
Mécanisme de rétribution pour les systèmes P2P d'échange de fichiers

Comment résoudre le problème du cavalier seul

Maxime Morge* — Philippe Mathieu**

* *Dipartimento di Informatica - Università di Pisa*
via F. Buonarroti, 2
I-56127 Pisa
morge@di.unipi.it

** *Équipe SMAC*
Laboratoire d'Informatique Fondamentale de Lille
Bat M3 - F-59655 Villeneuve d'Ascq cedex
mathieu@lfl.fr

RÉSUMÉ. Les systèmes pair-à-pair d'échange de fichiers constituent un domaine d'application privilégié pour les systèmes multi-agents. Ces systèmes font émerger de nouvelles problématiques en termes de coopération. Notre objectif consiste à adapter le modèle théorique d'allocation de ressources à ce contexte applicatif pour proposer un mécanisme incitant les agents à coopérer. Le mécanisme proposé ici permet de résoudre le problème du cavalier seul. Il permet de converger vers un équilibre de Nash et propose une distribution de ressources qui est optimale et calculable.

ABSTRACT. Peer-to-peer systems for file-sharing consist of a domain which could be adressed by multiagent systems. We adapt the theoretical framework for resource allocation to this application and we propose here a computational mechanism to solve the free-riding problem.

MOTS-CLÉS : P2P, système multi-agent, théorie des jeux computationnelle.

KEYWORDS: P2P, multiagent systems, game theory, computational economics.

DOI:10.3166/ISI.12.3.61-84 © 2007 Lavoisier, Paris

RSTI - ISI - 12/2007. Systèmes d'information pair-à-pair, pages 61 à 84

1. Introduction

Depuis l'apparition de nouveaux protocoles applicatifs comme Gnutella, l'internet constitue une plateforme pour des systèmes pair-à-pair (P2P) d'échange de fichiers. Domaine d'application privilégié pour les systèmes multi-agents (SMA), ces systèmes font émerger de nouvelles problématiques en termes de coopération telles que le problème du cavalier seul (Adar *et al.*, 2000; Groupe Intelligence collective de la Fing, 2005). L'émergence de ces nouveaux usages sur l'internet nécessite que le point de vue économique et le point de vue informatique soient désormais considérés conjointement.

Dans cet article, nous modélisons un système P2P d'échange de fichiers à l'aide d'un système multi-agent. Même si les approches adoptées sont très différentes, ce paradigme a été considérablement étudié que cela soit en Informatique ou en Economie (von Neumann *et al.*, 1944; Hardin, 1968; Groves *et al.*, 1975). En Informatique, les agents logiciels sont considérés comme obéissants, *i.e.* ils exécutent sans autonomie un algorithme. A l'inverse, la théorie des jeux considère que les agents répondent à l'incitation, *i.e.* ils sont autonomes et agissent selon leur propre intérêt. Notre objectif consiste à proposer un mécanisme de rétribution incitant les agents à ne pas se contenter de consommer mais à produire. Nous proposons ici un mécanisme qui est, contrairement au mécanisme de Vickrey-Clarke-Groves, décentralisé et adapté aux échanges P2P. Ce mécanisme aboutit, pour notre stratégie d'offres ascendantes, à une situation du jeu qui est un équilibre de Nash et qui consiste en un partage optimal et calculable de ressources.

Dans un premier temps, nous considérons les échanges de fichiers P2P et plus particulièrement le protocole Gnutella comme un domaine d'application des SMAs (cf. section 2). La section 3 introduit le problème du cavalier seul dans ce contexte applicatif et parcourt quelques-unes des solutions déjà envisagées pour répondre à cette problématique. Dans la section 4, nous adaptons le modèle théorique d'allocation de ressources à ce problème applicatif. Dans cet article, nous présentons et évaluons trois mécanismes pour la distribution d'une ressource. Le premier d'entre-eux est le mécanisme bien connu de Vickrey-Clarke-Groves (Groves *et al.*, 1975), que nous avons adapté à notre contexte applicatif (cf. sections 5). Nous proposons ensuite un mécanisme dans la section 6 qui est, contrairement au précédent, décentralisé. Le dernier mécanisme, proposé dans la section 7, en est une extension qui a pour propriété d'être optimal et stable. Dans la section 8, nous généralisons la dernière de ces procédures à la distribution d'un lot de ressources. La section 9 conclut et dresse quelques-unes des nombreuses perspectives de ce travail.

2. Système P2P d'échange de fichiers

Depuis l'apparition de nouveaux protocoles applicatifs tels que Gnutella¹, l'internet constitue une plateforme pour des systèmes P2P d'échange de fichiers.

Napster² est considéré comme le premier système P2P d'échange de fichiers bien que son architecture soit centralisée. En effet, Napster mettait à disposition sur un serveur une liste des fichiers musicaux disponibles sur le réseau. Chacun pouvait ainsi rechercher puis télécharger des fichiers à l'aide d'un logiciel particulier, appelé **servent** (contraction de « serveur » et de « client »). Comme l'illustre la figure 1, les différents **servents** :

- signalent, à un serveur central, les fichiers dont ils disposent ;
- l'interrogent pour obtenir les coordonnées des fichiers qu'ils recherchent ;
- téléchargent les fichiers auprès d'autres **servents**.

Représentés par leur **servent**, les utilisateurs (alors appelés *pairs*), se comportent soit comme des consommateurs soit comme des producteurs. Ils sont mis en relation afin d'échanger des fichiers.

Gnutella, contrairement à Napster, dispose d'une architecture décentralisée. Il permet de distribuer non seulement le transfert mais également la recherche de fichiers. Comme l'illustre la figure 1, le réseau Gnutella ne dispose pas d'un serveur mais de plusieurs supernœuds branchés en permanence qui propagent les requêtes de recherche au travers du réseau pour mettre en relation les **servents**. Ainsi, on ne dispose plus d'une entité centrale qui supervise les échanges. Puisque les ressources échangées ne sont pas seulement des fichiers (musicaux, vidéos...) mais également des tables de routage, Gnutella fournit les fonctionnalités pour l'établissement d'un réseau *ad hoc*. Lorsqu'un pair souhaite rejoindre le réseau, il doit connaître au préalable un autre pair. C'est la raison pour laquelle, les logiciels pairs sont munis d'une liste d'adresses de supernœuds³.

Un *système multi-agent* (SMA) est un ensemble organisé d'entités informatiques appelées agents qui interagissent dans un environnement. Les SMAs constituent un nouveau paradigme de programmation. Etant donné un domaine d'application, pour savoir si ce paradigme est adapté, voici les bonnes questions à se poser :

-
1. <http://www.gnutella.com>
 2. <http://www.napster.com>
 3. Pour des raisons juridiques, ces derniers ne fournissent ni ne consomment aucun fichier, mais transmettent uniquement les coordonnées d'autres pairs. Dans cet article, nous supposons que, conformément à l'esprit des lois françaises à l'exception de la DAVDSI, la conception et l'usage de systèmes d'échanges de fichiers libres de droits sont légaux.

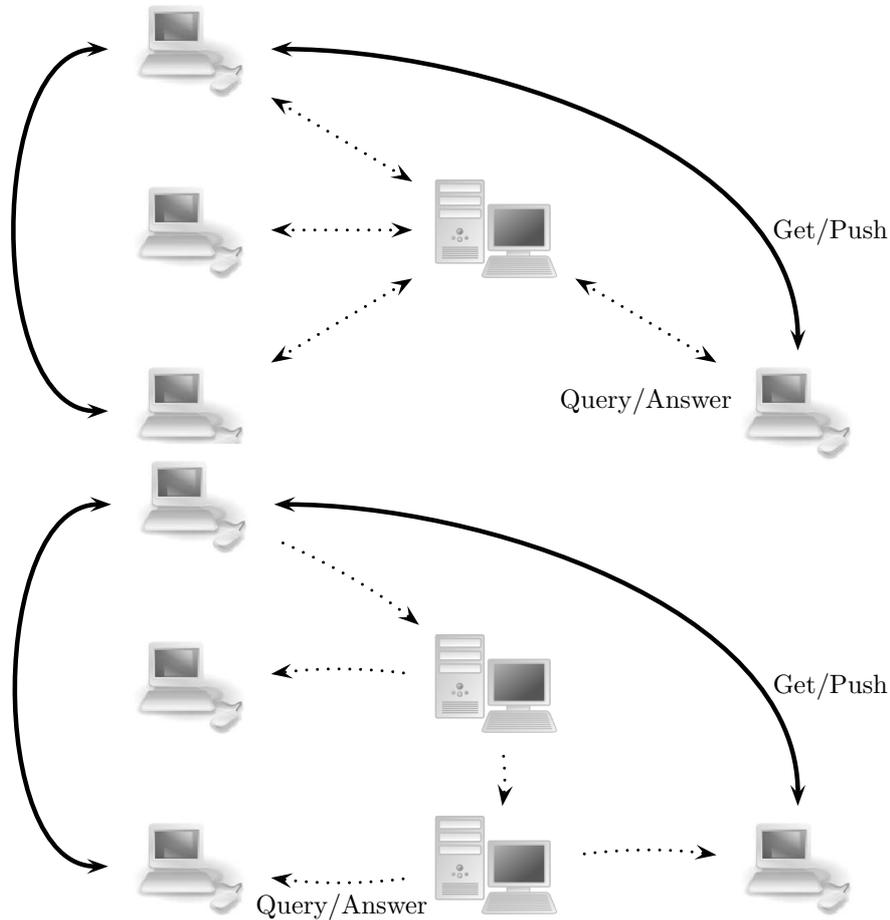


Figure 1. Représentation d'un réseau Napster (haut de la figure) et d'un réseau Gnutella (bas de la figure). Alors que le premier est constitué d'un serveur central et de quatre *servents*, le second est constitué de deux *supernœuds* et de quatre *nœuds simples*. Les échanges pour les recherches de fichiers sont représentés en trait pointillé et les échanges de fichiers sont représentés en trait plein

- **Q1.** Est-ce que le contrôle est décentralisé ?
C'est le cas des systèmes P2P d'échange de fichiers.
- **Q2.** Est-ce que ce domaine est représentable à l'aide de plusieurs entités ?
Oui, les pairs.
- **Q3.** Est-ce que ces entités ont une vue locale sur leur environnement ?
Les pairs ont accès à différentes sources d'information, *i.e.* leurs accointances.

- **Q4.** Est-ce que ces entités ont différents buts/préférences/utilités ?

Puisque les utilisateurs ont des préférences différentes vis-à-vis des fichiers qu'ils souhaitent mettre à disposition ou auxquels ils souhaitent accéder, ces pairs ont des buts différents.

- **Q5.** Est-ce que les relations entre ces entités sont dynamiques ?

Puisque les pairs peuvent entrer et sortir du réseau en cours d'exécution, de tels systèmes sont dits *ouverts*. Par conséquent, les relations organisationnelles entre ces entités sont dynamiques.

Ce sont les raisons pour lesquelles, de tels systèmes, sont de notre point de vue, un domaine d'application privilégié pour les systèmes multi-agents. C'est un nouvel outil pour faciliter le travail au sein d'un groupe d'acteurs au même titre que les places d'enchères virtuelles (Aknine *et al.*, 2005) ou les collecticiels (Morge, 2007) qui ont déjà été envisagés. Dans la suite de cet article, nous utilisons le terme agent (respectivement ressource) en lieu et place de pair (respectivement fichier). On peut noter que ce domaine d'application pose le problème du passage à très grande échelle. Les problèmes qui sont liés au passage à l'échelle (hétérogénéité, robustesse, adaptation dynamique, facteurs humains...) font l'objet de toutes les attentions de la communauté SMA (Briot *et al.*, 2003).

Au travers du protocole Gnutella, les agents interagissent *via* la diffusion ou la rediffusion de messages parmi lesquels on compte :

- des messages **Ping**, afin de s'assurer de la présence d'un agent ;
- des messages **Pong**, afin de répondre aux messages précédents ;
- des messages **Query/Answer** afin de rechercher des ressources ;
- des messages **Get** afin de demander les ressources localisées lors de précédentes recherches.

Afin d'éviter que les messages tournent indéfiniment dans le réseau, ceux-ci sont munis d'une durée de vie (*TTL*, *i.e.* *Time-To-Live*) qui est décrétementée à chaque réception jusqu'à ce qu'elle soit nulle. D'après ce protocole, les agents sont libres de répondre ou non aux sollicitations de leurs pairs.

3. Le problème du cavalier seul

Bien que certaines des caractéristiques de l'architecture P2P soient attrayantes (décentralisation, anonymat), celles-ci font émerger de nouvelles problématiques en termes de coopération comme le problème du cavalier seul.

Selon l'architecture P2P, la quantité et la qualité des ressources partagées ne peuvent être supervisées par une entité centrale. Puisque les agents sont rationnels et ont des capacités limitées (CPU, stockage, bande passante), leur intérêt individuel consiste à consommer sans produire. Toutefois, si tous les

agents adoptent un tel comportement, c'est-à-dire font cavalier seul (Adar *et al.*, 2000; Groupe Intelligence collective de la Fing, 2005), les performances globales du système se dégradent considérablement ce qui a des conséquences pour tous. Ce problème, appelé tragédie des biens communs, est le résultat des tensions qui existent entre l'intérêt individuel et l'intérêt général (Hardin, 1968). De plus, faire cavalier seul rend le réseau plus vulnérable. Si un petit nombre d'agents contribuent au bien commun, ceux-ci agissent en qualité de serveurs centraux et sont vulnérables (déni de service, perte de confidentialité et poursuite juridique).

Exemple 1 (Problème du cavalier seul) *Considérons la distribution de deux ressources \heartsuit et \diamondsuit . Les agents n° 1 et n° 2 souhaitent obtenir la première ressource dont disposent les agents n° 4 et n° 6. L'agent n° 6 veut obtenir la seconde ressource dont disposent les agents n° 1, n° 2 et n° 4. Lorsque les agents n° 1 et n° 2 recherchent la ressource \heartsuit , leur requête est transmise au travers du réseau par les agents n° 3 et n° 5. L'agent n° 6 est autonome, il est donc libre de révéler ou non qu'il en dispose. Son intérêt individuel le pousse à ne pas en informer ses accointances, ce qui ne l'empêche pas d'obtenir la ressource \diamondsuit auprès de l'agent n° 1. En d'autres termes, l'agent n° 6 fait cavalier seul. Son comportement dégrade le système. En effet, les agents n° 1 et n° 2 ont plus de difficultés à obtenir la ressource \heartsuit puisque l'agent n° 6 ne la distribue pas. De plus, l'agent n° 4 tient le rôle de serveur en étant le seul agent du groupe à distribuer la ressource \heartsuit .*

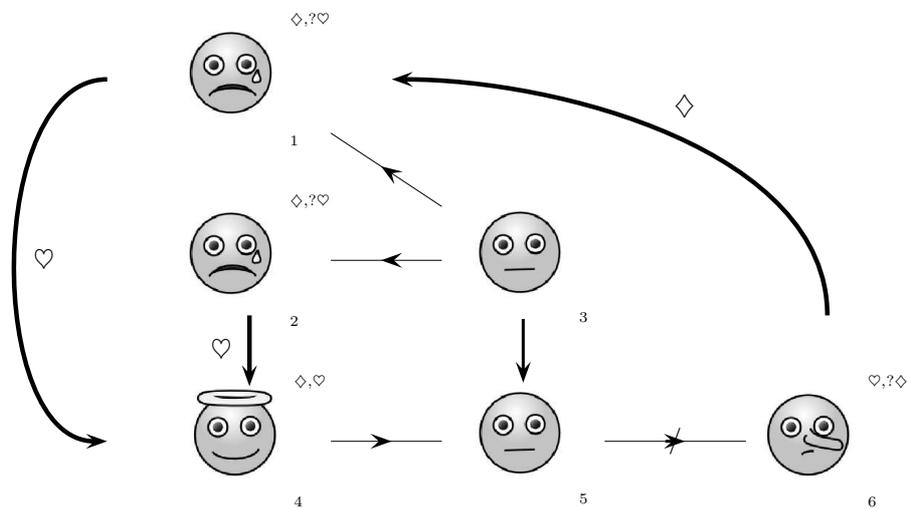


Figure 2. *L'agent n° 6 fait cavalier seul*

eMule⁴ est un logiciel *servent* sous licence GPL qui permet de se connecter au réseau eDonkey et qui décompte le temps passé par un consommateur dans sa file d'attente afin de déterminer sa priorité. Certaines implémentations combinent ce mécanisme de gestion de file d'attente avec un système artificiel⁵ de crédit. Ce mécanisme permet de rétribuer les agents qui coopèrent pour raccourcir d'autant le temps d'attente que les échanges entre les deux agents considérés ont été nombreux. Ainsi, plus un agent fournit de ressources à un second agent, plus le premier agent progresse rapidement dans la file d'attente du second. Toutefois, selon ce mécanisme, seuls les échanges entre ces deux agents sont considérés. En d'autres termes, ce critère est local. De plus, comme nous l'avons signalé précédemment, un nombre restreint de *servent* implémente un tel mécanisme de rétribution.

Dans le réseau BitTorrent⁶, le débit de téléchargement est proportionnel au débit de télé-déchargement. Ceci se veut être une solution au problème du cavalier seul. Malheureusement, cette approche est problématique lorsqu'on considère des systèmes à très grande échelle, puisqu'elle ne prend pas en considération l'hétérogénéité des agents, notamment leurs capacités (bande passante, CPU, stockage) qui peuvent être très différentes.

Lai *et al.* (2003) proposent également un mécanisme incitatif pour les systèmes P2P d'échange de fichiers. Afin de modéliser ce problème, le dilemme itéré des prisonniers (DIP) est envisagé et des stratégies selon lesquelles un agent peut consommer des ressources en fonction de sa propre production ont été étudiées. Pour le lecteur non familier avec le DIP, nous rappelons que celui-ci est un modèle formel de coopération au travers duquel deux agents égoïstes s'affrontent (Mathieu *et al.*, 1999). Comme l'illustre le tableau 1, chacun d'eux a le choix entre deux stratégies : soit coopérer, soit trahir. Le gain de chacun des joueurs dépend de la situation de jeu :

- s'ils ont tous les deux coopéré, ils obtiennent chacun une *récompense* pour coopération, R_c pour le consommateur et R_p pour le producteur ;
- s'ils ont tous les deux trahi, ils obtiennent chacun une *punition* pour égoïsme, respectivement P_c et P_p ;
- si le producteur (respectivement le consommateur) a choisi de trahir et le consommateur (respectivement le producteur) de coopérer, alors le producteur (respectivement le consommateur) se voit attribuer le score de la *trahison* T_p (respectivement T_s) alors que le consommateur (respectivement le producteur) se voit attribuer le salaire de la *dupe*, S_c (respectivement S_p).

La tentation est plus avantageuse que la coopération mutuelle, qui est elle-même plus avantageuse que la punition, qui est plus valorisée que la duperie.

4. <http://www.emule-project.net>

5. Les crédits échangés ne correspondent en aucun cas à un échange monétaire entre les utilisateurs.

6. <http://www.bittorrent.org>

Plus formellement, $T > R > P > S$. Dans ce jeu, la coopération est exploitable par des traîtres. Toutefois, le gain moyen sur plusieurs itérations est plus faible dans ce genre de situation que dans une population constituée uniquement d'agents qui coopèrent. Ce dilemme social semble *a priori* similaire à celui rencontré dans notre contexte applicatif. C'est la raison pour laquelle Lai *et al.* (2003) a considéré ce modèle. Toutefois, le DIP est en fait inadapté à la représentation des interactions entre agents dans un système P2P d'échange de fichiers. En effet, les interactions sont, dans les systèmes P2P, séquentielles et non pas simultanées comme dans le DIP. En conséquence, dans un système P2P, contrairement au DIP, seul le producteur peut choisir entre coopérer et trahir dans le cadre d'un système P2P. De plus, le consommateur ne peut pas distinguer si le producteur le trahit ou s'il coopère sans disposer de la ressource demandée.

	Producteur		
Consommateur		Coopérer	Trahir
Coopérer		(R_c, R_p)	(S_c, T_p)
Trahir		(T_c, S_p)	(P_c, P_p)

Tableau 1. Matrice de gain dans le DIP où le score du joueur de ligne est donné en premier

Notre objectif consiste donc à proposer un mécanisme incitant les agents, dans un système P2P d'échange de fichiers, à produire et ne pas seulement consommer. A cette intention, nous devons tout d'abord adapter le modèle théorique d'allocation de ressources à notre contexte applicatif.

4. Procédure d'allocation de ressources

L'allocation de ressources est un processus qui consiste à distribuer un certain nombre d'unités parmi des agents. Nous pouvons distinguer différents types de ressources. Une ressource peut être un objet physique ou un bien immatériel comme une information digitalisée. Alors que cette caractéristique est intrinsèque à la ressource, d'autres caractéristiques ont attrait au système d'allocation : le fait qu'une ressource soit ou non divisible, par exemple. Ces propriétés ont une influence sur la manière dont la ressource est négociée, c'est-à-dire sur la procédure que l'on doit envisager.

Afin de préciser la sémantique des différentes caractéristiques d'une ressource, le tableau 2 présente plusieurs ressources envisagées dans différents domaines d'application et selon différents points de vue. Lorsqu'un humain achète un bien matériel (e.g. un ordinateur, un CD, de l'énergie), il considère des ressources non *duplicables*. En effet, dupliquer un bien matériel consiste à proposer une nouvelle ressource distincte. A l'inverse, une information digitale (comme un fichier musical) est une ressource duplicable. Nous considérons

ici que les fichiers ne sont pas munis de mesures techniques de protection (en anglais, *DRM* c'est-à-dire *Digital Right Management*). Munir un fichier d'un DRM revient à le rendre *non duplicable*. Les biens matériels considérés dans le e-commerce sont le plus souvent des ressources *discrètes*. A l'inverse les sources d'énergie considérées dans le e-business (e.g. gaz, électricité, pétrole...) sont des ressources *continues*. Contrairement à un ordinateur portable, un ordinateur de bureau peut être acheté en pièces détachées chez un assembleur, c'est une ressource discrète mais *divisible*. Pour un utilisateur, un fichier est une ressource discrète et *indivisible*. En effet, seuls les fichiers dans leur intégralité ont une valeur sur les plateformes de téléchargement. D'un point de vue informatique, les ressources échangées sont divisibles et continues. En effet, dans un système P2P les téléchargements peuvent, contrairement aux téléchargements *via* le protocole ftp, être partiels. Puisque nous nous intéressons à ce domaine d'application, nous souhaitons proposer un mécanisme de rétribution pour l'allocation de ressources continues, divisibles et duplicables. Les crédits, tels qu'ils sont envisagés dans la suite de cet article, sont virtuels. Ils ne correspondent en aucun cas à un échange monétaire entre les utilisateurs.

Points de vue	Ressources	divisible	discrète	duplicable
humain	ordinateur portable		✓	
	ordinateur de bureau	✓	✓	
	énergie	✓		
	CD		✓	
	fichier musical		✓	✓
	fichier musical avec DRM		✓	
informatique	<i>fichier par P2P</i>	✓		✓
	fichier par ftp			✓

Tableau 2. *Types de ressources*

Dans cet article, on considère $\mathcal{A} = \{ag_1, \dots, ag_n\}$ un groupe de $n > 0$ agents et $\mathcal{R} = \{r_1, \dots, r_m\}$ un lot de $m > 0$ ressources. Dans un premier temps, nous envisagerons l'allocation A d'une ressource $r \in \mathcal{R}$ au sein de ce groupe. L'agent ag_1 joue le rôle de *producteur* de la ressource en la distribuant et les autres agents jouent le rôle de *consommateurs* en la téléchargeant. Nous supposons que les différents consommateurs communiquent avec le producteur au travers d'un canal sécurisé. Dans la suite de cet article, ag_i (respectivement ag_j) désigne un consommateur (respectivement un agent) quelconque. Puisque la ressource est continue et divisible, on peut noter $q \in \mathbb{R}_+^*$ la *provision* de cette ressource, c'est-à-dire la quantité disponible de cette ressource. Chaque consommateur ag_i exprime quantitativement ses préférences au travers d'une *fonction de valuation* (notée $v_i(q)$). De manière alternative, ses préférences peuvent être exprimées qualitativement à l'aide d'une relation binaire entre différentes quantités potentielles. Toutefois, l'expression qualitative de préférences n'est pas adaptée à notre contexte applicatif. Elle ne permet ni la comparaison interpersonnelle des préférences ni l'expression de différentes intensités. A

l'inverse, on peut définir pour toute fonction de valuation une relation binaire réflexive, transitive et complète, *i.e.* un ordre total. De même, la tâche d'un producteur, qui consiste à partager une ressource, est associée à une fonction de valuation négative, appelée *fonction de coût* (notée $c(q)$). D'après Groves *et al.* (1975), la fonction de valuation (respectivement la fonction de coût), représentée dans la partie supérieure (respectivement dans la partie inférieure) de la figure 3 est croissante et concave (respectivement strictement décroissante et convexe). Ces fonctions sont dérivables. La fonction inverse $d \equiv c^{-1}$ est définie telle que : $\forall q \in \mathbb{R}_+^* \ d(c(q)) = q$. Celle-ci est décroissante, concave et dérivable ($\frac{\delta d(q)}{\delta q} = \frac{1}{\frac{\delta c(q)}{\delta q}}$).

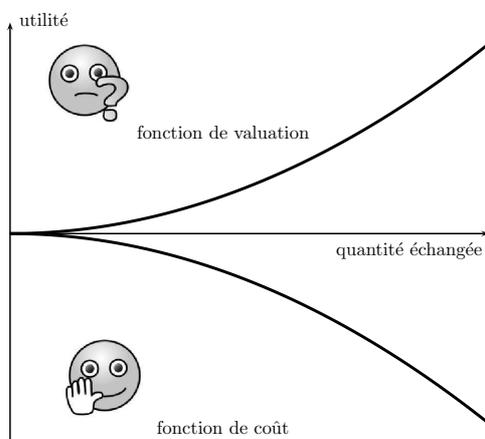


Figure 3. *Fonctions de valuation et de coût*

Le protocole représenté en AUML⁷ (en anglais, *Agent-based Unified Modeling Language*) dans la figure 4 est générique et peut légèrement varier d'une procédure à l'autre. A chaque tour de négociation :

- 1) l consommateurs font des offres (notées $b_i \in \mathbb{R}_+^*$ pour l'agent ag_i);
- 2) le producteur ag_1 informe chaque consommateur ag_i de la quantité q_i qui lui a été affectée et du paiement p_i exigé en retour. Bien entendu, ce dernier est toujours inférieur à l'offre émise.

Le producteur peut lors de cette procédure informer les consommateurs de la provision disponible pour cette ressource (q) et de la somme des paiements obtenus (p). Nous supposons que les paiements s'effectuent au travers d'un service « bancaire » qui centralise les paiements. Si les opérations bancaires sont centralisées, les informations concernant les fichiers échangés ne le sont pas. Un tel service ne peut superviser ni la qualité ni la quantité des ressources

7. <http://www.auml.org>

partagées. De même, la recherche de fichiers et le stockage des fichiers restent décentralisés.

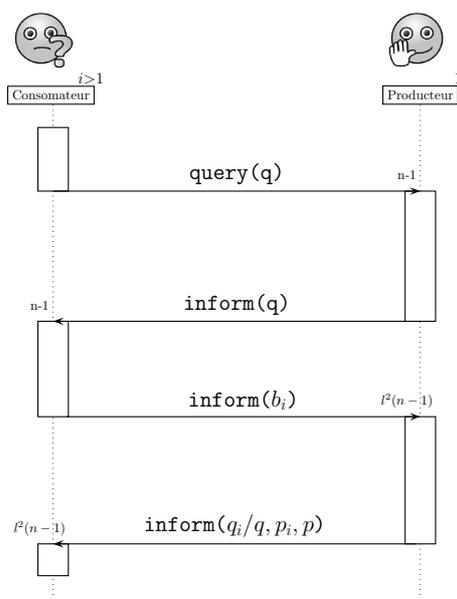


Figure 4. *Protocole de négociation d'une ressource*

Etant donnée une allocation A , la rationalité des agents est économique :

- la satisfaction d'un consommateur ag_i dépend de son estimation de la valeur de la quantité de ressource qui lui est affectée et du paiement effectué en retour : $\pi_i(A) = v_i(q_i) - p_i$;

- la satisfaction du producteur ag_1 dépend du coût de la provision de la ressource et de la somme des paiements : $\pi_1(A) = c_1(q) + p$.

John Rawls (1971) désigne par voile d'ignorance, le principe selon lequel une procédure sociale peut être évaluée par un agent *a priori* indépendamment de la position qu'il y occupera. En d'autres termes, la procédure doit fournir une allocation optimale vis-à-vis d'une métrique qui dépend des préférences individuelles de tous les agents. À cette intention, différentes notions ont été envisagées par la théorie du choix social. Par exemple, une allocation est un *optimum de Pareto* si et seulement si il n'existe pas d'autres allocations qui permettent d'accroître la satisfaction individuelle d'un ou plusieurs agents sans faire diminuer celle d'au moins un autre agent. Puisque ce critère est ordinal, il n'est pas adapté à des préférences quantitatives. Une *allocation ne suscite aucune jalousie* (en anglais, *envy-free*), si chaque agent est au moins aussi

satisfait avec son lot de ressources qu'avec le lot de n'importe quel autre agent. Puisque les ressources sont duplicables et donc attribuables simultanément à plusieurs agents, cette notion est ici dénuée de sens. D'autres critères sont plus adaptés à notre domaine d'application :

– le *bien-être social utilitariste*, qui mesure la somme des gains individuels : $sw(A) = \sum_{j>0} \pi_j(A)$;

– le *bien-être social égalitariste*, qui est défini par le gain de l'agent le plus pauvre : $sw(A) = \min_{j>0} \{\pi_j(A)\}$;

– le *produit de Nash*, qui est le produit des gains individuels : $sw(A) = \prod_{j>0} \pi_j(A)$.

Le bien-être social utilitariste évalue la qualité d'une allocation du point de vue du système global. Le bien-être social égalitariste mesure l'équité de l'allocation. On peut noter que le produit de Nash constitue un bon compromis. Toutefois, ce dernier nécessite que les gains soient tous positifs ce qui ne peut pas être garanti ici. Nous ne considérons dans la suite de cet article que le bien-être social utilitariste.

Pour des raisons de confidentialité, les consommateurs ne révèlent non pas leurs préférences c'est-à-dire leur fonction de valuation mais simplement une offre. Si chaque consommateur ag_i (avec $i > 1$) propose une offre particulière b_i parmi un ensemble d'offres possibles $B_i \subseteq \mathbb{R}_+^*$ alors on appelle *profil de stratégie* ou *situation de jeu*, le vecteur $\vec{s} = (s_2, \dots, s_n)$ qui correspond à l'ensemble des offres : $\forall i > 1 \ s_i = b_i$. Si le producteur ag_1 , qui dispose de la provision q d'une ressource, accepte de lui affecter une partie q_i en échange du paiement p_i , alors l'allocation $A = \langle o, g \rangle$ correspond au résultat de la négociation : d'une part une *distribution*, notée $o = (q_2, \dots, q_n)$, et d'autre part les *paiements*, notés $g = (p_2, \dots, p_n)$. Le gain d'un consommateur ag_i dépend de la situation de jeu : $\pi_i(A) = \pi_i(\vec{s})$. On peut distinguer sa stratégie s_i (en l'occurrence, son offre) de celles des autres participants s_{-i} . On a donc : $\pi_i(A) = \pi_i(\vec{s}) = \pi_i(s_i, s_{-i})$.

Pour une procédure, la stabilité de la situation de jeu est essentielle. A cette intention, la notion d'équilibre en stratégies dominantes est la plus intuitive mais également la plus exigeante. Une stratégie est *dominante* pour l'agent ag_i (notée \hat{s}_i) si et seulement si elle permet d'optimiser son gain quelles que soient les stratégies des autres agents : $\forall s_i \neq \hat{s}_i \ \forall s_{-i} \ \pi_i(\hat{s}_i, s_{-i}) \geq \pi_i(s_i, s_{-i})$. Un *équilibre en stratégie dominante* est une situation pour laquelle tous les agents ont joué une stratégie dominante : $\forall i \geq 1 \ \forall \vec{s} = (s_i, s_{-i}) \ \pi_i(\hat{s}_i, s_{-i}) \geq \pi_i(s_i, s_{-i})$. Puisque les consommateurs ne révèlent pas leurs préférences mais simplement une offre, il n'est pas raisonnable d'espérer pouvoir garantir un équilibre en stratégie dominante. Toutefois, on est en mesure d'exiger un équilibre de Nash. Un *équilibre de Nash* est une situation $\vec{s} = (\hat{s}_2, \dots, \hat{s}_n)$ pour laquelle aucun consommateur n'a intérêt à modifier sa stratégie, compte tenu des stratégies des autres joueurs : $\forall s_i \neq \hat{s}_i \ \pi_i(\hat{s}_i, \hat{s}_{-i}) \geq \pi_i(s_i, \hat{s}_{-i})$.

Ayant contraint le modèle théorique d'allocation de ressources en fonction de notre contexte applicatif, nous souhaitons proposer une procédure (appelée également mécanisme) qui réponde à ces exigences. La conception de mécanisme⁸ est un art qui consiste à élaborer les règles d'un jeu, en l'occurrence une procédure de négociation, pour garantir qu'un certain résultat sera atteint. On dit alors que la procédure implémente le résultat désiré.

La description d'un problème de conception de mécanisme comprend le type des agents impliqués, la spécification d'une situation de jeu et l'objectif de chacun des agents.

Définition 1 (Problème de conception de mécanisme) *On appelle problème de conception de mécanisme un triplet $\langle \vec{\theta}, \vec{s}, \vec{\pi} \rangle$ où :*

– $\vec{\theta} = (\theta_1, \dots, \theta_n)$ est appelé vecteur de types. $\theta_j \in \Theta_j$ correspond à l'état interne de l'agent ag_j ;

– $\vec{s} = (s_2, \dots, s_n) \in \vec{S}$ est appelé vecteur de stratégies où la stratégie d'un consommateur ag_i correspond à son offre b_i ;

– $\vec{\pi} = (\pi_1, \dots, \pi_n)$ et appelé vecteur d'utilité et pour lequel chaque agent ag_j dispose d'une fonction d'utilité $\pi_j(\vec{s}, \theta_j)$ qui donne son évaluation du résultat de la négociation selon son type θ_j .

Les gains dépendent, comme précédemment, de la situation de jeu, mais également du type de l'agent. Le type d'un agent est un paramètre intrinsèque et privé à l'agent. Le type d'un consommateur correspond à son désir d'obtenir une ressource. Le type du fournisseur est intuitivement une mesure quantitative de sa générosité.

L'objectif d'un mécanisme consiste à garantir qu'un résultat sera atteint lorsque les agents tentent de maximiser leur gain.

Définition 2 (Mécanisme) *Soit $\langle \vec{\theta}, \vec{s}, \vec{\pi} \rangle$ un problème de conception de mécanisme. On appelle **mécanisme** d'allocation un couple $M = \langle o, g \rangle$ qui vérifie que :*

– le mécanisme propose à chaque consommateur un ensemble d'offre B_i à partir duquel il en choisit une, notée $b_i \in B_i$;

– le mécanisme affecte une provision q de ressource disponible au producteur à partir de laquelle il choisit la quantité q_i affectée au consommateur ag_i ($\sum_{i>1} q_i = q$) ;

– le mécanisme propose au producteur un ensemble de paiement $P \subseteq \mathbb{R}_+^*$ à partir duquel il choisit le paiement demandé à un consommateur, celui-ci devant être inférieur à son offre ($\forall i > 1 p_i \in P \wedge p_i \leq b_i$) ;

8. Parfois appelée théorie de jeux inverse.

- le mécanisme garantit que le résultat $o = (q_2, \dots, q_n) \in O$ est atteint ;
- le mécanisme détermine les paiements $g = (p_2, \dots, p_n) \in G$.

Nous faisons l'hypothèse selon laquelle l'ensemble des types Θ et l'ensemble des offres B sont prédéterminés et communs à l'ensemble des agents.

La *fonction de choix social* ($\text{scf} : \Theta^n \times \vec{S} \rightarrow O \times G$) associe une allocation à chaque vecteur de type et à chaque profil de stratégie. Comme vu précédemment, différents critères peuvent être envisagés pour définir l'efficacité de l'allocation. Nous souhaitons ici maximiser le bien-être social utilitariste :

$$\text{scf}(\vec{\theta}, \vec{s}) = \max_{o \in O} \sum_{j>0} \pi_j(\vec{s}, \theta_j)$$

En d'autres termes, le vecteur des quantités $\vec{q}^* = (q_2^*, \dots, q_n^*)$ est *optimal* ssi le bien-être social utilitariste est maximal :

$$\forall \vec{q} \in O \quad \sum_{i>1} (v_i(q_i^*) - c(q)) \geq \sum_{i>1} (v_i(q_i) - c(q)) \quad [1]$$

De plus, on souhaite garantir que la situation de jeu soit un équilibre de Nash. En résumé, nous souhaitons proposer un mécanisme qui implémente un résultat optimal, c'est-à-dire qui propose une distribution optimale, et qui aboutit à un équilibre de Nash. Le mécanisme le plus connu est celui de Vickrey-Clarke-Groves (VCG).

5. Mécanisme de Vickrey-Clarke-Groves

Dans cette section, nous souhaitons adapter ici le mécanisme de Vickrey-Clarke-Groves (Groves *et al.*, 1975) à notre contexte applicatif.

Une enchère est un mécanisme qui permet de déterminer, pour une ressource mise en jeu, qui l'obtient et à quel prix. Selon l'enchère de Vickrey⁹ (appelée également en anglais *second-price sealed-bid auction*), chaque enchérisseur remet une offre confidentielle, sous pli cacheté et scellé (*via* un canal de communication sécurisé), au commissaire-priseur (le producteur) qui examine toutes les offres. La ressource est attribuée au plus offrant, qui paie le prix offert par le deuxième meilleur enchérisseur. Cette enchère incite les enchérisseurs à ne pas adopter un comportement stratégique c'est-à-dire à révéler la valeur qu'il assigne véritablement à la ressource. Ce mécanisme permet de distribuer une seule ressource indivisible.

Le mécanisme d'enchère à prix uniforme (en anglais, *uniform-price auction*) est fondé sur le même principe et permet de distribuer une ressource divisible. Il consiste à faire payer à tous les vainqueurs de l'enchère l'offre perdante la plus élevée. Malheureusement, ce mécanisme incite les enchérisseurs à adopter

9. Proposé par William Spencer Vickrey (1914-1996) prix Nobel d'économie en 1996.

un comportement stratégique. C'est la raison pour laquelle le mécanisme de Vickrey-Clarke-Groves a été introduit. Il incite les enchérisseurs à révéler les valeurs qu'ils assignent véritablement aux différentes distributions possibles d'une ressource divisible¹⁰.

Nous souhaitons adapter le mécanisme de Vickrey-Clarke-Groves à la distribution d'une ressource continue, divisible et duplicable. A cette attention, les enchérisseurs ne doivent pas seulement soumettre une offre mais une fonction qui détermine pour chaque distribution l'offre correspondante. Etant donné le vecteur de ces fonctions d'offre (noté $b(\vec{q})$), le mécanisme que nous proposons (noté $M_{VCG} = \langle o_{VCG}, g_{VCG} \rangle$) est constitué :

1) de la distribution suivante :

$$o_{VCG}(b(\vec{q})) = \operatorname{argmax}_{\vec{q}}[(\sum_{i>1} b_i(q_i)) - c(q)]$$

2) des paiements calculés tels que $\forall i > 1$:

$$p_i(b(\vec{q})) = \max_{\vec{q}}[\sum_{k>1, k \neq i} b_k(q_k) - c(q)] - \left(\sum_{k>1, k \neq i} [b_k(o_{VCG}(b(\vec{q}))) - c(o_{VCG}(b(\vec{q}))) \right]$$

Nous souhaitons ainsi maximiser la somme des offres et faire payer à chaque enchérisseur ce que coûte sa présence. Comme le producteur calcule l'allocation à partir des fonctions d'offre révélées, le contrôle de ce mécanisme est centralisé.

Exemple 2 (Vickrey-Clarke-Groves) *Considérons la distribution d'une unique ressource au sein d'un groupe de quatre agents : un producteur ag_1 et trois consommateurs ag_2 , ag_3 et ag_4 . Le producteur dispose d'une provision $q = 2$ qu'il peut soit distribuer entièrement à un seul consommateur soit diviser en deux pour la distribuer à deux consommateurs de son choix. Les fonctions, qui déterminent pour une distribution les offres correspondantes, sont définies de la manière suivante :*

- l'enchérisseur ag_2 désire la quantité $q_2 \geq 1$ et offre 5 € ;
- l'enchérisseur ag_3 désire la quantité $q_3 \geq 1$ et offre 4 € ;
- l'enchérisseur ag_4 désire la quantité $q_4 = 2$ et offre 6 €.

Affecter la totalité de la provision à l'agent ag_4 ne maximise pas la somme des offres. Puisque ce mécanisme incite les enchérisseurs à révéler les valeurs qu'ils assignent véritablement aux différentes distributions, la valeur selon l'agent ag_4 (respectivement les agents ag_2 et ag_3) de cette distribution est de 6 € (respectivement 0 €).

10. Sans perte de généralité, nous faisons l'hypothèse que l'ensemble des consommateurs sont des enchérisseurs.

Distribuer $q_2 = 1$ et $q_3 = 1$ maximise la somme des offres. La valeur selon l'agent ag_2 (respectivement ag_3) de cette distribution est de 5 € (respectivement 4 €). Évidemment cette distribution n'a aucune valeur pour l'agent ag_4 .

En l'absence de l'agent ag_2 , l'agent ag_4 aurait remporté l'enchère. L'agent ag_2 doit donc payer $6 - 5 = 1$ €. En l'absence de l'agent ag_3 , l'agent ag_4 aurait remporté l'enchère. L'agent ag_3 doit donc payer $6 - 4 = 2$ €. Que l'agent ag_4 soit ou non présent, la distribution reste inchangée. Il ne doit donc rien payer.

Ces résultats sont résumés dans le tableau 3. D'après ce mécanisme, les fonctions de valuation de chacun des consommateurs sont transmises au producteur qui seul calcule le résultat de jeu afin de maximiser le bien-être social utilitariste.

Résultats	v_2	v_3	v_4	p_2	p_3	p_4
$q_2 = 0 \wedge q_3 = 0 \wedge q_4 = 2$	0	0	6	0	0	3
$q_2 = 1 \wedge q_3 = 1 \wedge q_4 = 0$	5	4	0	1	2	0

Tableau 3. Distributions et allocation via le mécanisme de Vickrey-Clarke-Groves

Nous avons adapté ici le mécanisme de Vickrey-Clarke-Groves (Groves *et al.*, 1975) à la distribution d'une ressource continue, divisible et duplicable. Malheureusement, un tel mécanisme exige des consommateurs de révéler leurs fonctions d'offre, Afin de palier cette carence en termes de confidentialité, nous souhaitons proposer dans la section suivante un mécanisme décentralisé.

6. Mécanisme décentralisé non optimal

Nous proposons ici un mécanisme décentralisé très simple et nous évaluons ses propriétés.

Etant donné que nous souhaitons proposer un mécanisme décentralisé, celui-ci doit se contenter d'un vecteur d'offres pour implémenter une allocation. Nous proposons ici le mécanisme le plus simple possible (noté $\check{M} = \langle \check{o}, \check{g} \rangle$). On note B la somme des offres ($B = \sum_{i>1} b_i$). Notre mécanisme est constitué :

- 1) de la distribution $\check{o}(b(q)) = d(B)$;
- 2) des paiements calculés tels que $\forall i > 1 \check{p}_i(b(q)) = b_i$.

Malheureusement, ce mécanisme ne permet pas toujours d'aboutir à une allocation satisfaisante¹¹.

11. Toutes les preuves des théorèmes présentés dans cet article sont dans l'appendice A.

Théorème 1 (Mécanisme non optimal) *Considérons le mécanisme $\check{M} = \langle \check{\delta}, \check{g} \rangle$. S'il existe plus de deux consommateurs, ce mécanisme ne permet pas d'implémenter une allocation qui soit optimale et d'aboutir à un équilibre de Nash quelles que soient les fonctions de valuation et la fonction de coût.*

L'exemple 3 illustre ce résultat négatif.

Exemple 3 (Mécanisme non optimal) *Comme précédemment, nous considérons la distribution d'une unique ressource au sein d'un groupe de quatre agents : un producteur ag_1 et trois consommateurs ag_2 , ag_3 et ag_4 . Nous supposons :*

- la fonction de coût est : $c(q) = -5q$;
- la valuation de l'agent ag_2 est $v_2(q) = 7$ si $q \geq 1$;
- la valuation de l'agent ag_3 est $v_3(q) = 2$ si $q \geq 1$;
- la valuation de l'agent ag_4 est $v_4(q) = 10$ si $q = 2$.

Dans la situation de jeu où l'enchérisseur ag_2 offre 4 €, l'enchérisseur ag_3 offre 5 € et l'enchérisseur ag_4 offre 10 €. Distribuer une provision $q = 2$ à l'agent ag_4 contre un paiement de 10 € lui permet de recouvrir le coût de cette production. Dans cette situation, cette distribution est optimale. Toutefois cette situation n'est pas un équilibre de Nash. En effet, l'agent ag_2 a intérêt à modifier sa stratégie compte tenu des stratégies des autres enchérisseurs. S'il offre non plus 4 € mais 6 €, son gain est supérieur. Ces résultats sont résumés dans le tableau 4.

Le mécanisme que nous avons proposé ici est décentralisé, puisque la distribution et les fonctions de paiement sont déterminées par le producteur en fonction du vecteur d'offres qui elles sont calculées par les consommateurs. Malheureusement ce mécanisme n'est ni optimal ni stable.

7. Mécanisme décentralisé, optimal et stable

Dans cette section, nous souhaitons proposer un mécanisme décentralisé, optimal et stable.

Nous proposons ici le mécanisme $\hat{M} = \langle \hat{\delta}, \hat{g} \rangle$ qui spécifie pour l'ensemble des offres :

- 1) la distribution $\hat{\delta}(\vec{b}) = d(B)$;
- 2) les paiements calculés tels que $\forall i > 1$:

$$p_i(\vec{b}) = b_i - B_{-i} \log\left(1 + \frac{b_i}{B_{-i}}\right) \text{ avec } B_{-i} = \sum_{k \neq i, k > 1} b_k$$

$c(q) =$	$-5q$	
$v_2(q) =$	$7 \text{ si } q \geq 1$	
$v_3(q) =$	$2 \text{ si } q \geq 1$	
$v_4(q) =$	$10 \text{ si } q = 2$	
Situation	$b_2 = 4 \wedge b_3 = 5 \wedge b_4 = 10$	$b_2 = 6 \wedge b_3 = 5 \wedge b_4 = 10$
$p_2 =$	0	6
$p_3 =$	0	5
$p_4 =$	10	0
$p =$	10	11
$q =$	2	$22/10$
$q_2 =$	0	$6/5$
$q_3 =$	0	1
$q_4 =$	2	0
$\pi_1(\vec{s}) =$	0	0
$\pi_2(\vec{s}) =$	0	1
$\pi_3(\vec{s}) =$	0	-3
$\pi_4(\vec{s}) =$	0	0
$\sum_{j>0} \pi_i(\vec{s})$	0	-2

Tableau 4. Distributions et allocation via le mécanisme non optimal

On vérifie bien que selon ce mécanisme, un consommateur ne paie jamais plus que le montant de son offre ($0 < p_i < b_i$).

Comme le montre le théorème 2, le mécanisme \hat{M} , contrairement au mécanisme précédent, permet d'aboutir à un résultat satisfaisant.

Théorème 2 (Mécanisme optimal et stable) *Considérons le mécanisme $\hat{M} = \langle \hat{o}, \hat{g} \rangle$. Selon ce mécanisme, les situations qui sont des équilibres de Nash donnent lieu à des résultats qui sont optimaux.*

Exemple 4 (Mécanisme optimal et stable) *Reprenons l'exemple envisagé précédemment. Quelle que soit la situation, notamment celle représentée dans le tableau 5, la distribution est identique à celle du précédent mécanisme mais le paiement exigé est toujours inférieur à l'offre.*

La subvention d'une allocation permet d'évaluer son équilibre budgétaire : $S = B - \sum_{i>1} p_i(\vec{b})$. Pour notre mécanisme, celle-ci peut être bornée par une fraction du coût total : $S \leq B(n-1) \log \frac{n}{n-1}$

Considérons une procédure au cours de laquelle ce mécanisme est itéré. Afin que les agents convergent vers un équilibre de Nash, les agents doivent suivre une stratégie d'offre ascendante :

$c(q) =$	$-5q$
$v_2(q) =$	$7 \text{ si } q \geq 1$
$v_3(q) =$	$2 \text{ si } q \geq 1$
$v_4(q) =$	$10 \text{ si } q = 2$
Situation	$b_2 = 4 \wedge b_3 = 5 \wedge b_4 = 10$
$p_2 =$	0
$p_3 =$	0
$p_4 =$	$\sim 7,07$
$p =$	$\sim 7,07$
$q =$	2
$q_2 =$	0
$q_3 =$	0
$q_4 =$	2
$\pi_1(\vec{s}) =$	$\sim 2,93$
$\pi_2(\vec{s}) =$	0
$\pi_3(\vec{s}) =$	0
$\pi_4(\vec{s}) =$	$\sim -2,93$
$\sum_{j>0} \pi_j(\vec{s})$	0

Tableau 5. *Distribution via le mécanisme optimal et stable*

Théorème 3 (Stratégie convergente) *Soit $|R|$ le nombre de tour de négociation. Si les consommateurs adoptent la stratégie d'offre ascendante suivante :*

$$\begin{aligned} \frac{\delta b_i}{\delta |R|} &= \frac{\delta}{\delta i} \pi_i(b_i, B_{-i}) \\ &= \frac{\delta}{\delta |R|} v_i(d(B)) \cdot \frac{\delta}{\delta |R|} d(B) - \frac{b_i}{B} \end{aligned}$$

alors le profil des stratégies \vec{s} converge vers un équilibre de Nash et les allocations sont optimales.

On peut noter que pour surenchérir, les agents doivent connaître du tour de négociation précédent la provision de la ressource et la somme des gains offerts par les consommateurs sans en connaître le détail (c'est-à-dire les offres émises par chacun des consommateurs et le nombre de consommateurs).

La complexité computationnelle liée à ce mécanisme est linéaire selon le nombre de consommateurs. De même, la complexité communicationnelle, c'est-à-dire le nombre de messages échangés est linéaire selon le nombre de consommateurs. De plus, le langage de communication n'est pas nécessairement très expressif.

Nous avons proposé ici un mécanisme décentralisé pour la distribution d'une ressource continue, divisible et duplicable. Celui-ci peut être itéré. Nous avons alors garanti que, pour la stratégie d'offre ascendante identifiée ici, ces différents

mécanismes implémentent des allocations optimales et calculables. De plus, les agents convergent vers un équilibre de Nash.

8. Lot de ressources

Dans cette section, nous étendons le mécanisme précédent à la distribution d'un lot de ressources.

Considérons la production de m ressources. Le vecteur des provisions distribuées est noté $\vec{q} = (q^1, \dots, q^m)$. Ces quantités sont toutes strictement positives ($q^l \in \mathbb{R}_+^*$, avec $1 \leq l \leq m$). Comme nous l'avons fait précédemment, il est raisonnable d'effectuer les hypothèses suivantes concernant les fonctions de valuation et de coût. Chaque consommateur ag_i est muni d'une fonction de valuation $v_i(\vec{q})$ qui est concave et strictement croissante en chaque coordonnée. Nous faisons l'hypothèse qu'il n'existe aucune synergie, positive ou négative, entre ces ressources. En d'autres termes, la valeur associée par un agent à une ressource est indépendante du fait qu'il dispose ou non d'une autre ressource. Les coûts de production ($c^l(q^l), 1 \leq l \leq m$) associés à ces quantités sont convexes et strictement décroissantes. Toutes ces fonctions sont dérivables. Les fonctions inverses $d^l \equiv c_{-1}^l$ sont définies telles que : $\forall l, 1 \leq l \leq m \forall q^l \in \mathbb{R}_+^* d^l(c^l(q^l)) = q^l$. Les fonctions d sont décroissantes, concaves et dérivables ($\frac{\delta d^l(q^l)}{\delta q^l} = \frac{1}{\frac{\delta c^l(q^l)}{\delta q^l}}$).

Quel que soit le mécanisme envisagé pour la distribution d'un lot de ressources, il procède de la manière suivante. Dans un premier temps, chaque consommateur ag_i doit soumettre un vecteur d'offre $\vec{b}_i \in (\mathbb{R}_+^*)^m$. Le producteur doit alors mettre en correspondance le vecteur de ces vecteurs d'offres \vec{b} avec d'une part une distribution, *i.e.* le vecteur des vecteurs quantités affectés à chacun des consommateurs ($\vec{o} = (\vec{q}_2, \dots, \vec{q}_n)$) et d'autre part, les paiements qui sont affectés à chacun des consommateurs. Comme précédemment, nous souhaitons maximiser le bien-être social utilitariste : $\text{scf}(\vec{b}) = \max_{o \in O} \sum_{j>0} \pi_j(o(\vec{s}), \theta_j)$. En d'autres termes, le vecteur des vecteurs $\vec{q}^* = (q_2^*, \dots, q_n^*)$ est *optimal* ssi le bien-être social utilitariste est maximal :

$$\forall \vec{q} \in (\mathbb{R}_+^*)^m \sum_{i>1} (v_i(q_i^*) - c(\vec{q})) \geq \sum_{i>1} (v_i(\vec{q}_i) - c(\vec{q}))$$

Nous proposons ici le mécanisme $\hat{M} = \langle \hat{o}, \hat{g} \rangle$ qui est constitué :

1) de la distribution suivante :

$$\forall l, 1 \leq l \leq m \hat{o}^l(b^l) = d^l(B^l)$$

2) des paiements calculés tels que $\forall i > 1$:

$$p_i(\vec{b}) = \sum_{l=1}^{l=m} p_i^l(b_i^l) = \sum_{l=1}^{l=m} b_i^l - B_{-i}^m \log\left(1 + \frac{b_i^l}{B_{-i}^l}\right)$$

avec $B_{-i}^l = \sum_{k \neq i, k > 0} b_k^l$.

Comme le mécanisme précédent, \hat{M} permet d'aboutir à un résultat satisfaisant.

Théorème 4 (Mécanisme multiressource) *Considérons le mécanisme $\hat{M} = \langle \hat{o}, \hat{g} \rangle$. Celui-ci implémente une distribution optimale dans une situation d'équilibre de Nash.*

La démonstration est identique à celle du théorème 2.

Nous avons proposé ici un mécanisme décentralisé pour la distribution d'un lot de ressources continues, divisibles et duplicables. Ce mécanisme garantit que la distribution est optimale, calculable et stable.

Nous avons proposé ici un mécanisme décentralisé pour la distribution d'un lot de ressources continues, divisibles et duplicables sans synergie. Celui-ci implémente une allocation optimale et calculable.

9. Conclusions et perspectives

Dans cet article, nous avons montré que les systèmes P2P d'échange de fichiers sont un domaine d'application privilégié pour les SMAs. Bien que certaines des caractéristiques de l'architecture P2P soient attrayantes, nous avons vu que celle-ci fait émerger des comportements problématiques : les pairs ont intérêt à télécharger sans mettre à disposition de fichiers. Afin de proposer une solution à ce problème qui soit robuste au passage à l'échelle, nous avons souhaité proposer un mécanisme de rétribution des producteurs. Puisque nous nous intéressons aux systèmes P2P d'échange de fichiers, nous nous sommes focalisés sur les procédures d'allocation de ressources continues, divisibles et duplicables. Nous avons proposé un mécanisme itéré et décentralisé qui implémente une distribution optimale et qui converge vers un équilibre de Nash.

D'un point de vue théorique, Feigenbaum *et al.* (2002) mettent en exergue les fondements de la conception de mécanisme décentralisé et présente quelques résultats récents ainsi que quelques-uns des problèmes ouverts. C'est sur celui des systèmes P2P d'échange de fichiers que nous nous sommes focalisés ici. Comme le préconisent Chevaleyre *et al.* (2006), nous avons évalué les contraintes que notre contexte applicatif impose sur le modèle information d'allocation et nous avons adopté un point de vue économique afin de concevoir un mécanisme « efficace ». Nous avons proposé un mécanisme qui implémente une distribution

optimale. Comme le bien-être social utilitariste, le bien-être social égalitariste est un critère d'évaluation pertinent dans notre contexte applicatif. Nous souhaitons proposer prochainement un mécanisme qui implémente une distribution équitable et étudier sous quelles conditions ces deux types de distributions peuvent coïncider. Parkes *et al.* (2004) proposent une version décentralisée du mécanisme de Vickrey-Clarke-Groves. Comme l'exige notre contexte applicatif, nous souhaitons l'adapter à l'allocation de ressource duplicable. Quel que soit le mécanisme proposé, nous devons non seulement vérifier que celui-ci est décentralisé mais aussi qu'il incite les agents à ne pas adopter de comportement stratégique. A cette intention, nous souhaitons nous inspirer de Petcu *et al.* (2006) qui mixent les techniques de conception de mécanisme et d'optimisation de contraintes distribuées. Finalement, lorsque nous avons considéré l'allocation de lots de ressources, nous n'avons envisagé aucune synergie entre ressources. Bien entendu, cette hypothèse mériterait d'être levée.

D'un point de vue pratique, nous souhaitons valider notre approche à l'aide de simulations.

Remerciements

Le premier auteur est soutenu par le 6^e programme IST de la CE, au travers du projet ARGUGRID 035200. Le travail du second auteur a été cofinancé par le CPER TAC de la région Nord-Pas de Calais et les fonds européens FEDER. Nous remercions les relecteurs anonymes pour leurs commentaires détaillés de ce travail.

10. Bibliographie

- Adar E., Huberman B. A., "Free Riding on Gnutella", *First Monday*, October, 2000. http://firstmonday.org/issues/issue5_10/adar/index.html.
- Aknine S., Pinson S., « Modèles de Négociations Multi-agents Combinées », *Actes des Journées Francophones sur les Systèmes Multi-Agents*, Hermès, 2005.
- Briot J.-P., Ghedira K. (eds), « Déploiement des systèmes multi-agents », *Revue des Sciences et Technologies de l'Information*, Hermès Sciences Publication, Hammamet, Tunisie, 2003.
- Chevalyere Y., Dunne P. E., Endriss U., Lang J., Lemaître M., Maudet N., Padget J., Phelps S., Rodríguez-Aguilar J. A., Sousa P., "Issues in Multiagent Resource Allocation", *Informatica*, 2006, vol. 30, p. 3-31.
- Feigenbaum J., Shenker S., "Distributed algorithmic mechanism design : recent results and future directions", *DIALM '02 : Proceedings of the 6th international workshop on Discrete algorithms and methods for mobile computing and communications*, 2002, ACM Press, New York, NY, USA, p. 1-13.
- Groupe Intelligence collective de la Fing, « La coordination de groupes », Portail InternetActu.net, aout, 2005. <http://www.internetactu.net/index.php?p=6119>.

- Groves T., Loeb M., “Incentives and public inputs”, *J. of Public Economics*, 1975.
- Hardin G., “The Tragedy of the Commons”, *Science*, 1968, vol. 162, p. 1243-1248.
- Lai K., Feldman M., Stoica I., Chuang J., “Incentive for Cooperation in P2P Networks”, *Proc. of the 1st Workshop Economics of Peer-to-Peer systems*, Berkley, 2003. <http://www.sims.berkeley.edu:8000/research/conferences/p2pecon>.
- Mathieu P., Beauflis B., Delahaye J.-P., “Studies on Dynamics in the Classical Iterated Prisoner’s Dilemma with Few Strategies”, *Artificial Evolution*, 1999, vol. 1829 of *Lecture Notes in Computer Science*, Springer, p. 177-190.
- Morge M., *Humanités numériques*, Traité Information-Commande-Communication, Hermès science publication, chapitre « Se concerter à l’aide d’un système multi-agents », 2007. À paraître.
- Parkes D. C., Shneidman J., « Distributed Implementations of Vickrey-Clarke-Groves Mechanism », *Proc. of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, IEEE Computer Society, 2004, p. 261-268.
- Petcu A., Faltings B., Parkes D. C., “MDPOP : faithful distributed implementation of efficient social choice problems”, *Proc. of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, ACM Press, 2006, p. 1397-1404.
- Rawls J., *A Theory of Justice*, Belknap Press of Harvard University Press, Cambridge, 1971.
- von Neumann J., Morgenstern O., *Theory of Games and Economic Behavior*, Princeton University Press, 1944.

A. Preuves

Démonstration 1 (Mécanisme non optimal) *On peut facilement démontrer qu’une distribution q^* est optimale ssi :*

$$\sum_{i>1} \frac{\delta v_i(q_i^*)}{\delta q} = \frac{\delta c(q)}{\delta q}$$

Si le vecteur d’offre \check{b} donne lieu à un équilibre de Nash alors on peut déduire l’équation du premier ordre suivante :

$$\forall i > 1 \quad \frac{\delta v_i(d(\check{b}))}{\delta q} \leq \frac{\delta c(d(\check{b}))}{\delta q}, \text{ avec } \check{b}_i > 0$$

On déduit des deux équations précédentes que, lorsqu’on considère plus de deux consommateurs, dans une situation d’équilibre de Nash, ce mécanisme n’implémente pas à une allocation optimale.

Démonstration 2 (Mécanisme optimal et stable) *On constate que les fonctions de valuation v_i , considérées comme des fonctions de leur propre offre*

b_i , sont concaves. Le vecteur d'offre \check{b} donne lieu à un équilibre de Nash $\vec{q} = d(\check{b})$ ssi :

$$\forall i > 1 \quad \frac{\delta v_i(d(\check{b}))}{\delta q} \cdot \frac{\delta d(\check{b})}{\delta q} - 1 + \frac{\check{B}_{-i}}{\check{B} + b_i} = 0$$

On peut réécrire cette équation de la manière suivante :

$$\forall i > 0 \quad \frac{\delta v_i(\vec{q})}{\delta q} = \frac{\check{b}_i}{\check{B}} \cdot \frac{\delta c(\vec{q})}{\delta q}$$

Ce qui équivaut à \vec{q} est optimale.

Démonstration 3 (Stratégie convergente) Considérons \vec{q}^* une distribution optimale. La somme des offres correspondantes est définie telle que :

$$B^* = \operatorname{argmax}_B \sum_{i>1} [v_i(d(B))] - B$$

Etant donnée la stratégie d'offre ascendante proposée, on obtient l'équation suivante :

$$\frac{\delta B}{\delta |R|} = \sum_{i>1} \frac{\delta}{\delta |R|} v_i(d(B)) \cdot \frac{\delta}{\delta |R|} d(B) - 1$$

La dérivée de la somme des offres est le résultat des dérivées individuelles pour chacun des agents qui converge vers un résultat optimal.