

Partage équitable de biens indivisibles sous risque

Charles Lumet*
charles.lumet@onera.fr

Sylvain Bouveret*
sylvain.bouveret@onera.fr

Michel Lemaître*
lmtr@club.fr

* Onera Centre de Toulouse
2, avenue Édouard Belin
31055 Toulouse Cedex 4 – FRANCE

Résumé :

Cet article concerne le problème de partage équitable d'objets indivisibles entre des agents ayant des préférences additives sur ces objets. Nous nous intéressons au cas où chaque objet peut être dans deux états possibles : normal ou dégradé. Nous supposons de plus que l'état réel des objets n'est pas connu au moment du partage, mais que le décideur connaît la probabilité pour chaque objet d'être normal ou dégradé. Nous proposons un modèle formel de ce problème s'appuyant sur les notions d'équité *ex-ante* et *ex-post*, et nous proposons des algorithmes de calcul de partages optimaux au sens de l'égalitarisme *ex-post* que nous évaluons sur des jeux d'instances aléatoires.

Mots-clés : Partage équitable, risque, algorithmique, *timing effect*

Abstract:

We study the problem of fairly allocating a set of indivisible goods to a set of agents having additive preferences. More precisely, we consider the problem in which each object can be in two possible states : good or bad. We further assume that the actual object state is not known at the allocation time, but that the decision-maker knows the probability for each object to be in each state. We propose a formal model of this problem, based on the notions of *ex-ante* and *ex-post* fairness, and we propose some algorithms aiming at computing optimal allocations in the sense of *ex-post* egalitarianism, the efficiency of these algorithms being tested on random instances.

Keywords: Fair share, risk, algorithmics, timing effect

1 Introduction

Dans nombre d'applications réelles, plusieurs objets indivisibles doivent être partagés entre plusieurs agents (individus, entités,...), d'une manière équitable, c'est-à-dire en tenant compte des préférences souvent antagonistes des agents. Un exemple de problème de partage réel, cité entre autres dans [11], concerne la gestion d'une constellation de satellites d'observation de la Terre. Dans cette application, plusieurs agents, représentant les pays ayant cofinancé la constellation, se répartissent les créneaux d'utilisation des satellites en communiquant chaque jour au centre de contrôle des demandes d'observation assorties d'un poids reflétant leur importance. Les demandes ne pouvant pas être toutes réalisées en raison de conflits entre elles, le centre de

contrôle est chargé de sélectionner celles qui seront satisfaites. Cette sélection doit tenir compte des importances des demandes et doit faire en sorte de léser le moins possible les agents (en d'autres termes, elle doit satisfaire, entre autres, des contraintes d'équité). Ce problème se modélise naturellement comme un problème de partage équitable d'objets indivisibles.

Si le partage de biens indivisibles est étudié depuis longtemps en microéconomie, on observe depuis quelques années un intérêt croissant pour les problèmes de décision collective de la part la communauté informatique, particulièrement dans le domaine de l'intelligence artificielle et de la recherche opérationnelle. La raison d'être de cette nouvelle communauté du «choix social computationnel», formée à la fois d'économistes et d'informaticiens, est double : d'une part, la microéconomie apporte des réponses aux problèmes de décision collective qui apparaissent naturellement en informatique (dans les systèmes multiagents par exemple), et d'autre part, l'étude de la décision collective soulève de nombreuses questions d'ordre calculatoire, auxquelles les théoriciens de l'informatique peuvent apporter des réponses : représentation compacte de préférences, complexité, aspects algorithmiques.

Même si la majorité des travaux sur les aspects computationnels des problèmes de partage d'objets indivisibles est issue du domaine des enchères combinatoires [7] – dans lequel on ne se préoccupe que rarement d'équité entre les agents – on commence à trouver quelques références sur la complexité [13, 3], la représentation compacte [6, 2] et les aspects algorithmiques [16, 1, 4, 12] liés au partage équitable.

Dans certains problèmes de partage équitable, la valeur des objets pour les agents peut être incertaine au moment où le partage est décidé. Dans l'exemple du partage de la constellation de satellites, les conditions météorologiques sur la zone où l'observation doit avoir lieu sont incertaines¹ et peuvent dégrader l'utilité de l'ob-

1. Plus précisément, elles sont risquées, ce qui signifie que l'on

servation pour l'agent qui la demande (s'il y a des nuages sur la zone, par exemple).

Alors que l'incertitude (ou le risque) dans les problèmes de décision collective et de partage a été étudié dans le domaine de l'économie [15, 8], ce problème n'a jamais été traité, à notre connaissance, sous ses aspects computationnels (en particulier, aucun des travaux d'intelligence artificielle ou de recherche opérationnelle évoqués ci-dessus n'aborde la question de l'incertitude). Notre travail propose une première approche pour tenter de combler ce vide.

Dans cet article, nous faisons trois hypothèses. (i) Le partage est abordé dans un contexte centralisé, c'est-à-dire qu'il incombe à un arbitre supposé bienveillant de décider à qui revient chaque objet, au regard des préférences des agents. (ii) Les agents ont des préférences additives sur les objets, c'est-à-dire représentées pour chaque agent a par un ensemble de poids w_{ao} , chacun traduisant l'utilité (ou la satisfaction) de l'agent en question pour l'objet o seul. L'utilité de l'agent pour un ensemble d'objets S est simplement donnée par la somme des poids de chacun des objets $o \in S$. (iii) En ce qui concerne le risque, nous nous restreignons au cas où chaque objet o ne peut prendre que deux états – normal ou dégradé – et les variables aléatoires donnant la réalisation de l'état pour chaque objet sont indépendantes et de lois de probabilité connues ; nous supposons enfin qu'un objet dans un état dégradé a une utilité nulle pour l'agent qui le reçoit.

Ce cadre d'étude peut sembler assez restrictif à première vue, mais nous pensons que les problèmes posés sont assez intéressants pour justifier une étude sur le sujet. D'une part, l'hypothèse d'additivité est l'une des plus naturelles dès que l'on souhaite modéliser des préférences numériques sur des ensembles d'objets de manière compacte. D'autre part, dans un grand nombre de problèmes réels (tels que le problème de partage de satellites d'observation cité plus haut) l'incertitude porte sur la qualité des objets elle-même, et peut être modélisée de manière naturelle comme nous le proposons. Enfin, ainsi que nous nous attachons à le montrer dans cet article, notre modèle, malgré son apparente simplicité, pose des problèmes de calcul de partages optimaux non triviaux.

Cet article est organisé de la manière suivante.

connaît la loi de probabilité associée à cet aléa (selon l'acception classique de la notion de risque et d'incertitude [10]).

Dans la section 2 nous introduisons notre modèle du problème de partage de biens indivisibles sous risque, ainsi que quelques propriétés des critères considérés. Dans la section 3 nous nous intéressons à la complexité de ce problème, et nous proposons deux algorithmes de résolution. Enfin, nous comparons l'efficacité de ces algorithmes sur des instances aléatoires dans la section 4.

2 Cadre d'étude

2.1 Modèle

Résoudre un problème de partage de biens indivisibles consiste à déterminer l'attribution d'un ensemble fini d'objets $\mathcal{O} = \{1, \dots, l\}$ à un ensemble fini d'agents $\mathcal{A} = \{1, \dots, n\}$. Une *décision de partage* (ou simplement *partage*, ou *allocation*) décrit une manière d'effectuer cette attribution ; on appelle *allocation complète* une décision qui attribue tous les objets. On se place dans le cadre classique où le partage est soumis à une contrainte de préemption, stipulant qu'un même objet ne peut être attribué qu'à un seul agent. Un partage peut alors être modélisé par une décision $\vec{d} = (d_1, \dots, d_l) \in (\mathcal{A} \cup \{na\})^l$, où d_o indique à quel agent l'objet o est attribué, ou na si l'objet n'a pas été attribué. L'ensemble des décisions possibles $(\mathcal{A} \cup \{na\})^l$ est noté \mathcal{D} . On appelle *part de l'agent a* l'ensemble $\pi_a(\vec{d}) = \{o \in \mathcal{O} \mid d_o = a\}$ des objets que la décision \vec{d} lui attribue.

Chaque objet se trouve dans l'un des deux états suivants : l'état *normal* ou l'état *dégradé*. L'état d'un objet n'est connu qu'après l'opération de partage, mais le décideur dispose tout de même d'une information probabiliste : à chaque objet $o \in \mathcal{O}$, est attachée une variable aléatoire binaire X_o pouvant prendre les valeurs *normal* ou *dégradé*. On suppose que l'on dispose d'un vecteur $\vec{p} \in [0; 1]^l$ donnant pour chaque objet o la probabilité $p_o = \mathbb{P}(X_o = \textit{normal})$, et $\bar{p}_o = 1 - p_o = \mathbb{P}(X_o = \textit{dégradé})$. Les variables X_o , $o \in \mathcal{O}$ sont supposées indépendantes les unes des autres.

Chaque état de la nature du problème est donc caractérisé par l'ensemble $O \in 2^{\mathcal{O}}$ des objets qui sont dans l'état normal (les autres étant dans l'état dégradé). On notera $\mathcal{S} = \{1, \dots, k\}$ l'ensemble des *états de la nature* (où $k = 2^l$), isomorphe à l'ensemble $2^{\mathcal{O}}$: à chaque état s de

cet ensemble est associé l'ensemble $F(s) \subseteq \mathcal{O}$ des objets dans l'état normal dans cet état. Par la suite, afin de simplifier les notations, on assimilera s à $F(s)$. L'ensemble des états de la nature est muni d'une distribution de probabilités, complètement caractérisée par les coefficients p_o :

$$\forall s \in \mathcal{S}, \Pr(s) = \prod_{o \in s} p_o \prod_{o \notin s} \bar{p}_o \quad (1)$$

Une résolution satisfaisante du problème nécessite de connaître les goûts des agents vis-à-vis des objets. Ces *préférences* sont exprimées numériquement par les agents sous forme de *fonctions d'utilités*, qui, pour chaque état s , associe à chaque décision \vec{d} une valeur numérique $u_{a,s}(\vec{d})$ traduisant l'attrait de l'agent pour cette décision si cet état se réalise. Cette utilité est construite à partir de l'attribution par chaque agent de « poids » à chaque objet ; le poids noté w_{ao} représente ainsi l'intensité de la préférence de l'agent a pour l'objet o ; on fait alors l'hypothèse que l'utilité d'un agent vis-à-vis d'une décision et d'un état de la nature s'exprime comme la somme des poids des objets *dans l'état normal* reçus par cet agent. En d'autres termes, les agents ont des préférences *additives* sur les objets, et tout objet dans un état dégradé apporte une utilité nulle à l'agent qui le reçoit.

$$\forall a \in \mathcal{A}, \forall s \in \mathcal{S}, u_{a,s}(\vec{d}) = \sum_{\substack{o \in s \\ o \in \pi_a(\vec{d})}} w_{ao}$$

Par la suite, on notera $W = [w_{a,o}] \in \mathcal{M}_{n,l}(\mathbb{R}^+)^2$ la matrice de poids donnés par les agents aux objets. On appellera *profil d'utilité* d'une décision $\vec{d} \in \mathcal{D}$ la matrice $U(\vec{d}) = [u_{a,s}(\vec{d})] \in \mathcal{M}_{n,k}(\mathbb{R}^+)$.

Nous sommes maintenant en mesure de définir une instance des problèmes de partage étudiés dans cet article.

Définition 1 (Instance d'un problème de partage). *Une instance d'un problème de partage est un tuple $(\mathcal{A}, \mathcal{O}, \vec{p}, W)$, où :*

- $\mathcal{A} = \{1, \dots, n\}$ est un ensemble d'agents ;
- $\mathcal{O} = \{1, \dots, l\}$ est un ensemble d'objets ;

2. On désigne par $\mathcal{M}_{m,p}(\mathbb{K})$ l'ensemble des matrices d'éléments de \mathbb{K} à m lignes et p colonnes.

- $\vec{p} \in [0; 1]^l$ exprime les probabilités pour chaque objet d'être dans l'état normal ;
- $W \in \mathcal{M}_{n,k}(\mathbb{R}^+)$ regroupe les poids attribués par les agents aux objets.

Exemple 1. *Voici l'exemple d'une instance d'un problème de partage définie comme suit :*

- Agents : $\mathcal{A} = \{1, 2\}$.
- Objets : $\mathcal{O} = \{1, 2, 3, 4\}$.
- Probabilités : $\vec{p} = \langle 0.8, 0.5, 0.5, 0.2 \rangle$
- Poids :

$$W = \begin{bmatrix} 10 & 2 & 4 & 7 \\ 3 & 8 & 4 & 10 \end{bmatrix}$$

La table 1 montre les probabilités associées à chaque état de la nature possible, dans la ligne 2. Le profil d'utilité associé à la décision qui attribue les objets 2 et 3 à l'agent 1 et les objets 1 et 4 à l'agent 2 apparaît dans les lignes 3 et 4.

2.2 Le timing effect

Choisir une décision à partir des préférences des agents nécessite de définir quelles propriétés l'on souhaite voir cette décision vérifier. En présence de préférences numériques, on cherche à produire une fonction, ou *critère de décision*, qui permette d'agrèger les préférences des agents en une valeur numérique unique pour chaque décision, traduisant la qualité de celle-ci au vu des propriétés précédemment choisies.

Pour un état du monde fixé, la qualité d'un partage dépend du niveau de satisfaction de tous les agents. La manière classique de déterminer cette qualité est d'*agrèger* le vecteur d'utilité des agents grâce à une *fonction d'utilité collective* commutative et croissante $\mathfrak{M} : (\mathbb{R}^+)^n \rightarrow \mathbb{R}^+$, qui mesure le bien-être social. La commutativité implique que les agents sont considérés comme étant égaux vis-à-vis du partage. La croissance garantit qu'il existe au moins une décision Pareto-optimale parmi celles qui maximisent \mathfrak{M} .

Le panorama des fonctions d'agrégation pertinentes pour \mathfrak{M} est étendu. Deux choix classiques sont les choix de $\mathfrak{M} = +$ et de $\mathfrak{M} = \min$. $\mathfrak{M} = \min$ est un choix d'équité (ou d'égalitarisme), dans le sens où les meilleurs partages sont ceux pour lesquels l'agent le moins satisfait est le plus heureux possible, alors que le cas de $\mathfrak{M} = +$ privilégie l'efficacité (ou l'utilitarisme), et tend à distribuer les objets aux agents qui en font le meilleur usage en terme d'utilité. D'autres choix intermédiaires sont possibles,

| | | | | | | | | | |
|-------------------------|-------------|---------|---------|---------|---------|------------|------------|------------|------------|
| s | \emptyset | $\{1\}$ | $\{2\}$ | $\{3\}$ | $\{4\}$ | $\{1, 2\}$ | $\{1, 3\}$ | $\{1, 4\}$ | $\{2, 3\}$ |
| p | 0.04 | 0.16 | 0.04 | 0.04 | 0.01 | 0.16 | 0.16 | 0.04 | 0.04 |
| u_1 | 0 | 0 | 2 | 4 | 0 | 2 | 4 | 0 | 6 |
| u_2 | 0 | 3 | 0 | 0 | 10 | 3 | 3 | 13 | 0 |
| $\mathfrak{M}(\vec{u})$ | 0 | 0 | 0 | 0 | 0 | 2 | 3 | 0 | 0 |

| | | | | | | | | |
|-------------------------|------------|------------|---------------|---------------|---------------|---------------|------------------|-------------------------|
| s | $\{2, 4\}$ | $\{3, 4\}$ | $\{1, 2, 3\}$ | $\{1, 2, 4\}$ | $\{1, 3, 4\}$ | $\{2, 3, 4\}$ | $\{1, 2, 3, 4\}$ | $\mathfrak{E}(\vec{u})$ |
| p | 0.01 | 0.01 | 0.16 | 0.04 | 0.04 | 0.01 | 0.04 | — |
| u_1 | 2 | 4 | 6 | 2 | 4 | 6 | 6 | 3 |
| u_2 | 10 | 10 | 3 | 13 | 13 | 10 | 13 | 4.4 |
| $\mathfrak{M}(\vec{u})$ | 2 | 4 | 3 | 2 | 4 | 6 | 6 | 3 1.88 |

TABLE 1 – Profil d'utilité pour le partage $\vec{d} = (2, 1, 1, 2)$ tel que $\pi(1) = \{2, 3\}$ and $\pi(2) = \{1, 4\}$, et calcul des utilités *ex-ante* et *ex-post* pour $\mathfrak{M} = \min$ et $\mathfrak{E} = \text{E}$.

notamment le critère de Nash, les moyennes généralisées ou les moyennes pondérées ordonnées. On pourra consulter l'ouvrage de référence de Moulin [14] pour un panorama complet de ces fonctions d'utilité collective.

De même, on sait, pour un agent fixé, agréger ses utilités dans les différents états du monde en une unique « utilité espérée ». On utilise une fonction d'agrégation stochastique $\mathfrak{E} : (\mathbb{R}^+ \times [0, 1])^k \rightarrow \mathbb{R}^+$. Nous supposons dans la suite de cet article que les agents sont rationnels au sens de [18] et nous faisons par conséquent le choix classique de l'espérance mathématique : $\mathfrak{E} = \text{E}$.

On utilise pour ces fonctions la notation « opérateur » et on écrira ainsi $\mathfrak{M}(\vec{u}) = \mathfrak{M}_{a \in \mathcal{A}} u_a$ et $\mathfrak{E}((u_1, \lambda_1), \dots, (u_k, \lambda_k)) = \mathfrak{E}_{s \in \mathcal{S}} \langle u_s, \lambda_s \rangle$.

L'utilisation conjointe des deux opérateurs d'agrégation collective et stochastique permet de construire deux fonctions différentes, chacune attribuant une valeur numérique à toute décision : selon que l'on agrège d'abord avec \mathfrak{E} puis \mathfrak{M} ou l'inverse, on parlera d'agrégation *ex-ante* ou *ex-post* [15] :

La fonction $uca : \mathcal{D} \rightarrow \mathbb{R}^+$, définie par $\forall \vec{d} \in \mathcal{D}$,

$$uca(\vec{d}) = \mathfrak{M}_{a \in \mathcal{A}} \mathfrak{E}_{s \in \mathcal{S}} \langle u_{a,s}(\vec{d}), \text{Pr}(s) \rangle \quad (2)$$

est appelée fonction d'utilité collective *ex-ante*.

La fonction $ucp : \mathcal{D} \rightarrow \mathbb{R}^+$, définie par $\forall \vec{d} \in \mathcal{D}$,

$$ucp(\vec{d}) = \mathfrak{E}_{s \in \mathcal{S}} \langle \mathfrak{M}_{a \in \mathcal{A}} u_{a,s}(\vec{d}), \text{Pr}(s) \rangle \quad (3)$$

est appelée fonction d'utilité collective *ex-post*.

Remarque 1. *En l'absence de risque, c'est-à-dire si pour tout objet $o \in \mathcal{O}$, $p_o = 1$, les fonctions d'utilité collective *ex-post* et *ex-ante* sont égales.*

Harsanyi [9] montre que les seules fonctions d'agrégation collective conduisant à une égalité des utilités *ex-post* et *ex-ante* sont linéaires ou affines, ce qui implique qu'à l'inverse, toute fonction d'agrégation collective favorisant l'équité conduira à des utilités *ex-post* et *ex-ante* différentes. Il y a donc un conflit – on parle de *timing effect* – entre l'approche *ex-post* d'une part, qui s'intéresse à l'espérance du bien-être social, et l'approche *ex-ante* de l'autre, qui s'intéresse au bien-être social mesuré à partir des utilités espérées.

Exemple 1 (suite). *La table 1 illustre le calcul des utilités *ex-ante* et *ex-post* pour une décision particulière. Le calcul de l'utilité *ex-ante* se fait en agrégeant les utilités ligne par ligne, grâce à l'espérance mathématique (les valeurs apparaissent dans la dernière colonne), puis en agrégeant ces valeurs grâce à la fonction d'agrégation collective (le résultat apparaît dans la partie supérieure droite de la case inférieure droite du tableau). Le calcul de l'utilité *ex-post* se fait en agrégeant les utilités colonne par colonne, grâce à la fonction d'agrégation collective (les valeurs apparaissent dans la dernière ligne), puis en agrégeant ces valeurs grâce à la fonction d'agrégation stochastique (le résultat apparaît dans la partie inférieure gauche de la case inférieure droite du tableau).*

2.3 Comparaison des utilités *ex-post* et *ex-ante*

Même s'il n'existe *a priori* aucun lien entre l'utilité *ex-post* et l'utilité *ex-ante* pour une décision \vec{d} donnée, on peut montrer que, sous certaines hypothèses raisonnables sur la fonction d'agrégation collective, l'utilité collective *ex-ante* est toujours inférieure ou égale à l'utilité collective *ex-post*.

Définition 2 (Séparabilité). *On dit qu'un opérateur d'agrégation collective \mathfrak{M} est séparable si et seulement s'il existe une fonction $g : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ croissante telle que pour tout vecteur d'utilité $\vec{u} \in (\mathbb{R}^+)^n$, $\mathfrak{M}(\vec{u}) = \sum_{a=1}^n g(u_a)$.*

Cette hypothèse n'est pas très restrictive en pratique, dans la mesure où la plupart des fonctions d'utilité collective usuelles (à l'exception notable des moyennes pondérées ordonnées) sont séparables ou équivalentes (au sens de l'ordre engendré sur les vecteurs d'utilité) à des fonctions d'utilité collectives séparables.

La propriété suivante caractérise les opérateurs d'agrégation collective qui favorisent l'équité de la solution, en s'intéressant à leur comportement face à un « déplacement » d'utilité depuis les agents les mieux dotés vers ceux les moins bien dotés. Une telle redistribution d'utilité est appelée *transfert de Pigou-Dalton* et constitue le fondement du *principe de réduction des inégalités* (voir en particulier [14]).

Définition 3 (Transfert de Pigou-Dalton).

Soient \vec{u} et \vec{u}' deux vecteurs d'utilité. \vec{u}' est obtenu par transfert de Pigou-Dalton à partir de \vec{u} si et seulement si on peut trouver un couple d'agents distincts (a_1, a_2) , $a_1 \neq a_2$ dans \mathcal{A}^2 tels que :

- la somme de leurs utilités est conservée : $u_{a_1} + u_{a_2} = u'_{a_1} + u'_{a_2}$
- les inégalités sont réduites : $u_{a_1} < \{u'_{a_1}, u'_{a_2}\} < u_{a_2}$
- les utilités des autres agents restent inchangées : $\forall a \in \mathcal{A} \setminus \{a_1, a_2\}, u_a = u'_a$

On dit qu'une fonction d'agrégation collective \mathfrak{M} réduit les inégalités si pour tout \vec{u}, \vec{u}' , on a $\mathfrak{M}(\vec{u}') \geq \mathfrak{M}(\vec{u})$ dès lors que \vec{u}' est obtenu par transfert de Pigou-Dalton à partir de \vec{u} .

La comparaison des utilités *ex-post* et *ex-ante* conduit à la proposition 1.

Proposition 1. *Soit \mathfrak{M} un opérateur d'agrégation collective séparable et qui réduit les inégalités. Alors, si l'opérateur d'agrégation stochastique choisi est l'espérance mathématique ($\mathfrak{E} = \bar{\mathfrak{E}}$), on a l'inégalité :*

$$\forall d \in \mathcal{D}, \text{ucp}(\vec{d}) \leq \text{uca}(\vec{d}) \quad (4)$$

Démonstration. La séparabilité de \mathfrak{M} donne l'existence d'une fonction $g : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ croissante telle que pour tout $\vec{u} \in (\mathbb{R}^+)^n$, $\mathfrak{M}(\vec{u}) = \sum_{a=1}^n g(u_a)$. \mathfrak{M} réduit les inégalités, d'où la concavité de g (voir [14]). D'après l'inégalité de Jensen, on a $g(\sum_{s=1}^k \text{Pr}(s)u_{a,s}) \geq \sum_{s=1}^k \text{Pr}(s)g(u_{a,s})$, ce qui nous donne directement le résultat. \square

Notons que la fonction d'agrégation collective égalitariste $\mathfrak{M} = \min$ n'est pas séparable, et donc n'entre pas dans le cadre de la proposition 1³. Cependant, l'inégalité $\min_{a \in \mathcal{A}} \sum_{s \in \mathcal{S}} \text{Pr}(s)u_{a,s}(\vec{d}) \leq \sum_{s \in \mathcal{S}} \text{Pr}(s) \min_{a \in \mathcal{A}} u_{a,s}(\vec{d})$ est une application immédiate de l'inégalité triangulaire sur la fonction \min . La proposition 1 est donc vraie également pour $\mathfrak{M} = \min$.

3 Complexité et algorithmique

On se place pour toute la suite dans le cadre de l'égalitarisme classique, autrement dit $\mathfrak{M} = \min$. On rappelle qu'on suppose également $\mathfrak{E} = \bar{\mathfrak{E}}$.

3.1 Complexité

Nous nous intéressons dans cette section aux problèmes d'optimisation des utilités collectives *ex-post* et *ex-ante* de la définition 4.

Définition 4. *Le problème d'optimisation de l'utilité collective *ex-ante* (respectivement *ex-post*) s'exprime comme suit :*

« Étant donné une instance d'un problème de partage (définition 1) et un nombre $K \in \mathbb{R}$, existe-t-il une décision $\vec{d} \in \mathcal{D}$ telle que $\text{uca}(\vec{d}) \geq K$ (respectivement telle que $\text{ucp}(\vec{d}) \geq K$) ? »

3. Même si un raffinement de la fonction \min , l'ordre de bien-être collectif *leximin*, peut être représenté comme une fonction d'utilité collective séparable si l'espace des utilités est fini.

Considérons d'abord le problème d'optimisation *ex-ante*. L'utilité collective *ex-ante* est définie par l'équation 2 ; en considérant des « poids espérés » $\tilde{w}_{ao} = p_o w_{ao}$, l'expression se simplifie comme suit :

$$\begin{aligned} \forall \vec{d} \in \mathcal{D}, uca(\vec{d}) &= \min_{a \in \mathcal{A}} \sum_{o \in \pi_a(\vec{d})} \tilde{w}_{ao} \\ &= \mathfrak{M}_{a \in \mathcal{A}} u_a(\vec{d}) \end{aligned}$$

où $u_a(\vec{d}) = \sum_{o \in \pi_a(\vec{d})} \tilde{w}_{ao}$. Le simple calcul des \tilde{w}_{ao} , qui s'effectue en temps linéaire en la taille du problème (i.e. $O(nl)$) permet ainsi de ramener le problème à un problème d'optimisation similaire sans risque, dans lequel les poids affectés par les agents aux objets sont les poids espérés du problème initial.

Le problème d'optimisation *ex-post*, contrairement au problème d'optimisation *ex-ante*, ne peut pas être simplement ramené à un problème de partage sans risque, car l'application des opérateurs d'agrégation se fait dans l'ordre inverse. De manière intéressante, le simple calcul de l'utilité *ex-post* semble à lui seul très complexe, car la structure combinatoire de l'espace d'états impose la réalisation une agrégation stochastique sur un nombre exponentiel d'états.

On peut montrer assez facilement que le problème de calcul de l'utilité *ex-post* d'une décision peut se ramener à un problème de comptage, ce qui suggérerait le fait que ce problème de calcul est dans la classe $\#P$ [17]. Cela impliquerait donc que le problème d'optimisation de l'utilité *ex-post* est dans la classe $NP^{\#P}$ des problèmes pouvant être résolus en un temps polynomial par une machine de Turing non déterministe faisant appel à un oracle pour un problème de $\#P$. La question de savoir si ces problèmes sont complets pour leurs classes respectives reste ouverte...

3.2 Algorithmique

On a vu dans la section précédente que l'optimisation de l'utilité collective *ex-ante* se ramène à l'optimisation de l'utilité collective d'un problème de partage sans risque, qui a déjà été étudiée (voir par exemple les travaux cités en introduction). On se consacre donc exclusivement dans cette partie à l'algorithmique associée à l'optimisation de l'utilité collective *ex-post*.

Quelle que soit l'approche algorithmique adoptée pour la résolution du problème, il est indis-

pensable de pouvoir calculer l'utilité collective *ex-post* d'une décision donnée pour une instance donnée. L'application directe de la formule (3) conduit à un algorithme énumérant l'ensemble des états de la nature, donc exponentiel en le nombre d'objets.

Cependant, pour certains états de la nature, la connaissance de l'état d'un sous-ensemble d'objets suffit à calculer entièrement l'utilité collective dans cet état de la nature. En effet, dès que tous les objets attribués à un agent sont dans l'état dégradé, l'utilité de cet agent est nulle, et donc l'utilité collective l'est également, indépendamment de l'état des autres objets.

Il convient donc de construire les états de la nature en déterminant l'état des objets les uns après les autres, en vérifiant à chaque étape qu'aucun agent ne se trouve dans la situation précédente ; si tel est le cas, il est alors inutile d'énumérer les états de la nature associés aux états des objets restants à définir, puisque l'on sait que l'utilité collective dans de tels états de la nature sera toujours nulle.

Enfin, il est judicieux de spécifier l'ordre dans lequel les états des objets sont déterminés, de façon à traiter en premier les objets appartenant à des parts de cardinal faible (i.e. à des agents recevant peu d'objets), et ainsi éliminer le plus vite possible les états de la nature dont l'énumération est superflue.

Les remarques précédentes conduisent à l'algorithme 1. On utilise dans cet algorithme la fonction TRI définie de la manière suivante : $TRI(\vec{u}, f)$ renvoie un vecteur \vec{u}^\uparrow formé par les composantes de \vec{u} permutées, et tel que $i < j \Rightarrow f(u_i^\uparrow) \leq f(u_j^\uparrow)$.

Revenons maintenant au problème d'optimisation. Nous développons deux approches algorithmiques de ce problème : un algorithme de recherche exacte de type *branch and bound* d'une part, et d'autre part un algorithme de recherche approchée de type glouton stochastique.

La recherche exacte utilise un algorithme de *branch and bound* classique où les variables d_o sont associées aux objets, et peuvent prendre leur valeur dans $\mathcal{A} \cup \{na\}$. L'efficacité d'un algorithme de ce type dépend en grande partie de sa capacité à détecter rapidement les affectations de piètre qualité afin de « couper » des parts significatives de l'arbre de recherche. Une coupe repose sur l'utilisation d'une fonction

Algorithme 1 : Fonction EXPOST : calcul de l'utilité collective *ex-post*

Données : Une allocation complète \vec{d}

Résultat : L'utilité collective *ex-post* $ucp(\vec{d})$

```

1  $\vec{\pi}^\uparrow \leftarrow \text{TRI}(\langle \pi_1(\vec{d}), \dots, \pi_n(\vec{d}) \rangle, \mathcal{X} \mapsto |\mathcal{X}|)$ 
2 return PARCOURS( $\langle 0, \dots, 0 \rangle, 1, \vec{\pi}^\uparrow, 1$ )

```

Fonction PARCOURS($\vec{u}, pr, \langle \pi_1, \dots, \pi_n \rangle, a$)

Données : Un vecteur d'utilités \vec{u} , un réel $pr \in [0; 1]$, un vecteur de parts d'objets $\vec{\pi}$, un agent a

Résultat : L'utilité collective *ex-post* $ucp(\vec{d})$

```

1 si  $\pi_a = \emptyset$  alors
2   si  $a = n$  alors
3     return  $\min(\vec{u}) \times pr$ 
4   sinon
5     si  $u_a = 0$  alors return 0
6     return PARCOURS( $\vec{u}, pr, \vec{\pi}, a + 1$ )
7 sinon
8    $o \leftarrow$  objet arbitraire de  $\pi_a$ 
9    $\vec{\pi}' \leftarrow \langle \dots, \pi_{a-1}, \pi_a \setminus \{o\}, \pi_{a+1}, \dots \rangle$ 
10   $\vec{u}' \leftarrow \langle \dots, u_{a-1}, u_a + w_{ao}, u_{a+1}, \dots \rangle$ 
11  return PARCOURS( $\vec{u}, pr \cdot \bar{p}_o, \vec{\pi}', a$ ) +
    PARCOURS( $\vec{u}', pr \cdot p_o, \vec{\pi}', a$ )

```

majorante de la grandeur à optimiser et algorithmiquement simple à calculer ; cette seconde condition élimine immédiatement toute fonction construite à partir de l'utilité *ex-post* : en effet le calcul de l'utilité *ex-post* est long et cette grandeur n'est pas « maintenable » au cours de l'algorithme, c'est-à-dire qu'il n'est pas envisageable lors de l'affectation d'une nouvelle variable, de réutiliser des calculs déjà faits ; le calcul d'utilité *ex-post* n'a donc lieu qu'une fois une allocation complète réalisée pour en mesurer la qualité, mais n'est pas utilisé comme stratégie de coupe.

À la place, nous tirons parti de l'inégalité (4) en choisissant pour majorant la fonction \overline{uca} qui s'exprime comme l'utilité *ex-ante* d'une décision virtuelle \vec{d}^\uparrow qui attribuerait à *tous les agents* les objets non encore attribués par la décision courante \vec{d} :

$$\begin{aligned} \overline{uca}(\vec{d}) &= uca(\vec{d}^\uparrow) \\ &= \min_{a \in \mathcal{A}} \left(\sum_{o \in \pi_a(\vec{d})} \tilde{w}_{ao} + \sum_{\substack{o \in \mathcal{O} \\ d_o = na}} \tilde{w}_{ao} \right) \end{aligned}$$

La croissance de la fonction \min donne immédiatement $\overline{uca}(\vec{d}) \geq uca(\vec{d})$ pour toute décision \vec{d} de \mathcal{D} , et l'inégalité (4) montre que \overline{uca} est bien un majorant de l'utilité collective *ex-post*. Si \overline{uca} majore grossièrement ucp , elle se révèle toutefois rapide à calculer, sa mise à jour après l'affectation de chaque nouvel objet étant très simple.

On dispose donc d'une part d'un majorant grossier et rapide à calculer, et d'autre part de la fonction à optimiser, dont le calcul est très long. Il nous a semblé intéressant de chercher une fonction intermédiaire, qui constituerait un meilleur majorant qu' \overline{uca} et dont le calcul serait plus rapide que celui d' ucp . Il faudrait en quelque sorte calculer l'utilité collective de façon *ex-post* pour un sous-ensemble \mathcal{O}_{ep} des objets et *ex-ante* pour ceux restants ($\mathcal{O}_{ea} = \mathcal{O} \setminus \mathcal{O}_{ep}$); nous définissons dans ce sens l'*utilité mixte*, notée $um_{a,s}^{\mathcal{O}_{ep}}$ pour un agent a et un état de la nature s donnés.

$$um_{a,s}^{\mathcal{O}_{ep}}(\vec{d}) = \sum_{\substack{o \in \mathcal{O}_{ep} \cap s \\ o \in \pi_a(\vec{d})}} w_{ao} + \sum_{\substack{o \notin \mathcal{O}_{ep} \\ o \in \pi_a(\vec{d})}} \tilde{w}_{ao}$$

L'utilité mixte correspond à l'utilité d'un agent qui considère que les objets de \mathcal{O}_{ea} sont à coup sûr en bon état ($p_o = 1$) et qui leur attribue des poids \tilde{w}_{ao} . L'*utilité collective mixte* est définie par l'équation (5) comme l'utilité collective *ex-post* issue des utilités mixtes individuelles.

$$ucm^{\mathcal{O}_{ep}}(\vec{d}) = \mathbb{E}_{s \in \mathcal{S}} \left\langle \min_{a \in \mathcal{A}} um_{a,s}^{\mathcal{O}_{ep}}(\vec{d}), \text{Pr}(s) \right\rangle \quad (5)$$

On remarque que les utilités mixtes individuelles sont indépendantes de l'état des objets qui n'appartiennent pas à \mathcal{O}_{ep} . Le calcul de l'espérance mathématique dans l'équation 5 peut donc se simplifier, conduisant à la formule 6, où pour $s \subseteq \mathcal{O}_{ep}$, on note $\text{Pr}_{\mathcal{O}_{ep}}(s) =$

$\sum_{t \subseteq \mathcal{O}_{ea}} \Pr(t \cup s) = \prod_{o \in s} p_o \prod_{o \in \mathcal{O}_{ep} \setminus s} \bar{p}_o$ la probabilité que les objets de \mathcal{O}_{ep} soient dans l'état spécifié par s , indépendamment de l'état des autres objets. Le nombre d'états du monde à parcourir pour le calcul est sensiblement réduit (il est divisé par 2 pour chaque objet n'appartenant pas à \mathcal{O}_{ep}), d'où l'intérêt algorithmique de l'utilité collective mixte.

$$ucm_{\mathcal{O}_{ep}}(\vec{d}) = \sum_{s \subseteq \mathcal{O}_{ep}} \Pr(s) \min_{a \in \mathcal{A}} um_{a,s}^{\mathcal{O}_{ep}}(\vec{d}) \quad (6)$$

On montre que l'utilité collective mixte est comprise entre l'utilité collective *ex-post* et *ex-ante*.

Proposition 2 (Utilité collective mixte). *Pour toute décision $\vec{d} \in \mathcal{D}$ et pour tout sous-ensemble $\mathcal{O}_{ep} \subseteq \mathcal{O}$, on a :*

$$uca(\vec{d}) \geq ucm^{\mathcal{O}_{ep}}(\vec{d}) \geq ucp(\vec{d}) \quad (7)$$

Démonstration. (1) Montrons que $ucm^{\mathcal{O}_{ep}}(\vec{d}) \leq uca(\vec{d})$. D'après l'équation 6 :

$$\begin{aligned} & ucm^{\mathcal{O}_{ep}}(\vec{d}) \\ &= \sum_{s \subseteq \mathcal{O}_{ep}} \Pr(s) \min_{a \in \mathcal{A}} \left(\sum_{\substack{o \in s \\ o \in \pi_a(\vec{d})}} w_{ao} + \sum_{\substack{o \notin \mathcal{O}_{ep} \\ o \in \pi_a(\vec{d})}} \tilde{w}_{ao} \right) \end{aligned}$$

D'après l'inégalité triangulaire :

$$\begin{aligned} & ucm^{\mathcal{O}_{ep}}(\vec{d}) \\ &\leq \min_{a \in \mathcal{A}} \sum_{s \subseteq \mathcal{O}_{ep}} \Pr(s) \left(\sum_{\substack{o \in s \\ o \in \pi_a(\vec{d})}} w_{ao} + \sum_{\substack{o \notin \mathcal{O}_{ep} \\ o \in \pi_a(\vec{d})}} \tilde{w}_{ao} \right) \\ &= \min_{a \in \mathcal{A}} \left(\sum_{s \subseteq \mathcal{O}_{ep}} \Pr(s) \sum_{\substack{o \in s \\ o \in \pi_a(\vec{d})}} w_{ao} \right) + \sum_{\substack{o \notin \mathcal{O}_{ep} \\ o \in \pi_a(\vec{d})}} \tilde{w}_{ao} \\ &= \min_{a \in \mathcal{A}} \left(\sum_{\substack{o \in \mathcal{O}_{ep} \\ o \in \pi_a(\vec{d})}} p_o w_{ao} + \sum_{\substack{o \notin \mathcal{O}_{ep} \\ o \in \pi_a(\vec{d})}} \tilde{w}_{ao} \right) = uca(\vec{d}) \end{aligned}$$

(2) Montrons que $ucm^{\mathcal{O}_{ep}}(\vec{d}) \geq ucp(\vec{d})$. L'utilité mixte individuelle s'écrit, pour $a \in \mathcal{A}$,

et $s \in \mathcal{O}_{ep}$,

$$\begin{aligned} um_{a,s}^{\mathcal{O}_{ep}}(\vec{d}) &= \sum_{\substack{o \in s \\ o \in \pi_a(\vec{d})}} w_{ao} + \sum_{\substack{o \in \mathcal{O}_{ea} \\ o \in \pi_a(\vec{d})}} p_o w_{ao} \\ &= \sum_{t \subseteq \mathcal{O}_{ea}} \Pr(t) \left(\sum_{\substack{o \in s \\ o \in \pi_a(\vec{d})}} w_{ao} + \sum_{o \in t} w_{ao} \right) \end{aligned}$$

L'utilité collective mixte s'écrit alors :

$$\begin{aligned} & ucm^{\mathcal{O}_{ep}}(\vec{d}) \\ &= \sum_{s \subseteq \mathcal{O}_{ep}} \Pr(s) \min_{a \in \mathcal{A}} \left(\sum_{t \subseteq \mathcal{O}_{ea}} \Pr(t) \sum_{\substack{o \in t \text{ ou} \\ o \in s \cap \pi_a(\vec{d})}} w_{ao} \right) \end{aligned}$$

D'après l'inégalité triangulaire :

$$\begin{aligned} & ucm^{\mathcal{O}_{ep}}(\vec{d}) \\ &\geq \sum_{\substack{s \in \mathcal{O}_{ep} \\ t \in \mathcal{O}_{ea}}} \Pr(s) \Pr(t) \cdot \min_{a \in \mathcal{A}} \sum_{\substack{o \in s \cup t \\ o \in \pi_a(\vec{d})}} w_{ao} \\ &= ucp(\vec{d}) \end{aligned}$$

□

Notre algorithme de *branch and bound* utilise la fonction majorante \overline{uca} pour effectuer les coupes dans tout le corps de la recherche ; la fonction ucm est quant à elle mise à profit quand une allocation complète a été réalisée : elle ne permet donc pas de couper des parties de l'arbre de recherche, mais simplement d'éviter le calcul d'une utilité collective *ex-post* quand celle-ci n'est pas nécessaire.

Deux heuristiques dynamiques inspirées de [4] améliorent l'efficacité de l'algorithme : tout objet va d'abord être attribué à l'agent le plus « pauvre » (c'est-à-dire celui dont l'utilité espérée est actuellement la plus faible) ; quand un nouvel objet doit être affecté, on choisit parmi ceux restants à attribuer celui que l'agent le plus pauvre préfère.

L'algorithme de recherche approchée (algorithme 2) est basé sur un algorithme glouton stochastique [5]. Dès qu'une allocation complète est construite, un calcul approché de l'utilité collective *ex-post* est réalisé par EXPOSTA, de façon à déterminer si cette allocation va être conservée ou non. Le calcul approché est réalisé avec l'utilité collective mixte ou par la méthode de Monte-Carlo (suite de tirages aléatoires dans l'espace des états de la nature). On

conserve au cours de l'algorithme un nombre fixe $nbStockage$ de « bonnes » allocations ; si une allocation est meilleure, d'après le calcul approché, que la pire allocation actuellement stockée, la fonction `CONSERVER` sauvegarde cette nouvelle allocation (et l'autre est supprimée si la limite de stockage est atteinte). Dès que $nbStockage \times nbAvantCalculExact$ allocations ont été réalisées, le calcul exact de l'utilité *ex-post* des allocations stockées intervient, et on ne garde alors que la meilleure d'entre elles, toutes les autres étant supprimées.

Lors de la construction d'une allocation, le choix des objets et des agents est réalisé en suivant la même heuristique que pour l'algorithme de *branch and bound*, aléatoirement biaisée. Les données du problème intervenant dans les heuristiques (poids w_{ao} et utilités *ex-ante*) sont temporairement altérées pendant le choix d'un objet et d'un l'agent auquel l'attribuer par la fonction $alea : \mathbb{R} \rightarrow \mathbb{R}$, définie par l'équation 8, où ϕ est un paramètre réel positif et X une variable aléatoire de loi normale centrée réduite.

$$\forall y \in \mathbb{R}, alea(y) = y \cdot (1 + \phi X) \quad (8)$$

4 Résultats

Les algorithmes proposés dans cet article ont été implémentés en Java sur des instances générées aléatoirement. A la création d'une instance, les poids w_{ao} sont tirés uniformément dans $\{0, 1, \dots, 99\}$, et les probabilités p_o uniformément dans $[0; 1]$.

La table 2 et la figure 1 montrent les résultats obtenus avec l'algorithme de recherche exacte. L'algorithme est testé en générant pour un nombre d'objets et d'agents fixés, 100 instances aléatoires. Quatre configurations sont testées : l'algorithme est d'abord exécuté avec une coupe basée simplement sur la fonction \overline{uca} (cas (a)), puis en utilisant les heuristiques dynamiques (cas (b)), ensuite en mettant à profit le calcul intelligent de l'utilité collective *ex-post* (cas (c)), et enfin en réalisant des coupes à partir de l'utilité collective mixte (cas (d)). La figure 1 montre l'efficacité de la configuration (d), où l'utilisation de l'utilité collective mixte permet de réaliser de bonnes coupes, et ainsi réduire le nombre de calculs d'utilité collective *ex-post* effectués au cours de l'algorithme.

L'algorithme de calcul exact nous permet en particulier d'évaluer l'influence du *timing effect*

Algorithme 2 : Glouton stochastique

Données : Une instance d'un problème de partage

Résultat : Un bon partage selon l'utilité collective *ex-post*

```

1 Stock ← ∅
2  $\vec{d}^* = (d_1^*, \dots, d_l^*) \leftarrow (na, \dots, na)$ 
3  $ucp^* \leftarrow 0$ 
4  $i \leftarrow 0$ 
5 tant que le temps imparti n'est pas écoulé faire
6    $\vec{d} \leftarrow \text{CONSTRUIREALLOCATION}()$ 
7   si  $uca(\vec{d}) \geq ucp^*$  alors
8      $ucp_{app} \leftarrow \text{EXPOSTA}(\vec{d})$ 
9     si  $ucp_{app} > \min_{\vec{d} \in Stock} (\text{EXPOSTA}(\vec{d}))$ 
10    alors
11       $\text{CONSERVER}(\vec{d})$ 
12     $i \leftarrow i + 1$ 
13    si  $i = nbStockage \times nbAvantCalculExact$ 
14    alors
15      pour  $\vec{d} \in Stock$  faire
16         $ucp \leftarrow \text{EXPOST}(\vec{d})$ 
17        si  $ucp > ucp^*$  alors
18           $\vec{d}^* \leftarrow \vec{d}$ 
19           $ucp^* \leftarrow ucp$ 
20       $Stock \leftarrow \emptyset$ 
21 renvoyer  $\vec{d}^*$ 

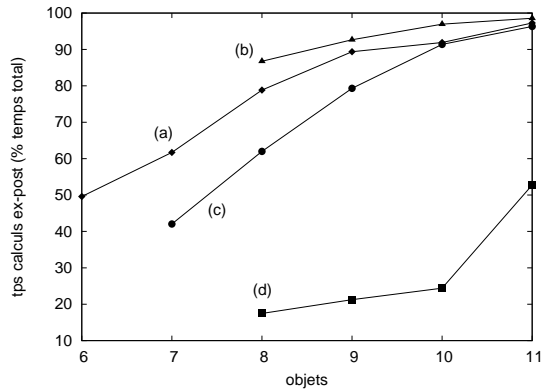
```

Procédure CONSTRUIREALLOCATION()

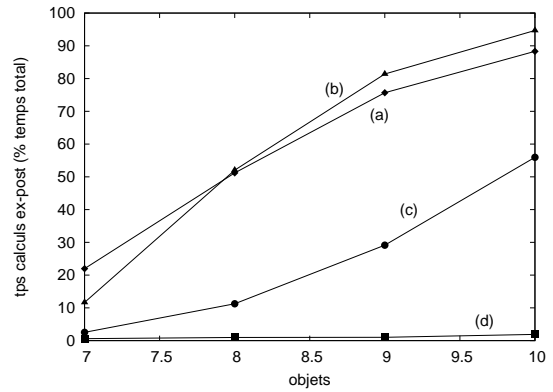
```

1  $\vec{u} = (u_1, \dots, u_n) \leftarrow (0, \dots, 0)$ 
2  $\vec{d} = (d_1, \dots, d_l) \leftarrow (na, \dots, na)$ 
3 tant que  $\exists o \in \mathcal{O}, d_o = na$  faire
4    $\hat{a} \leftarrow \text{argmax}_{a \in \mathcal{A}} (alea(u_a))$ 
5    $\hat{o} \leftarrow \text{argmax}_{o \in \mathcal{O}, d_o = na} (alea(w_{\hat{a}o}))$ 
6    $d_{\hat{o}} \leftarrow \hat{a}$ 
7    $u_{\hat{a}} \leftarrow ua_{\hat{a}} + w_{\hat{a}\hat{o}}$ 
8 renvoyer  $\vec{d}$ 

```



Instances avec 5 agents



Instances avec 7 agents

FIGURE 1 – Résolution exacte. Temps pris par les calculs d'utilité collective *ex-post*, en pourcentage du temps total d'exécution, pour 5 et 7 agents (moyennes sur 100 instances)

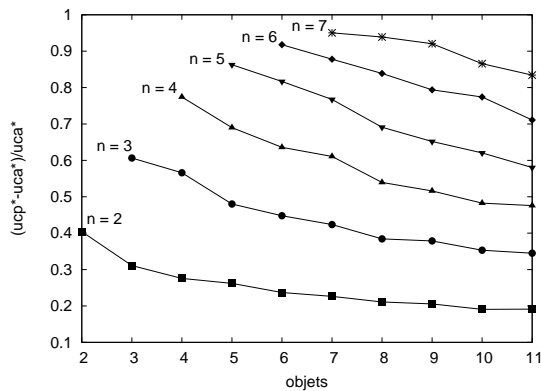


FIGURE 2 – Influence du *timing effect*. Le rapport $(uca^* - ucp^*)/uca^*$ est représenté en fonction du nombre d'objets, pour différents nombres d'agents.

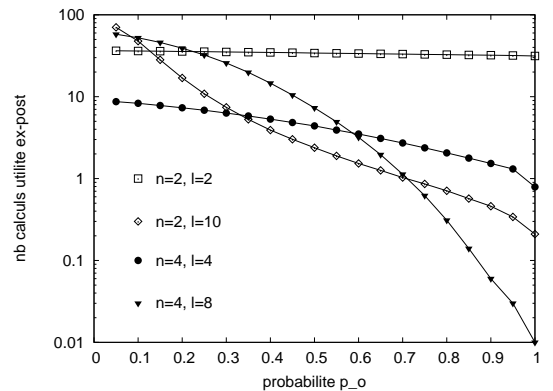
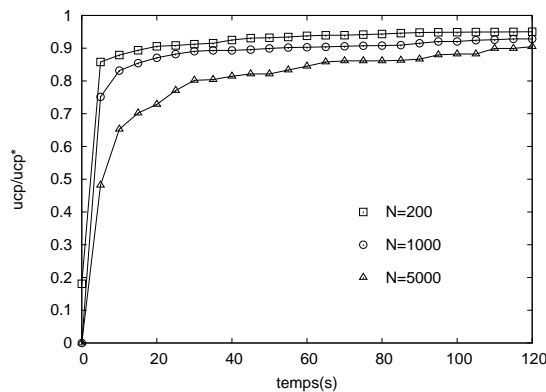
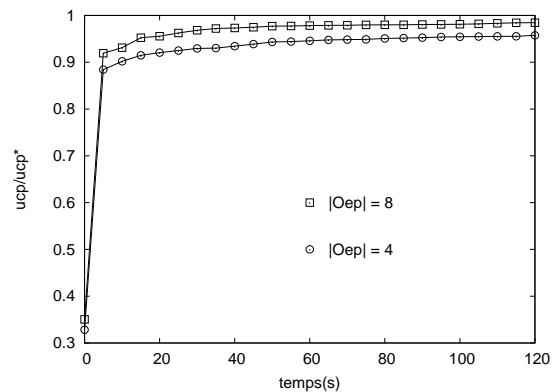


FIGURE 3 – Influence des probabilités. Nombre de branches de l'arbre de recherche totalement parcourues en pourcentage du nombre total de branches (moyennes sur 100 instances pour différents nombres d'agents et d'objets « équiprobables »)



(a) approximation par méthode de Monte Carlo, pour des nombres de tirages différents.



(b) approximation par l'utilité collective mixte, pour différentes tailles de l'ensemble \mathcal{O}_{ep} .

FIGURE 4 – Résolution approchée. Evolution de l'utilité collective *ex-post* de la meilleure solution au cours du temps, pour deux méthodes d'approximation différentes (moyennes sur 100 instances de 5 agents et 12 objets).

| n | l | (a) | (b) | (c) | (d) |
|-----|-----------|-----|-----|-----|-----|
| 5 | ≤ 9 | 100 | 100 | 100 | 100 |
| 5 | 10 | 49 | 52 | 89 | 100 |
| 5 | 11 | 1 | 1 | 10 | 52 |
| 5 | ≥ 12 | 0 | 0 | 0 | 0 |

| n | l | (a) | (b) | (c) | (d) |
|-----|-----------|-----|-----|-----|-----|
| 7 | ≤ 8 | 100 | 100 | 100 | 100 |
| 7 | 9 | 27 | 47 | 100 | 100 |
| 7 | 10 | 0 | 1 | 19 | 32 |
| 7 | ≥ 11 | 0 | 0 | 0 | 0 |

TABLE 2 – Résolution exacte. Nombre d’instances résolues en 30 secondes (sur 100 instances)

selon le nombre d’agents et le nombre d’objets des instances considérées. Les données recueillies sont résumées sur le graphique de la figure 2. L’écart relatif entre les utilités collectives *ex-post* et *ex-ante* augmente quand le nombre d’agents augmente ou que le nombre d’objets diminue.

D’autre part, on constate empiriquement que l’efficacité de l’algorithme est grandement dépendante de la valeur des probabilités \vec{p} . Cette corrélation est très clairement illustrée par la figure 3 ; on s’y intéresse au nombre de branches de l’arbre de recherche totalement explorées (i.e. le nombre de calculs d’utilité collective *ex-post* réalisés) lors de l’exécution de l’algorithme de *branch and bound* sur des instances dont les probabilités p_o sont identiques pour tous les objets. Les instances dans lesquelles les objets ont de bonnes chances d’être dans l’état normal sont résolues plus rapidement que les autres (nombre réduit de branches totalement explorées). Cela s’explique par la plus grande proximité entre les valeurs des utilités collectives *ex-ante* et *ex-post* quand les probabilités p_o sont proches de 1, et donc une plus grande efficacité des fonctions utilisées pour les coupes. Nous avons vu que le cas où $\vec{p} = (1, \dots, 1)$ correspond à un problème de partage sans risque, et que par conséquent dans un tel cas $uca = ucp$.

L’algorithme 2 de calcul approché est testé sur 100 instances comportant chacune 5 agents et 12 objets, sur une durée de 2 minutes. La figure 4 retrace l’exécution de l’algorithme qui utilise l’utilité collective mixte (respectivement la méthode de Monte-Carlo) pour approximer l’utilité collective *ex-post*, avec différentes tailles de l’ensemble \mathcal{O}_{ep} (respectivement différents nombres N de tirages). \mathcal{O}_{ep} est construit avec les objets qui ont le plus de chance d’être en bon état. La méthode de Monte-Carlo se révèle moins efficace que l’utilisation de l’utilité collective mixte. La figure 5 montre l’influence des valeurs $nbStockage$ et $nbAvantCalculExact$ sur l’algorithme : la qualité de la solution fournie augmente significativement avec le nombre de solutions stockées au cours de l’exécution.

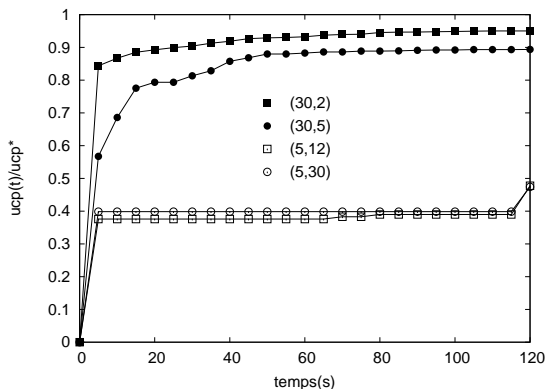


FIGURE 5 – Résolution approchée. Evolution de l’utilité collective *ex-post* pour différents couples $(nbStockage, nbAvantCalculExact)$ (moyennes sur 100 instances de 5 agents et 12 objets).

5 Conclusion

Dans cet article, nous avons introduit un modèle simple de problème de partage équitable de biens indivisibles en présence de risque. Nous avons vu que le *timing effect* impose de considérer à la fois les utilités collectives *ex-ante* et *ex-post* quand on s’intéresse à un problème de partage en présence de risque. Nous avons de plus montré que, sous certaines hypothèses raisonnables, le problème d’optimisation de l’utilité *ex-ante* pouvait se ramener à des problèmes de partage sans risque, mais que le problème d’optimisation *ex-post* semblait beaucoup plus complexe. Nous avons proposé la notion d’utilité collective mixte comme base pour construire un algorithme exact ainsi qu’un algorithme de recherche approchée pour la résolution des problèmes d’optimisation *ex-post*.

Il reste encore beaucoup de travail à accomplir sur les problèmes de partage équitable sous risque. Le premier travail est bien entendu de caractériser exactement la complexité du problème de calcul et d’optimisation de l’utilité *ex-post*. On pourra envisager ensuite d’étendre le modèle, par exemple pour qu’il prenne en compte des dépendances préférentielles et/ou

probabilistes entre les objets, ou encore une notion de risque qui ne se limite pas à des états binaires attachés aux objets.

Remerciements Nous tenons à remercier Jérôme Lang pour ses nombreux commentaires au cours de la genèse de cet article.

Références

- [1] Arash Asadpour and Amin Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. In *Proc. of STOC-2007*, 2007.
- [2] Sylvain Bouveret, Ulle Endriss, and Jérôme Lang. Conditional importance networks : A graphical language for representing ordinal, monotonic preferences over sets of goods. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*, pages 67–72, Pasadena, California, July 2009.
- [3] Sylvain Bouveret and Jérôme Lang. Efficiency and envy-freeness in fair division of indivisible goods : Logical representation and complexity. *Journal of Artificial Intelligence Research (JAIR)*, 32 :525–564, 2008.
- [4] Sylvain Bouveret and Michel Lemaître. Computing leximin-optimal solutions in constraint networks. *Artificial Intelligence*, 173(2) :343–364, 2009.
- [5] John L. Bresina. Heuristic-Biased Stochastic Sampling. In *Proceedings of the 13th AAAI Conference on Artificial Intelligence (AAAI-96)*, pages 271–278, Portland, OR, 1996.
- [6] Yann Chevaleyre, Ulle Endriss, Sylvia Estivie, and Nicolas Maudet. Multiagent resource allocation with k -additive utility functions. In *Proc. DIMACS-LAMSADE Workshop on Computer Science and Decision Theory*, volume 3 of *Annales du LAMSADE*, pages 83–100, 2004.
- [7] Peter Cramton, Yoav Shoham, and Richard Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2006.
- [8] Thibault Gajdos and Éric Maurin. Unequal uncertainties and uncertain inequalities : an axiomatic approach. *Journal of Economic Theory*, 116(1) :93–118, 2004.
- [9] John Charles Harsanyi. Cardinal welfare, individualistic ethics, and interpersonal comparisons of utility. *Journal of political economy*, 63 :309–321, 1955.
- [10] Frank H. Knight. *Risk, Uncertainty and Profit*, 1921. *Boston and New York*, 1921.
- [11] Michel Lemaître, Gérard Verfaillie, and Nicolas Bataille. Exploiting a common property resource under a fairness constraint : a case study. In Thomas Dean, editor, *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 206–211, Stockholm, Sweden, July 1999. Morgan Kaufmann.
- [12] Julien Lesca and Patrice Perny. Lp solvable models for multiagent fair allocation problems. In *European Conference on Artificial Intelligence*, pages 387–392, 2010.
- [13] Richard Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of divisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC-04)*, New York, NY, May 2004. ACM.
- [14] Hervé Moulin. *Axioms of Cooperative Decision Making*. Cambridge University Press, 1988.
- [15] Roger B. Myerson. Utilitarianism, egalitarianism, and the timing effect in social choice problems. *Econometrica*, 49(4) :883–897, 1981.
- [16] Włodzimierz Ogryczak. On the lexicographic minimax approach to location problems. *European Journal of Operational Research*, 100 :566–585, 1997.
- [17] Leslie G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8 :189–201, 1979.
- [18] John Von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ, 1947.