polynomial ([RHP 95], [dVV 00], [LLN 02]) ; la principale voie d'accès à ces restrictions repose sur le fait que l'algorithme dit de l'ellipsoïde résoud en temps polynomial les problèmes de programmation linéaire (non entière). Dès lors, les cas favorables seront ceux où l'on *relâche* le problème de programmation entière en un problème de programmation linéaire mais dont les solutions sont cependant entières. La relaxation linéaire de (ZOLP2) est le problème suivant :

(RZOLP2)

max $\sum_{i \in N} \sum_{S \in \wp(I)} \hat{v}_i(S).x_{(S,i)}$

sujet à

pour tout $S \in \wp(I)$, $i \in N$, $x_{(S,i)} \geq 0$

pour tout $j \in I$, $\sum_{\{S:j \in S\}} \sum_{i \in N} x_{(S,i)} \leq 1$

pour tout $i \in N$, $\sum_{S \in \wp(I)} x_{(S,i)} \leq 1$

Bien sûr, si dans une certaine enchère cela a un sens de distribuer des fractions des biens en jeu, alors on peut très bien se contenter de résoudre (RZOLP2). Mais dans le cas général, on suppose évidemment l'indivisibilité de biens. Le problème dual de (RZOLP2), nommé (DRZOLP2), se construit en introduisant une variable par contrainte linéaire de (RZOLP2) soit $|I| + |N|$ variables et en appliquant la définition donnée au Chapitre I. On note ces variables respectivement p_j pour $j \in I$ et z_i pour $i \in N$

(DRZOLP2)

min $\sum_{j \in I} p_j + \sum_{i \in N} z_i$

sujet à

pour tout $j \in I$, $i \in N$, $p_j, z_i \geq 0$

pour tout $i \in N$, $S \subseteq I$, $z_i + \sum_{j \in S} p_j \geq \hat{v}_i(S)$

On peut alors rechercher systématiquement les endroits où, pour ce problème, la relaxation linéaire produit des solutions entières. La difficulté, néanmoins, de procéder de cette façon "purement mathématique", c'est que la pertinence des cas favorables est passablement aléatoire. Une approche plus fructueuse consiste à imposer des restrictions sur les valuations révélées.

Exemple 1 : valuations quantitativement restreintes et additivement fondées sur les objets

Voici une restriction relativement simple sur les valuations qui nous permet de nous ramener à un problème combinatoire calculable en temps polynomial.

Définition

Un profil de valuations $\hat{v}$ est **restreint quantitativement et additivement fondé sur les objets** si pour tout joueur i la valuation v_i est de la forme $v_i = ((a_1, p_1), ..., (a_k, p_k), q_i)$ ce qui signifie que i est prêt à payer p_j pour les objets $a_j \in I$ et la somme des p_j pour un ensemble d'objets a_j pourvu que les a_j soient en nombre inférieur ou égal à q_i.

Théorème ([T 00])

Un (PDG) où le profil de valuations est restreint quantitativement et additivement fondé sur les objets est calculable en temps polynomial.

Preuve

On procède par réduction au problème suivant :

Weighted b-MATCHING (WbM)

données :
un graphe $G = (V, E)$
un coût w_e pour chaque arête e
une suite $b = ((b_1, l_1), ..., (b_{|V|}, l_{|V|})$ avec $b_i \in \mathbf{N}$ et $l_i = 0$ ou b_i

problème :
trouver un ensemble d'arêtes $M \subseteq E$ tel que
- pour tout noeud $i \in V$, le cardinal des arêtes de M incidentes à i est supérieur à l_i et inférieur à b_i
- $M = \max_{S \subseteq E} \sum_{e \in S} w_e$

En effet, on construit une graphe G où
$V = I \bigcup (v_i)_{i \in N}$
$E = I \times (v_i)_{i \in N}$
pour $i \in N$ et $j \in I$, $w_{e_{ij}} = v_i(j)$
pour un noeud v_i, la paire correspondante dans b est $(0, q_i)$
pour un noeud $j \in I$, la paire correspondante dans b est $(0, 1)$
♠

Exemple 2 : valuations ayant la propriété de gross substitutes

Un exemple plus classique et beaucoup plus sophistiqué est celui, détaillé dans [LLN 02], de la propriété de *gross substitutes*. L'intérêt de cette propriété, c'est qu'elle permet d'exploiter les qualités allocatives de l'équilibre dit walrasien dans la résolution du (PDG).

La notion d'équilibre walrasien exige que l'on regarde les agents, leurs préférences et les objets à allouer comme formant une économie d'échange pour laquelle on va introduire un système de prix (un prix par objet et valable pour tous les agents, dans le cas le plus simple) ; on définit alors un équilibre pour cette économie comme un vecteur de prix et une allocation tels que chaque agent voit sa demande satisfaite (c'est-à-dire maximise son profit, le profit étant la différence entre la valeur qu'il accorde aux biens qu'il obtient et la somme de leurs prix) (cf. [MWG 95], Part.IV). Dans ce contexte, pour le dire succinctement, une valuation satisfait à la propriété de *gross substitutes* si la demande pour un objet ne diminue pas quand le prix des autres objets augmente.

Définitions

Soient un vecteur de prix $\vec{p} \in \mathbf{R}_+^m$ pour l'ensemble d'objets I et une valuation v_i ;

• le **surplus** d'un ensemble d'objets $S \subseteq I$ pour la valuation v_i relativement au vecteur de prix $\vec{p}$ est $v_i(S/\vec{p}) = v_i(S) - \sum_{j \in S} p_j$

• un ensemble S est un **ensemble préféré** par v_i pour le vecteur $\vec{p}$ si $v_i(S/\vec{p}) = \max_{S' \subseteq I} v_i(S'/\vec{p})$

• l'**ensemble de demande** de la valuation v_i pour le vecteur $\vec{p}$ est l'ensemble des ensemble préférés soit $D(v_i/\vec{p}) = \{S \quad / \quad v_i(S/\vec{p}) = \max_{S' \subseteq I} v_i(S'/\vec{p})\}$

• une valuation v_i a la propriété de ***gross substitutes*** si pour tout objet $j \in I$, pour tout vecteur de prix $\vec{p}$ et tout $\vec{q} \geq \vec{p}$ avec $p_j = q_j$, alors s'il existe $S \subseteq I$ tel que $j \in S$ et $S \in D(v_i/\vec{p})$, il existe $S' \subseteq I$ tel que $j \in S'$ et $S' \in D(v_i/\vec{q})$

Définition

Soit un profil de valuations $v = (v_1, ..., v_N)$; un **équilibre walrasien** est une allocation $S = (S_1, ..., S_N)$ et un vecteur de prix $\vec{p}$ tels que pour tout joueur i, $S_i \in D(v_i/\vec{p})$.

Proposition

L'allocation $S = (S_1, ..., S_N)$ d'un équilibre walrasien est efficace.

Preuve ([GS 99])

Soit $(S, \vec{p})$ un équilibre walrasien, et T une partition de I ; par définition, pour tout joueur i, $v_i(S_i/\vec{p}) \geq v_i(T_i/\vec{p})$ donc

$\sum_{i \in N} v_i(S_i) - \sum_{i \in N} \sum_{j \in S_i} p_j \geq \sum_{i \in N} v_i(T_i) - \sum_{i \in N} \sum_{j \in T_i} p_j$

et donc, pourvu que $\bigcup_{i \in N} S_i = \bigcup_{i \in N} T_i$,

$\sum_{i \in N} v_i(S_i) \geq \sum_{i \in N} v_i(T_i)$ ♠

Théorème (Gul et Stachetti 1999, Kelso et Crawford, 1982)

Toute enchère où le profil de valuations v satisfait la propriété de *gross substitutes* a un équilibre walrasien

Ce théorème fournit un "point d'appui" computationnel : si le profil de valuations satisfait la propriétés de *gross substitues*, il suffit de calculer un équilibre walrasien et de donner la partition qu'il propose comme partition optimale donc solution au (PDG). La Proposition précédente nous garantit que l'on peut se utiliser ainsi l'allocation walrasienne.

Théorème

Soit un profil de valuations qui satisfait la propriété de *gross substitutes* ; alors $V(ZOLP2) = V(RZOLP2)$.

Preuve (d'après [BdVSV 01])

Soit $\vec{p^*}$ une solution à (DRZOLP2) et soit le problème suivant :

$V_{\vec{p}}(RZOLP2) = \max \sum_{i \in N} \sum_{S \subseteq I} v_i(S).x_{(S,i)} - \sum_{j \in I} p_j[\sum_{S \ni j} \sum_{i \in N} x_{(S,i)}] + \sum_{j \in I} p_j$

sujet à

pour tout $S \in \wp(I)$, $i \in N$, $x_{(S,i)} \geq 0$

pour tout $i \in N$, $\sum_{S \in \wp(I)} x_{(S,i)} \leq 1$

On peut calculer que

$V_{\vec{p}}(RZOLP2) = \max \sum_{i \in N} \sum_{S \subseteq I} [v_i(S) - \sum_{j \in I} p_j].x_{(S,i)} + \sum_{j \in I} p_j.$

Puisque $\vec{p^*}$ est une solution duale optimale $V_{\vec{p^*}}(RZOLP2) = V(DRZOLP2)$; donc en vertu du Théorème de Dualité, $V_{\vec{p^*}}(RZOLP2) = V(RZOLP2)$.

Supposons qu'il existe une partition de l'ensemble I des items telle que chaque agent i reçoive au plus un élément de $D(v_i, \vec{p^*})$ (on dit que le vecteur de prix **supporte** cette partition) ; alors, puisque la fonction objectif doit maximiser le surplus des agents, on dispose d'une solution à $V_{\vec{p^*}}(RZOLP2)$ où tout $x_{(S,i)} \in \{0, 1\}$ donc $V_{\vec{p^*}}(RZOLP2) = V(ZOLP2)$. On aura donc bien $V(ZOLP2) = V(RZOLP2)$.

Maintenant, il faut montrer qu'il existe bien une telle partition ; [GS 00] exploitent une connexion avec la théorie des matroïdes dont voici les grandes lignes.

Définitions

• pour une valuation v_i, un ensemble d'objets $A \subseteq I$ et un vecteur de prix $\vec{p}$, on définit la **fonction d'exigence minimale** $r(v_i, A, \vec{p}) = \min_{S \subseteq I \in D(v_i, \vec{p})} |A \bigcap S|$.

Remarques :

- [GS 00] appelle cette fonction de la sorte car, pour chaque sous-ensemble A, elle donne comme valeur le plus petit nombre d'items de A dont on a besoin quand on veut construire un ensemble préféré

- la fonction nous donne une condition nécessaire pour être un ensemble préféré : si $A \in D(v_i, \vec{p})$, alors pour tout $S \subseteq I$, $|(S \bigcap A)| \geq r(v_i, S, \vec{p})$

- en outre, pour tout $A \subseteq I$ tel que pour tout $S \subseteq I$, $|(S \bigcap A)| \geq r(v_i, S, \vec{p})$ il existe $A' \subseteq A$ tel que $A' \in D(v_i, \vec{p})$ ([GS 00], Lemme 1)

• $\rho : 2^I \rightarrow N$ est une **fonction duale de classement** (*dual rank function*) si pour tout $A, B \subseteq I$,

(i) $\rho(\emptyset) = 0$ et $\rho(A \bigcup B) \leq \rho(A) + \rho(B)$

(ii) si $A \supseteq B$, alors $\rho(A) \geq \rho(B)$

(iii)$\rho(A \bigcup B) + \rho(A \bigcap B) \geq \rho(A) + \rho(B)$

Théorèmes (Gul et Stachetti)

Supposons que les valuations satisfont la propriété de *gross substitutes* ;

• pour tout joueur i et tout vecteur $\vec{p} \in R^I_+$, $r_i(v_i, ., \vec{p})$ est une fonction duale de classement

• soit $S \subseteq I$ et deux vecteurs de prix $\vec{p} \geq \vec{q}$ tels que $p_j = q_j$ pour tout $j \in I \backslash S$; alors $r(v_i, S, \vec{p}) \leq r(v_i, S, \vec{q})$.

En vertu du Théorème de partition des matroïdes (Edmonds 1970), s'il n'existe pas de partition de I telle que chaque agent i reçoit au plus un élément de son ensemble de demande $D(v_i, \vec{p})$, il existe un $S \subseteq I$ tel que

$\sum_{i \in N} r_i(v_i, S, \vec{p^*}) > |S|$ $(\star)$

([GS 00], Corollaire du Théorème 3)

On définit alors un nouveau vecteur de prix $\vec{p'}$ tel que $p'_j = p^*_j$ pour tout $j \in I \backslash S$ et $p'_j = p^*_j + \varepsilon$ sinon. Pour chaque agent $i \in N$, on choisit
- $B_i \in D(v_i, \vec{p'})$ tel que $r_i(v_i, S, \vec{p'}) = |(B_i \bigcap S|$

- $C_i \in D(v_i, \vec{p^*})$ tel que $r_i(v_i, S, \vec{p^*}) = |(C_i \bigcap S|$

$V_{\vec{p'}}(RZOLP2) = \sum_{i \in N} [v_i(B_i) - \sum_{j \in I} p'_j] + \sum_{i \in N} p^*_j + \varepsilon.|T|$

$V_{\vec{p'}}(RZOLP2) = \sum_{i \in N} [v_i(B_i) - \sum_{j \in I} p^*_j] - \varepsilon. \sum_{i \in N} |B_i \bigcap S| + \sum_{i \in N} p^*_j + \varepsilon.|T|$

En vertu du second des deux Théorèmes, $|(C_i \bigcap S| \leq |(B_i \bigcap S|$; en outre, $C_i \in D(v_i, \vec{p^*})$. Donc

$V_{\vec{p'}}(RZOLP2) \leq \sum_{i \in N} [v_i(C_i) - \sum_{j \in I} p^*_j] - \varepsilon. \sum_{i \in N} |C_i \bigcap S| + \sum_{i \in N} p^*_j + \varepsilon.|T|$

$V_{\vec{p'}}(RZOLP2) \leq V_{\vec{p^*}}(RZOLP2) - \varepsilon.(\sum_{i \in N} r_i(v_i, S, p^*) - |T|$

Donc par $(\star)$,

$V_{\vec{p'}}(RZOLP2) < V_{\vec{p^*}}(RZOLP2)$.

Mais dans ce cas $\vec{p^*}$ n'est pas une solution duale optimale à (DRZOLP2), ce qui contredit l'hypothèse. ♠

Remarque 1

Il est important de noter que le passage par la notion d'équilibre walrasien a une signification algorithmique bien précise.

(DRZOLP2)

min $\sum_{j \in I} p_j + \sum_{i \in N} z_i$

sujet à

pour tout $j \in I$, $i \in N$, $p_j, z_i \geq 0$

pour tout $i \in N$, $S \subseteq I$, $z_i + \sum_{j \in S} p_j \geq \hat{v}_i(S)$

Si l'on regarde en effet le problème (DRZOLP2), on se rend compte qu'il introduit
- une variable p_j par item $j \in I$
- une variable z_i par agent $i \in N$
Pour tout agent i et tout $S \subseteq I$, il faut que
$z_i \geq v_i(S) - \sum_{j \in S} p_j$
Comme il s'agit de minimiser $\sum_{j \in I} p_j + \sum_{i \in N} z_i$, on peut toujours considérer $z_i = \max_{S \subseteq I}(v_i(S) - \sum_{j \in S} p_j$. Des valeurs p_j^* forment un vecteur de prix walrasien s'il existe en outre une allocation walrasienne adéquate (où chaque agent reçoit un lot d'items de son ensemble de demande et où tous les items sont distribués).

Remarque 2

Dans le Chapitre I, nous avons vu une variante d'un des principaux algorithmes de programmation linéaire, l'algorithme du simplexe. Un autre algorithme, dit algorithme Primal-Dual, prend comme point de départ les *conditions complémentaires* qui découlement du théorème de Dualité. L'algorithme part d'une solution faisable duale ; on calcule ensuite une solution faisable à un problème primal restreint ; si cette solution n'est pas optimale (= si les conditions complémentaires ne sont pas satisfaites), on calcule ensuite une solution duale restreinte qui améliore la solution duale initiale ; et on itère jusqu'à ce que les conditions de complémentarité soient satisfaites.Or, il existe un lien très fort entre les enchères itératives et l'algorithme primal-dual. En effet, dans une enchère itérative, on propose des prix provisoires aux agents et on leur demande ensuite d'enchérir en fonction ; sur la base de ces offres, on calcule une allocation provisoire et on ajuste les prix ; si le résultat n'est pas atteint, on propose à nouveau aux agents de faire des ofres, etc... Avec (RZOLP2) et (DRZOLP2), on peut interpréter l'algorithme primal-dual comme les règles d'une enchère combinatoire itérée [PPhd] : supposons que les agents ne donnent pas une valuation complète sur les ensembles d'objets possibles mais qu'ils la révèlent progressivement au cours de l'enchère. Une solution faisable duale sera contituée par un ensemble de prix et, pour ses prix, par leur surplus maximal. Dès lors, pour avoir une solution duale, il suffit de donner des prix aux agents et de leur demander d'enchérir sur la base de ses prix ; si l'on suppose que les agents maximisent leur surplus, alors on obtient une solution duale complète à partir de laquelle on peut poser un problème primal restreint.

Une enchère itérée, à la différence d'une enchère à révélation directe, est meilleure du point de vue du problème de la communication : dans un mécanisme à révélation directe comme le mécanisme VCG, chaque agent est censé donner sa valuation complète, ce qui est d'une redoutable complexité dès que le nombre d'items augmente. Notons en outre que dans certains cas favorables, les prix de la solution duale optimale sont les transferts de Clarke ; alors la réalisation de l'algorithme primal-dual par une anchère itérée permet d'obtenir les mêmes effets que l'enchère de Clarke ([BdVSV 01], p.14)

Approximation du (PDG)

En l'absence de cas favorables comme ceux que nous venons de voir, il est naturel pour l'organisateur de recourir à des approximations pour résoudre de manière satisfaisante le (PDG). On attend d'un algorithme d'approximation qu'il nous donne en temps polynomial une solution proche de la solution optimale. Néanmoins, les problèmes *NP-hard* s'approximent de manière très différentes

Définitions

• Soit $approx(x)$ la valeur d'une approximation pour une instance x d'un problème de maximisation donné de taille n ; une **borne** $f(n)$ à cette approximation du problème est telle que pour tout instance x du problème, $opt(x) \leq f(n).approx(x)$.

• un **langage** est un ensemble de suites finies de $\{0, 1\}$

• ZPP est la classe des problèmes qui peuvent être résolus en temps polynomial par un algorithme probabiliste qui ne fait jamais d'erreur : un langage L appartient à ZPP ssi il existe une constante c telle qu'il existe une machine de Turing probabiliste M telle que

- la machine répond à une instance x en temps espéré $\mathcal{O}(|x|^c)$

- M accepte x ssi $x \in L$

Pour le problème (MAXCLIQUE), une n-approximation est triviale puisqu'elle donne une solution de taille au moins 1 ; or tout singleton constitue une clique. On voudrait donc trouver des approximations dont les bornes sont inférieures à n. Malheureusement le résultat suivant, basé sur la conjecture $NP \neq ZPP$, montre que dans le pire des cas, c'est impossible : il n'existe pas d'algorithme polynomial dont les solutions soient toujours à $n^{1-\varepsilon}$ de la solution optimale. Notons que cela ne présage évidemment pas de l'existence d'algorithme d'approximation dont le comportement moyen ou empirique soit bon.

Proposition (Hastad 1999)

Soit G un graphe à n sommets ; à moins que $NP = ZPP$, aucun algorithme polynomial approximant (MAXCLIQUE) sur G ne peut garantir une borne $f(n) \leq n^{1-\varepsilon}$ pour $\varepsilon > 0$

Approximation et véracité

Dans cette section, nous nous demandons quelles sont les conséquences de l'approximation du (PDG) pour la DS-véracité (cf. [LOS 99], [PPhd], p.67, [NR 01]). Les résultats d'inapproximabilité nous indiquent que toute algorithme d'approximation polynomiale du (PDG) aura un mauvais comportement pour certaines instances au moins ; la question se pose donc de savoir si ces écarts risquent de rendre le mécanisme manipulable.

En effet, toute la difficulté de l'application d'outils de théorie de l'approximation à la théorie du mécanisme, c'est que l'on cherche à ce que les améliorations computationnelles ne nuisent pas aux propriétés du mécanisme ; plus précisément, on cherche à ce que le mécanisme reste DS-vérace, en dépit de l'approximation de la fonction de choix k :

"*the challenge is to make mechanisms computationally feasible without sacrificing useful game-theoretic properties, such as efficiency and strategy-proofness*" ([PPhd], p.65)

Supposons que pour un profil de valuations révélées $\hat{v}$, l'algorithme d'approximation sélectionne non plus $k^*(\hat{v})$ mais une approximation $k'(\hat{v})$; il est tout à fait possible que dire la vérité ne soit plus une stratégie dominante pour certains joueurs. Du point de vue de la théorie du mécanisme, on ne peut se contenter de chercher des heuristiques de plus en plus efficaces sans regarder l'impact sur la véracité du mécanisme.

Proposition (Ronen 1999)

Soit un quasi-mécanisme VCG dont la fonction d'allocation k approxime l'allocation optimale k^* à une constante $c \in (0, 1)$ près et soit un ensemble d'agents dont les espaces de types sont assez riches ; alors si pour certaines valeurs $k \neq k^*$, le mécanisme n'est pas DS-vérace.

Preuve (d'après [OCPhd])

Si k n'est pas optimal, il existe un profil de types θ tel que $k(\theta) < k^*(\theta)$; si les espaces de types sont assez riches, il existe un joueur i, un nombre q strictement positif et un type ϑ_i tel que
$v_i(k, \vartheta_i) =$
$* \sum_{j \neq i} v_j(k, \theta_j) + q$ si $k = k^*(\theta)$
$* \sum_{j \neq i} v_j(k, \theta_j)$ sinon.

Par hypothèse, pour toute allocation $d \in D$,

$\sum_{i \in N} v_i(k(\vartheta_i, \theta_{-i}), \theta_i) \geq c.v_i(d, \vartheta_i) + \sum_{j \neq i} v_j(d, \theta_j)$

Or si $k(\vartheta_i, \theta_{-i}) \neq k^*(\theta)$, $\sum_{i \in N} v_i(k(\vartheta_i, \theta_{-i}), \theta_i) = 0$ donc $k(\vartheta_i, \theta_{-i}) = k^*(\theta)$. Etant donnée la façon dont un paiement VCG est déterminé,

$u_i(k(\vartheta_i, \theta_{-i}), \theta_i) = u_i(k^*(\theta_i, \theta_{-i}), \theta_i) > u_i(k(\theta_i, \theta_{-i}), \theta_i)$

Donc la véracité n'est pas un DS-équilibre pour le joueur i. ♠

D'une manière générale, le recours à une heuristique nécessite l'analyse suivante :

étape(1)
regarder si l'approximation préserve ou non la véracité du mécanisme

étape (2)
• si c'est le cas, alors on peut considérer l'approximation comme satisfaisante du point de vue du mécanisme.
• si ce n'est pas le cas, on peut jouer sur différents paramètres :
– changer la notion d'équilibre
– changer les fonctions de transfert
– restreindre les situations possibles (par exemple les types possibles de valuation des joueurs)
– changer l'algorithme d'approximation

Analyse d'une approximation, étape (1)

Concernant l'étape (1) de l'analyse, les principaux résultats généraux proviennent de [NR 00] et [KMT 99].

[NR 00] donne une condition suffisante relativement triviale pour qu'une allocation approximée satisfasse à la DS-véracité.

Définitions

• un mécanisme est un **quasi-mécanisme VCG** si la fonction d'allocation k est quelconque et les fonctions de transfert sont celles des mécanismes VCG où k est substituée à la fonction d'allocation efficace.

• soit $k()$ une fonction qui associe à un profil de types révélés une allocation, $\Theta' \subseteq \Theta$ un sous-ensemble de l'espace des profils de types, $r_k(\Theta')$ l'image de Θ' par k. On dit que

– k est **efficace pour l'image de** Θ' si pour tout $\theta \in \Theta'$,
$k(\theta) = \max_{d \in r_k(\Theta')} \sum_{i=1}^n v_i(d, \theta_i)$

– k est **efficace pour son image** si elle est maximale pour l'image de Θ

Exemple

Soit l'algorithme "tout ou rien" qui attribue la totalité des biens à allouer à l'agent qui propose l'offre la plus élevée pour l'ensemble des biens I ; il est clair que cet algorithme est efficace pour son image, bien qu'il ne soit pas nécessairement efficace. $\diamondsuit$

Proposition

Un quasi-mécanisme VCG dont la fonction d'allocation est efficace pour son image est DS-vérace et budgétairement équilibré.

Preuve

Trivial $\spadesuit$

Cette condition nécessaire est également "presque" suffisante.

Définition

soit Θ un espace de types et D un ensemble d'allocations possibles ; on définit $\tilde{\Theta} \subseteq \Theta$ comme l'ensemble des profils de types θ tels que pour tous $d_1, d_2 \in D$, si $d_1 \neq d_2$, alors $\sum_{i=1}^n v_i(d_1, \theta_i) \neq \sum_{i=1}^n v_i(d_2, \theta_i)$.

Théorème ([NR 00])

Si un mécanisme quasi-VCG est vérace, alors il est maximal pour l'image de $\tilde{\Theta}$.

Définitions

• un mécanisme est un **quasi-mécanisme de Clarke** si la fonction d'allocation k est quelconque et les fonctions de transfert sont celles du mécanismes de Clarke où k est substituée à la fonction d'allocation efficace.

• pour tout joueur i, l'ensemble A_i de ses actions possibles contient ses types possibles Θ_i et un message dit *nul* tel que si le joueur envoie le message nul, son transfert est nul. On note $\hat{\theta}^{\circ}$ le profil de messages *nul*.

Soit un quasi-mécanisme de Clarke dont la fonction d'allocation est $k()$; on peut alors définir les propriétés suivantes :

(P1) Pour tout $\hat{\theta} \in \Theta = A - \{\hat{\theta}^\circ\}$, il existe j tel que $\hat{\theta}_j \neq nul$

(P 2) ou **1-efficacité** soient deux profils de types révélés $\hat{\theta}$ et $\hat{\theta}'$ avec leurs profils de valuations respectifs $\hat{v}$, $\hat{v}'$ tel qu'il existe un unique $i \in N$ tel que $\hat{\theta}_i \neq \hat{\theta}_i{}'$; alors

$\sum_{i \in N} \hat{v}_i(k(\hat{\theta}), \hat{\theta}_i) \geq \sum_{i \in N} \hat{v}_i(k(\hat{\theta}'), \hat{\theta}_i)$

(P3) pour tout joueur i et tout profil de types révélés $\hat{\theta}$,

$\sum_{j \neq i} v_j(k(\hat{\theta}_{-i}), \hat{\theta}_j) \geq \sum_{j \neq i} v_j(k(\hat{\theta}), \hat{\theta}_j)$

Le contenu de la propriété (P2) est relativement manifeste : un mécanisme satisfait à (P2) si la satisfaction globale des joueurs ne peut être augmenté par l'issue induite par un changement unilatéral de type révélé. En particulier donc, si les joueurs sont véraces, le mensonge de l'un des joueurs ne permet pas d'obtenir une issue qui les satisfasse mieux globalement.

La propriété (P3), enfin, assure que si un agent substitue à son type révélé un type nul, la satisfaction globale des autres agents ne peut diminuer.

Théorème ([KTM 99])

Si un quasi-mécanisme de Clarke satisfait les propriétés (P1)-(P3), il est DS-vérace et budgétairement équilibré.

Preuve

On ne montrera que la DS-véracité ; il faut donc établir que pour tout agent i, pour tout type θ_i et pour tout profil des autres agents θ_{-i},

$v_i(k(\theta_i, \theta_{-i})), \theta_i) - t_i(\theta_i, \theta_{-i}) \geq v_i(k(\theta'_i, \theta_{-i})), \theta_i) - t_i(\theta'_i, \theta_{-i})$

Par définition,

$t_i(\theta_i, \theta_{-i}) = \sum_{j \neq i} v_j(k_{-i}(\theta_{-i}), \theta_j) - \sum_{j \neq i} v_j(k(\theta_i, \theta_{-i})), \theta_j)$
$t_i(\theta'_i, \theta_{-i}) = \sum_{j \neq i} v_j(k_{-i}(\theta_{-i}), \theta_j) - \sum_{j \neq i} v_j(k(\theta'_i, \theta_{-i})), \theta_j)$

donc il suffit que

$\sum_{i \in N} v_i(k(\theta_i, \theta_{-i})), \theta_i) \geq \sum_{i \in N} v_i(k(\theta'_i, \theta_{-i})), \theta_i)$,

ce qui est garanti par la propriété (P2). ♠

Analyse d'une approximation, étape (2)

[LOS 99] proposent un *algorithme glouton* ("*greedy algorithm*") tel que, si l'on restreint les agents à ne pouvoir faire qu'une seule offre (sur un ensemble d'objets de leur choix), alors il existe des fonctions de transfert (différentes de cellles de l'enchère généralisée de Vickrey) qui rendent le mécanisme DS-vérace.

La restriction de [LOS 99] est la suivante : chaque joueur i est un "**enchérisseur atomique**" ("*single-minded bidder*"), c'est-à-dire qu'il ne fait une offre $\hat{v}_i$ que sur *un seul* sous-ensemble d'objets (qu'il choisit lui-même) de I, s_i ; on considère que i attribue la valeur nulle aux autres sous-ensembles, sauf aux surensembles de s_i (sinon on contredirait évidemment l'hypothèse de *monotonie*).

Définition

un profil de valuation $\hat{v}$ est **atomique** si pour tout i il existe $s_i \subseteq I$ tel que $\hat{v}_i(s_i) > 0$ et si $S' \supseteq s_i$, $\hat{v}_i(S') = \hat{v}_i(s_i)$, sinon $\hat{v}_i(S') = 0$.

Notation

Si une valuation annoncée $\hat{v}_i$ est atomique, elle peut se laisser représenter par un couple $< s_i, m_i >$ où s_i est l'ensemble d'objets pour lequel i offre le montant m_i.

algorithme glouton de [LOS 99] pour le (PDG) (valuations atomiques) :
- pour chaque agent, le montant moyen révélé est : $av_i = m_i/\lvert s_i\rvert^l$ - soit L la liste décroissante des montants moyens révélés - l'algorithme sélectionne le premier membre de la liste ; - le second est sélectionné s'il est compatible avec le premier, - et ainsi de suite jusqu'au n-ième élément de L.

Remarque : le montant moyen révélé satisfait la "bid monotonicity", c'est-à-dire qu'il augmente si la taille de l'ensemble diminue ou si le montant proposé pour l'ensemble augmente.

Quelle est la performance d'un tel algorithme ?

Proposition ([LOS 99])

Si $l = 1/2$, alors l'algorithme glouton fournit une $1/\sqrt{|I|}$-approximation.

Preuve
([LOS 99']) ♠

Mais l'exemple suivant montre qu'adopter une telle approximation fait perdre la DS-véracité à un quasi-mécanisme de Clarke :

Exemple

$I = \{a, b\}$, $N = \{Jean, Pierre, Marc\}$, $\hat{v}_{Jean} =< a, 10 >$, $\hat{v}_{Pierre} =< \{a, b\}, 19 >$ $\hat{v}_{Marc} =< b, 8 >$.

Clairement, l'algorithme glouton alloue $\{a\}$ à Jean et $\{b\}$ à Marc. Maintenant, si l'on regarde la fonction de transfert de Marc, on se rend compte qu'elle s'élève à :

$t_{Jean} = (19 + 0) - (8 + 0) = 11$

donc l'utilité de Marc est de -1. Si Marc avait annoncé une valuation inférieure, il aurait pu ne pas être choisi et ne pas avoir d'utilité négative.

Dans cet exemple, il est clair que la 1-efficacité n'est pas satisfaite puisque précisément la satisfaction globale s'élève à 19 quand Jean manipule le mécanisme et à 18 quand il ne le manipule pas.

$\diamondsuit$

La question qui se pose désormais est celle de savoir si l'on peut recouvrir la véracité ; la stratégie de [LOS 99], reprise depuis par [MN 02], consiste à modifier les fonctions de transfert. L'idée intuitive de cette nouvelle fonction de transfert consiste à faire payer aux agents qui ont une offre gagnante le montant de la valeur critique d'une telle offre relativement aux autres offres, c'est-à-dire la valeur en-dessous de laquelle, étant donné les autres offres, l'offre de l'agent n'aurait plus été gagnante.

Définition

On dit qu'une offre $\hat{v}_i$ est rejetée par une allocation si le joueur i ne se voit pas allouer l'ensemble sur lequel porte l'allocation.

On note $subst(\hat{v}_i)$ la première valuation de la liste L qui est refusée mais serait acceptée s'il n'y avait pas $\hat{v}_i$

Le **transfert glouton** est alors défini de la manière suivante :

- le joueur i ne paie rien si son offre est rejetée ou si $subst(\hat{v}_i)$ n'existe pas
- sinon, pour $\hat{v}_i =< s_i, m_i >$, le joueur i paie $t_i = |s_i|.av_i(subst(\hat{v}_i)$

Théorème ([LOS 99])

Le mécanisme à révélation directe composé de l'algorithme glouton et des transferts gloutons est DS-vérace.

Exemple

Reprenons l'exemple précédent ; clairement, $subst(\hat{v}_{Jean}) = \hat{v}_{Pierre}$ donc désormais $t_{Jean} = 1.(19/2)$ donc

$u_{Jean} = 10 - 19/2 = 1/2$. $\diamondsuit$

Directions de recherches

Dans cette sous-section, on se propose de jeter les bases d'un axe de recherche qui permette de rendre compatible l'approximation et la véracité.

De manière générale, le problème que l'on considère est celui de la compatibilité entre l'approximation du (PDG) et la DS-véracité d'un mécanisme basé sur une telle approximation ; une voie possible de résolution part de l'intuition suivante : si l'on cherche à approximer le (PDG), c'est parce que l'on considère que l'organisateur a des capacités computationnelles limitées. Mais si l'on s'en tient à cette limitation, on manque le fait évident que *les joueurs aussi* sont soumis à des restrictions computationnelles. La question que l'on peut se poser est celle de savoir si, quand on arrive à capturer les limitations des joueurs, l'approximation remet toujours en cause la DS-véracité du mécanisme.

Supposons que l'organisateur approxime avec une fonction d'allocation k^* ; il faut alors examiner les questions suivantes :

<u>Question 3</u> : à quelles conditions la fonction k^* n'est pas DS-vérace ?

<u>Question 4</u> : si elle n'est pas DS-vérace, à quelles conditions computationnelles un agent peut la manipuler à son avantage ?

Voici les notions à partir desquelles on pourrait essayer de répondre conjointement aux questions 3 et 4 :
Définitions

• la **correspondance subjective de meilleure réponse** est une fonction notée $sbr_i : \Theta_i \times \Theta_{-i} \rightarrow \wp(\Theta_i)$ qui prend comme arguments un profil de types des autres joueurs et un type de i, et rend des types de i. Intuitivement, $sbr_i(\theta_i, \hat{\theta}_{-i})$ est l'ensemble des meilleures réponses que le joueur i pense pouvoir donner si son type était θ_i et si les autres joueurs déclaraient $\hat{\theta}_{-i}$.

• un mécanisme est **SDS-vérace** (subjectivement vérace) si pour tout joueur i, et pour tout type $\theta_i \in \Theta_i$ et pour tout $\hat{\theta}_{-i} \in \Theta_{-i}$, $\theta_i \in sbr_i(\theta_i, \hat{\theta}_{-i})$

Fait

Si pour tout joueur i, $sbr_i(\theta_i, \hat{\theta}_{-i}) = br_i(\theta_i, \hat{\theta}_{-i})$, alors la SDS-véracité est identique à la DS-véracité.

Une première chose que l'on peut faire avec la notion de véracité subjective, c'est d'adpater le résultat de [KTM 99].

Définition

Soit un quasi-mécanisme de Clarke de fonction d'allocation k ; on dit qu'il est **subjectivement 1-efficace** (ou s-1-efficace) si pour tout joueur i, pour tout type $\theta_i \in \Theta_i$, pour tout $\theta_{-i} \in \Theta_{-i}$ et pour tout $\vartheta_i \in sbr_i(\theta_i, \theta_{-i})$,

$\sum_{j \in N} v_j(k(\theta), \theta_j) \geq \sum_{j \in N} v_j(k(\vartheta_i, \theta_{-i}), \theta_j)$

Clairement, un mécanisme peut être s-1-efficace sans être 1-efficace ; et si un mécanisme est 1-efficace, il est s-1-efficace.

Proposition

Si un quasi-mécanisme de Clarke est s-1-efficace, alors il est SDS-vérace.

Preuve

Pour qu'un mécanisme soit SDS-vérace, il faut que pour tout joueur i, pour tout type $\theta_i \in \Theta_i$, pour tout $\theta_{-i} \in \Theta_{-i}$ et pour tout $\vartheta_i \in sbr_i(\theta_i, \theta_{-i})$

$v_i(k(\theta_i, \theta_{-i})), \theta_i) - t_i(\theta_i, \theta_{-i}) \geq v_i(k(\vartheta_i, \theta_{-i})), \theta_i) - t_i(\vartheta_i, \theta_{-i})$

donc par le même raisonnement que pour le théorème analogue, il suffit que

$\sum_{j \in N} v_j(k(\theta), \theta_j) \geq \sum_{j \in N} v_j(k(\vartheta_i, \theta_{-i}), \theta_j)$

ce qui est garanti par la s-1-efficacité. ♠

Cette adaptation nous apprend que pour qu'un quasi-mécanisme de Clarke soit SDS-vérace, il suffit qu'aucun joueur ne puisse améliorer la satisfaction globale en déclarant une meilleure réponse subjective plutôt que son vrai type. La SDS-véracité est donc impliquée par le fait qu'aucun agent n'est capable d'augmenter le bien-être total en mentant. Reste à savoir, maintenant, si les notions que l'on vient de définir permettent de trouver des résultats originaux.

Le problème de la complexité des valuations

Quand un agent révèle ses préférences, c'est-à-dire annonce une fonction de valuation, il doit (implicitement au moins) associer une valeur aux $2^{|I|}$ issues possibles. La taille du domaine d'une telle fonction est donc exponentiel par rapport à la taille de l'ensemble des objets mis en jeu.

On peut restreindre les valuations *explicites* des agents ; le point de départ du travail de [LOS 99] que nous avons étudié n'est autre qu'une restriction de ce genre. A partir d'elle, [N 00] a proposé toute une famille de langage d'offres dont le niveau le plus élémentaire est celui des agents "single minded" de [LOS 99].

Il existe donc différents langages possibles pour faire des enchères, auxquels on associe les significations suivantes :

1) *langage atomique* : c'est le cas précédemment étudié : chaque agent propose un montant pour un ensemble d'objets

2) *langage OR* : chaque agent propose un certain nombre d'offres atomiques reliées par un OR ; il est donc implicite que si un joueur a dans ses offres atomiques $< s_1, m_1 >$ et $< s_2, m_2 >$ avec $s_1 \bigcap s_2 = \emptyset$, il est prêt à payer $m_1 + m_2$ pour l'union des deux ensembles.

Proposition Toute valuation sans substituabilité peut être représentée par le langage OR.

3) *langage XOR* : chaque agent propose un certain nombre d'offres atomiques reliées par un XOR ; il est implicite que l'agent souhaite obtenir au plus une de ses offres

Proposition Toute valuation peut être représentée par le langage XOR.

3.4 Routage de données

Une des motivations essentielles d'une approche computationnelle de la théorie du mécanisme vient du problème du routage de données ; l'idée est qu'il faut construire des protocoles qui considèrent les routeurs comme des agents rationnels qu'il faut nécessairement *motiver* à entreprendre les actions qu'on voudrait leur voir entreprendre. On aurait donc affaire selon [NR 01] ou [FPSS 02] à une situation qui mélange des caractéristiques "économiques" (agents interagissants) et computationnelles :

"*we cannot simply expect each computer on the Internet to faithfully follow the designed protocols or algorithms. Is is more reasonable to expect that each selfish computer will try to manipulate it for its owners' benefit. An algorithm or protocol intended for selfish computers must therefore be designed in advance for this kind of behavior!*" ([N 99])

Le problème est le suivant ([N 99]) : on a
- un réseau de communication avec une source s et une destination d ;
- des routeurs qui font parvenir l'information émise par la source à la destination, pour un certain coût.

L'objectif de l'organisateur est de trouver le chemin le plus économique. La situation peut être modélisée de différentes façons :

1/ modèle de [NR 01] et [HS 01]

On représente la situation par un graphe biconnexe G où

- les agents sont les arêtes, chaque arête e étant pondérée par son type $\theta_e \geq 0$, égal au coût de la transmission des données ; on suppose que chaque agent a connaissance de son type et sa valuation v_e est $v_e = -\theta_e$ s'il doit transmettre les données, 0 sinon

- la source et la destination sont deux noeuds distingués du graphe

Cet arrière-plan forme un contexte avec la fonction de choix social dont l'objectif est de minimiser le coût du chemin qui va de la source à la destination. On appelle ce problème de mécanisme (ROUTAGE1).

L'idée est de déterminer les fonctions de transfert t_e de la manière suivante :

- si e n'est pas sur le plus court chemin, $t_e = 0$

- si e est sur le plus court chemin,

$t_e = d_{G/e=\infty} - d_{G/e=0}$

où $d_{G/e=\infty}$ est la longueur du plus court chemin qui ne passe pas par e et $d_{G/e=0}$ est la longueur du plus court chemin si $\theta_e = 0$.

Ces fonctions de transfert tombent dans la famille des fonctions VCG avec $h_i(\hat{\theta}_{-i}) = d_{G/e=\infty}$.Du point de vue algorithmique, toutes les fonctions à calculer par le mécanisme sont des fonctions qui calculent le problème suivant :

Single-Pair LOWEST-COST PATH

données :
un graphe $G = (V, E)$, une fonction de poids $w : E \to \mathbf{R}$, une paire (a, b) de noeuds du graphe
problème :
trouver un chemin $c = (e_0, ..., e_k)$ de a à b tel que $c = \min_{c_{ab}} \sum_{i=1}^{k} w(e_i)$ où c_{ab} est l'ensemble des chemins de a à b.

Comment calculer les fonctions d'allocation et de transfert ? *A priori*, la méthode la plus lourde consiste à calculer un chemin minimal une fois pour l'allocation et ensuite une fois pour chaque arête sur ce chemin, en s'interdisant de passer par l'arête concernée. Pour un tel problème, si les coûts ne sont pas négatifs (hypothèse que l'on peut retenir ici), on dispose de l'algorithme de Disjktra qui le résoud en $\mathcal{O}(n + m \log m)$ où n est le nombre d'arêtes et m le nombre de noeuds (cf. [CLR 90], chap. 25)
Question : peut on faire mieux qu'itérer n fois cet algorithme ?

2/ modèle de [FPSS 02]

Cette fois, la situation est représentée par un graphe biconnexe où les agents i ne sont plus les arêtes mais les noeuds ; les types des agents sont les coûts de routage par paquet, θ_i. Ce second modèle se différencie du premier notamment en ce qu'il ne postule plus un mécanisme centralisé qui calculerait les fonctions de transfert et d'allocation, mais exige que ceci ait lieu de manière distribuée.

Données
- un graphe dont les noeuds sont l'ensemble des agents N
- pour toute paire d'agents i, j, T_{ij} est l'intensité du trafic entre i et j, c'est-à-dire le nombre de paquets qui doivent aller de i en j
- chaque transmission de paquet a pour le noeud i un coût unitaire de θ_i

Notation

Soit z un noeud du graphe ; la fonction $I_z(\theta; i, j)$ a pour valeur 1 si z est sur le chemin le plus économique entre i et j (pour un profil de coûts θ) et est distinct de i et de j ; 0 sinon.

Problème
La fonction de choix social associée au problème est la suivante :

$f(\theta) = \min_{I_k} \sum_{i,j \in N} T_{ij} \sum_{k \in N} I_k(\theta; i, j).\theta_k$

On appelle le problème de mécaisme correspondant à ce nouveau contexte (ROUTAGE2).

Puisque nous sommes toujours sous hypothèse de quasi-linéarité, l'utilité de l'agent z, pour une déclaration de types $\hat{\theta}$ et une fonction d'allocation $k()$ composée de fonctions $I_z(\hat{\theta}; i, j)$ qui calculent les chemins les plus économiques sur la base des types déclarés, est donc :

$u_z(k(\hat{\theta}), \theta_z) = t_z(\hat{\theta}) - \sum_{i,j \in N} T_{ij} I_z(\hat{\theta}; i, j).\theta_z$

où $t_z(\hat{\theta})$ est le transfert que reçoit le joueur z pour la déclaration de types $\hat{\theta}$).

[FPSS 02] proposent un mécanisme VCG pour implémenter ce contexte dont ils prouvent qu'il est le seul mécanisme pour ce contexte (1) à être DS-vérace et (2) à ne pas répartir les prix sur les agents qui ne bénéficient pas des transmissions.

Théorème ([FFPS 02])

Les fonctions de transfert t_k décrites ci-dessous sont les seules à garantir à (ROUTAGE2) une DS-implémentation où les agents non bénéficiares des transmissions n'ont jamais de charges à payer.

$t_k = \sum_{i,j \in N} T_{i,j} t_k^{i,j}$ où

$t_k^{i,j} = I_k(\hat{\theta}; i, j).\hat{\theta}_k + (\sum_{r \in N} I_r(\hat{\theta}_{-k}, \infty; i, j).\hat{\theta}_r - \sum_{r \in N} I_r(\hat{\theta}; i, j).\hat{\theta}_r)$

Preuve
On a un mécanisme VCG où $h_k(\hat{\theta}_{-k}) = \sum_{i,j \in N} T_{i,j} \sum_{r \in N} I_r(\hat{\theta}_{-k}, \infty; i, j).\hat{\theta}_r$; donc le mécanisme est DS-vérace. Il suffit de vérifier que si un joueur k ne fait transiter aucun paquet, il ne reçoit aucun transfert : si k ne fait rien transiter,

$\sum_{i,j \in N} T_{i,j} I_k(\hat{\theta}; i, j).\hat{\theta}_k = 0$

donc

$t_k = \sum_{i,j \in N} T_{i,j}(\sum_{r \in N} I_r(\hat{\theta}_{-k}, \infty; i, j).\hat{\theta}_r - \sum_{r \in N} I_r(\hat{\theta}; i, j).\hat{\theta}_r)$

or clairement si $\hat{\theta}_k = \infty$ et si le graphe est biconnexe, aucun chemin ne passera par k donc

$\sum_{i,j \in N} T_{i,j} \hat{\theta}_{-k}, \infty; i,j).\hat{\theta}_r = \sum_{i,j \in N} T_{i,j} \sum_{r \in N} I_r(\hat{\theta}; i,j).\hat{\theta}_r$

donc $t_k = 0$. ♠

Remarque : les deux exemples que nous avons donnés relèvent des mécanismes VCG ; plus précisément, le but est dans les deux cas d'aboutir à une allocation efficace, et les mécanismes VCG sont la réponse de la théorie du mécanisme à ce problème.

Maintenant, toutes les fonctions de choix social ne sont pas de ce genre, et les mécanismes VCG ne sont plus alors adaptées. C'est le cas par exemple de la planification de tâches étudiée par [NR 01].

3.5 Références

[CLR 90] T.Cormen, C.Leiserson et R.Rivest, Introduction to Algorithms, MIT Press

[DPS 02] Deng, Papadimitriou et Safra, "On the Complexity of Equilibria", STOC 02

[F 02] Feigenbaum, "Agent Privacy in Distributed Algorithmic Mechanisms", Position Paper, http ://www.cs.yale.edu/ jf

[FPSS 02] Feigenbaum, Papadimitriou, Sami et Schenker, "A BGP-based Mechanism for Lowest-Cost Routing", POC 2002

[HS 01] Herschberger et Suri, "Vickrey Prices and Shortest Paths : What is an edge worth ?"

[KTM 99] N.E.Kfir-Dahav, D.Monderer et M.Tennenholtz, "Mechanism Design for Resource Bounded Agents"

[MWG 95] Mas-Collel, Whinston et Green, Microeconomic Theory, OUP, 1995

[LOS 99] Lehman, O'Callaghan et Shoham, "Truth Revelation in Rapid, Approximately Efficient Combinatorial Auctions"

[LLN 01] B.Lehmann, D.Lehmann et N.Nisan, "Combinatorial Auctions with Decreasing Marginal Utilities", EC 2001

[N 99] Nisan, "Algorithms for Selfish Agents", STACS 99

[NR 00] Nisan et Rosen, "Computationally Feasible VCG Mechanisms", EC 2000

[NR 01] Nisan et Rosen, "Algorithmic Mechanism Design", ACEM 2001

[OR 94] Osborne et Rubinstein, A course in Game Theory, MIT Press, 1994

[P 01] Papadimitriou, "Algorithms, Games and the Internet", STOC 01

[PPhd] Parkes, "Iterative Combinatorial Auctions", Phd Thesis

[RPH 95] Rothkopf, Pekec et Harstad, "Computationally manageable combinatorial auctions", DIMACS Technical Report, 95-09

[S 02] Sandholm, "Algorithm for optimal winner determination in combinatorial auctions", Artificial Intelligence, 2002, pp. 1-54

[SSG 01] Sandholm, Suri, Gilpin et Levine, "Winner Determination in COmbinatorial Auction Generalizations"

[T 00] M.Tennenholtz, "Some Tractable Combinatorial Auctions", non publié

[dVV 00] de Vries et Vohra, "Combinatorial Auctions : A Survey", Kellogs School of Management, Technical Report n°1296