

# Jeux booléens statiques et représentation compacte de préférences (rapport préliminaire)

Elise Bonzon  
bonzon@irit.fr

Marie-Christine Lagasquie-Schiex  
lagasq@irit.fr

Jérôme Lang  
lang@irit.fr

IRIT-UPS  
118 route de Narbonne  
31062 Toulouse Cedex, FRANCE

## Résumé :

Dans cet article, nous proposons une extension des jeux booléens étudiés par [7, 5] qui devrait permettre à terme de modéliser de manière compacte les préférences des joueurs dans les jeux statiques.

Cette extension propose des jeux booléens à  $n$  joueurs et à somme non nulle, et en étudie les propriétés computationnelles. Nous montrons ensuite brièvement comment étendre ces jeux booléens de façon à exprimer des utilités non binaires.

**Mots-clés :** logique propositionnelle, théorie des jeux, complexité algorithmique, représentation compacte de préférences

## Abstract:

In that paper, we propose an extension of the boolean games studied by [7, 5] so as to allow for a compact representation of the player's preferences in static games. This extension allows for expressing  $n$ -player, non zero-sum Boolean games, of which we study some computational properties. We briefly show how to extend these so as to express non-binary utilities.

**Keywords:** propositional logic, game theory, computational complexity, compact preference representation

## 1 Introduction

La théorie des jeux est probablement le modèle formel le plus abouti pour l'étude des interactions stratégiques entre agents. Informellement, un jeu consiste en un ensemble d'agents (ou joueurs), et pour chaque agent, la donnée d'un ensemble de stratégies possibles et une fonction d'utilité associant une valeur réelle à chaque combinaison possible de stratégies, et, pour les jeux dynamiques à information incomplète, des hypothèses sur les croyances de l'agent au cours du jeu (que nous ne considérerons pas ici).

Pour simplifier un peu la tâche, nous nous plaçons maintenant (et jusqu'à la fin de l'article) dans le cadre des jeux statiques. Que le jeu soit statique signifie que les agents choisissent leur stratégie en parallèle, sans observer quoi que ce soit du choix de l'autre. En outre, pour les jeux statiques les deux modes de représentation usuels des jeux (forme extensive et forme stratégique) coïncident. Cette représentation ne fait

pas l'économie de la description explicite de la fonction d'utilité de chaque agent. Or, cette description est de taille exponentielle en fonction du nombre d'agents : par exemple, si  $n$  agents ont chacun un choix entre deux actions possibles, il faudra spécifier  $n \times 2^n$  valeurs numériques ; si deux agents contrôlent chacun un ensemble de  $p$  variables booléennes (il suffit de penser à de telles variables comme à des boutons que l'agent peut choisir d'enfoncer ou non), chaque agent a  $2^p$  stratégies possibles et il faudra donc expliciter  $2 \times (2^p)^2 = 2^{2p+1}$  valeurs numériques. Cette explosion combinatoire est encore plus flagrante lorsqu'à la fois l'ensemble des agents et l'ensemble des stratégies pour chacun des agents sont de taille importante. Il devient alors déraisonnable de spécifier les fonctions d'utilité de manière explicite, en listant les valeurs pour chaque combinaison de stratégies. Il est tout aussi déraisonnable de penser pouvoir calculer des propriétés du jeu en appliquant un algorithme nécessitant une énumération explicite des combinaisons de stratégies. Pensons par exemple au calcul des équilibres de Nash en stratégies pures (dont nous rappelons la définition plus loin) : ce calcul nécessite, dans le cas des jeux précédents, et dans le pire des cas, un temps de calcul de l'ordre de  $n \times 2^n$  (pour le jeu à  $n$  joueurs avec 2 actions chacun) et de  $2 \times 2^p \times 2^{2p} = 2^{3p+1}$  (pour le jeu à deux joueurs contrôlant chacun  $p$  variables booléennes).

D'un autre côté, une sous-branche de l'intelligence artificielle s'intéresse aux langages de représentation compacte de préférences (ordinales ou numériques). Ces langages permettent une représentation concise de relations de préférences, ou de fonctions d'utilité, sur un ensemble de conséquences qui possède une structure combinatoire (c'est-à-dire un produit cartésien de domaines de valeurs finis pour un ensemble fini de variables), en exploitant dans une large mesure des propriétés structurelles des relations de préférences (comme l'indépendance préférentielle entre variables). En particulier, lorsque les variables en jeu sont binaires, ces

langages sont fondés sur la logique propositionnelle, dont ils héritent l'expressivité et les méthodes algorithmiques (pour la déduction et la recherche de modèles, notamment). L'expressivité et le pouvoir de concision des langages de représentation logique de préférences sont étudiés dans [3] et leur complexité algorithmique dans [8].

Partant de là, puisque la spécification d'un jeu statique nécessite la description des préférences des agents, il apparaît naturel de représenter de tels jeux en utilisant des langages de représentation compacte de préférences. C'est l'objectif premier de cet article, qui donne des résultats très préliminaires et pour l'instant simples, mais qui par ailleurs pose quelques questions plus complexes.

Il existe déjà un cadre répondant (très) partiellement aux problèmes que nous avons posés plus haut : il s'agit des *jeux booléens* [7, 5]. Un jeu booléen est un jeu à deux joueurs et à somme nulle, la fonction d'utilité du joueur 1 (et donc celle du joueur 2 qui est son opposé) est représentée par une formule de la logique propositionnelle, appelée *forme booléenne* du jeu. Après avoir donné en section 3 une description (simplifiée) des jeux booléens, nous montrerons en section 4 que ces jeux booléens peuvent facilement être généralisés de manière à représenter des jeux avec un nombre arbitraire de joueurs et à somme non nulle, mais en gardant l'hypothèse que les préférences de chaque joueur sont représentées par une formule propositionnelle unique, ce qui ne permet de représenter que des utilités binaires. Nous verrons comment des outils simples issus de la logique propositionnelle permettent de caractériser certaines propriétés du jeu. Quelques résultats de complexité algorithmique seront donnés en section 5. En section 6, nous montrerons brièvement comment introduire dans les jeux booléens généralisés des préférences (ordinales ou numériques) non binaires décrites dans un langage de représentation logique. Puis nous poserons en section 7 plusieurs pistes de recherche.

## 2 Notations et rappels de logique propositionnelle

Soit  $V = \{a, b, \dots\}$  un ensemble fini de variables propositionnelles et  $L_V$  le langage propositionnel construit à partir de  $V$ , des connecteurs habituels et des constantes booléennes  $\top$  (vrai) et  $\perp$  (faux). Les formules de  $L_V$  seront notées  $\phi, \psi$

etc.

Un littéral est, soit une variable de  $V$ , soit sa négation. Une conjonction finie de littéraux est appelée *terme*. On note  $Lit(\alpha)$  l'ensemble des littéraux formant le terme  $\alpha$ .

$2^V$  est l'ensemble des interprétations pour  $V$  avec la convention suivante : soit  $M$  une interprétation pour  $V$  et pour tout  $x \in V$ ,  $M$  donne la valeur *vrai* à  $x$  si  $x \in M$  et *faux* sinon. Soit  $M$  une interprétation pour  $V$  et  $\psi \in L_V$ , la conséquence logique  $M \models \psi$  est définie de la manière usuelle.

Nous utiliserons aussi la notion d'interprétation partielle de  $L_V$ . Une *interprétation partielle* de  $L_V$  est une fonction  $PM : V \rightarrow \{\text{vrai}, \text{faux}, \text{inconnu}\}$ , où pour tout  $x \in V$ ,  $PM(x) = \text{vrai}$  (resp. *faux*, *inconnu*) signifie que  $PM$  donne à  $x$  la valeur *vrai* (resp. *faux*, resp. ne donne pas de valeur à  $x$ ). L'ensemble des interprétations totales étendant une interprétation partielle  $PM$  se note  $Ext(PM)$  et se définit par :  $Ext(PM) = \{M \in 2^V \mid \forall x \in V, PM(x) = \text{vrai} \rightarrow x \in M\}$ . La notion de conséquence logique est alors la suivante : Si  $PM$  est une interprétation partielle de  $L_V$  et  $\psi \in L_V$  alors  $PM \models \psi$  ssi  $\forall M \in Ext(PM), M \models \psi$ .

Nous aurons également besoin dans cet article de plusieurs notions d'impliquants premiers. Les définitions suivantes sont reprises de l'article de synthèse [10].

Intuitivement, un impliquant premier d'une formule propositionnelle  $\psi$  est un des plus petits termes dont tous les modèles sont des modèles de  $\psi$ .

### Définition 1 (Impliquant, impliquant premier)

Soit  $\psi$  une formule propositionnelle.

- Un terme  $\alpha$  est un impliquant de  $\psi$  ssi  $\alpha \models \psi$ .
- Un terme  $\alpha$  est un impliquant premier de  $\psi$  ssi
  - $\alpha$  est un impliquant de  $\psi$ , et
  - pour chaque impliquant  $\alpha'$  de  $\psi$ , si  $\alpha \models \alpha'$ , alors  $\alpha' \models \alpha$ .

On notera  $PI(\psi)$  l'ensemble des impliquants premiers de  $\psi$ .

Un  $L$ -impliquant premier est un impliquant premier dont tous les littéraux appartiennent à l'ensemble  $L$ .

### Définition 2 (L-impliquant, L-impliquant premier)

Soit  $L \subseteq V$  et soit  $\psi$  une formule propositionnelle de  $L_V$ .

- Un terme  $\alpha$  est un L-impliquant de  $\psi$  ssi  $\alpha \models \psi$  et  $\text{Lit}(\alpha) \subseteq L$ .
  - Un terme  $\alpha$  est un L-impliquant premier de  $\psi$  ssi
    - $\alpha$  est un L-impliquant de  $\psi$ , et
    - pour chaque L-impliquant  $\alpha'$  de  $\psi$ , si  $\alpha \models \alpha'$ , alors  $\alpha' \models \alpha$ .
- On notera  $PI_L(\psi)$  l'ensemble des L-impliquants premiers de  $\psi$ .

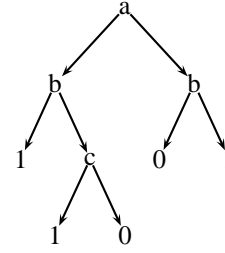


FIG. 1 – Jeu booléen  $(a, (b(\mathbf{1}, c(\mathbf{1}, \mathbf{0})), b(\mathbf{0}, \mathbf{1})))^3$

### 3 Introduction aux jeux booléens

Un jeu booléen sur  $V$  [7, 5] est un jeu à deux joueurs  $a_1$  et  $a_2$ , à somme nulle, ayant les spécificités suivantes :

- les actions que peuvent entreprendre les deux joueurs consistent à donner une valeur de vérité à des variables de  $V$  ;
- les fonctions d'utilité des deux joueurs sont représentées au moyen d'une formule propositionnelle  $\phi$  formée sur les variables  $V$ , appelée *forme booléenne* du jeu.

$\phi$  représente le but du joueur 1 : l'utilité de celui-ci est +1 lorsque  $\phi$  est satisfaite<sup>1</sup> (et alors le joueur 1 gagne), et -1 sinon (et c'est alors le joueur 2 qui gagne). Le jeu étant à somme nulle, c'est-à-dire  $u_2 = -u_1$ , le but du joueur 2 n'est autre que  $\neg\phi$ . Le jeu n'a donc que deux issues possibles : le gain de  $a_1$  ou celui de  $a_2$ .

Pour construire ces jeux booléens, [7, 5] commencent par définir deux jeux booléens atomiques, dénotés par  $\mathbf{1}$  et  $\mathbf{0}$ . Le premier est gagné par  $a_1$  sans qu'aucun des joueurs n'ait à prendre la moindre décision, tandis que le second est gagné par  $a_2$ . Des jeux booléens plus complexes sont ensuite construits récursivement à partir de ces jeux atomiques et d'un ensemble de variables propositionnelles que l'on appellera variables de décision binaires. Pour tous jeux booléens  $g_0$  et  $g_1$ , et pour toute variable de décision  $a$ , il existe un autre jeu booléen dénoté  $a(g_0, g_1)$ . Chaque variable de décision est contrôlée de manière exclusive par l'un des deux joueurs.

**Exemple 1** Soit  $V = \{a, b, c\}$  un ensemble de variables propositionnelles. Soit  $a_1, a_2$  deux joueurs ayant pour buts :  $\phi_1 = (a \leftrightarrow b) \vee (\neg a \wedge b \wedge \neg c)$ ,  $\phi_2 = \neg\phi_1 = (\neg a \wedge b \wedge c) \vee (a \wedge \neg b)$ . Le joueur  $a_1$  contrôle les variables  $a$  et  $c$ , tandis que  $a_2$  contrôle la variable  $b$ .

La représentation proposée par Harrenstein dans [5] de ce jeu booléen est donnée figure 1.

<sup>1</sup>Une formule booléenne est satisfaite ssi la formule est vraie.

<sup>3</sup>La flèche gauche partant du nœud  $x$  représente la mise à faux de  $x$ , tandis que la flèche droite représente la mise à vrai de  $x$ .

Comme l'ont constaté Dunne et Van der Hoek [4], cette construction basée sur un modèle dynamique n'est pas nécessaire. En effet, l'hypothèse disant que les stratégies des agents sont choisies en parallèle (c'est-à-dire sans que l'un observe la décision de l'autre) est implicite. Cette forme extensive, sous forme d'arbre, est donc inutile. Dans la suite, on va proposer une représentation des jeux booléens plus simple correspondant à un jeu statique, tout en étendant le cadre des jeux booléens classiques.

### 4 Jeux booléens à $n$ joueurs

Avant de revenir plus en détail aux jeux booléens tels qu'ils ont été définis dans [7, 5], nous allons d'abord les généraliser en nous intéressant à des jeux à  $n$  joueurs et à somme non nulle. Nous verrons ensuite que le cadre étudié par [7, 5] est un cas particulier de ce cadre plus général.

#### 4.1 Définitions

Commençons par formaliser la notion de jeu booléen.

**Définition 3 (Jeu booléen)** Soit un ensemble  $V$  de variables propositionnelles (appelées désormais variables de décision),  $A = \{a_1, a_2, \dots, a_n\}$  un ensemble de joueurs, une fonction d'assignement de contrôle  $\pi : A \rightarrow V$ , et  $\Phi = \{\phi_1, \dots, \phi_n\}$  l'ensemble des formules représentant les buts des joueurs, où chaque  $\phi_i$  est une formule de  $L_V$ . Le jeu booléen correspondant est alors représenté par le quadruplet  $(A, V, \pi, \Phi)$ .

$\pi$  représentant la fonction d'assignement de contrôle qui associe à chaque joueur les variables qu'il contrôle, on note  $\pi_i$  l'ensemble des variables contrôlées par le joueur  $a_i$ . Ainsi,  $\{\pi_1, \dots, \pi_n\}$  forme une partition de  $V$ .

Après avoir défini les jeux booléens, nous pouvons introduire la notion de stratégie pour un joueur.

**Définition 4 (Stratégie)** Soit  $G = (A, V, \pi, \Phi)$  un jeu booléen. Une stratégie  $s_i$  pour un joueur  $a_i$  dans  $G$  est une fonction qui associe à chaque variable contrôlée par le joueur  $a_i$  la valeur de vérité vrai ou faux.

Par abus de langage on considérera que  $s_i = \{x | x \in \pi_i \text{ et } x \text{ mis à vrai}\}$ . On a alors :

$$\forall i \in \{1, \dots, n\}, s_i \in 2^{\pi_i}$$

Cela signifie donc que  $\forall x \in \pi_i \setminus s_i$ ,  $x$  est mis à faux pour la stratégie  $s_i$ .

**Définition 5 (Profil de stratégie)** Un profil de stratégie  $S$  correspond à l'instanciation de toutes les variables de décision du jeu  $(A, V, \pi, \Phi)$  :

$$S = (s_1, s_2, \dots, s_n)$$

Par abus de langage, et sachant que chaque  $s_i$  est un ensemble de variables, on considère  $S$  comme étant l'union des  $s_i$ . On a donc  $S \in 2^V$ .  $S$  représente l'ensemble des variables mises à vrai (toutes celles qui ne seront pas dans  $S$  seront considérées comme étant mises à faux).  $S$  est donc une interprétation pour  $V$  et chaque  $s_i$  est une interprétation partielle de  $L_V$ .

*Notations* : Un ensemble de profils de stratégie sera désigné par  $\Omega$ .

Reprenons l'exemple 1 pour illustrer la notion de stratégie.

**Exemple 1 – suite** Soit  $G = (A, V, \pi, \Phi)$  un jeu booléen, avec

- $V = \{a, b, c\}$ ,  $A = \{a_1, a_2\}$ ,
- $\varphi_1 = (a \leftrightarrow b) \vee (\neg a \wedge b \wedge \neg c)$ ,
- $\varphi_2 = \neg \varphi_1 = (\neg a \wedge b \wedge c) \vee (a \wedge \neg b)$ ,
- $\pi_1 = \{a, c\}$  et  $\pi_2 = \{b\}$ .

Le joueur  $a_1$  a 4 stratégies possibles :  $s_{11} = \{a, c\}$ ,  $s_{12} = \{a\}$ ,  $s_{13} = \{c\}$  ou  $s_{14} = \emptyset$ .

Le joueur  $a_2$  a deux stratégies possibles :  $s_{21} = \{b\}$  ou  $s_{22} = \emptyset$ .

$G$  possède donc 8 profils de stratégie :  $S_1 = (s_{11}, s_{21})$ ,  $S_2 = (s_{11}, s_{22})$ ,  $S_3 = (s_{12}, s_{21})$ ,  $S_4 = (s_{12}, s_{22})$ ,  $S_5 = (s_{13}, s_{21})$ ,  $S_6 = (s_{13}, s_{22})$ ,  $S_7 = (s_{14}, s_{21})$  et  $S_8 = (s_{14}, s_{22})$ .

On a donc  $\Omega = (S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8)$ .

Les profils de stratégie  $S_1, S_3, S_6, S_7$  et  $S_8$  donnent la victoire à  $a_1$ , tandis que  $S_2, S_4$  et  $S_5$  permettent à  $a_2$  de gagner.

*Notations* : Soit  $G = (A, V, \pi, \Phi)$  un jeu booléen, avec  $A = \{a_1, \dots, a_n\}$ . Soit  $S = (s_1, \dots, s_n)$  et  $S' = (s'_1, \dots, s'_n)$  deux profils de stratégie.

On note  $s_{-i}$  le profil de stratégie  $S$  privé de la stratégie du joueur  $a_i$  :

$$s_{-i} = (s_1, s_2, \dots, s_{i-1}, s_{i+1}, \dots, s_n)$$

On note  $(s_{-i}, s'_i)$  le profil de stratégie  $S$  dans lequel on a remplacé la stratégie du joueur  $a_i$  par celle du profil  $S'$  :

$$(s_{-i}, s'_i) = (s_1, s_2, \dots, s_{i-1}, s'_i, s_{i+1}, \dots, s_n)$$

Similairement, on note  $\pi_{-i}$  l'ensemble des variables contrôlées par tous les joueurs sauf le joueur  $a_i$  :

$$\pi_{-i} = V \setminus \pi_i$$

On peut à présent définir la notion de stratégie gagnante.

**Définition 6 (Stratégie gagnante)**

Soit  $G = (A, V, \pi, \Phi)$  un jeu booléen, avec  $\Phi = \{\varphi_1, \dots, \varphi_n\}$ , et  $A = \{a_1, \dots, a_n\}$ . Un profil de stratégie est une stratégie gagnante pour un joueur  $a_i$  si, quels que soient les choix effectués par ses adversaires,  $a_i$  est sûr de gagner en choisissant cette stratégie.

$$\exists s_i \in 2^{\pi_i}, \forall s_{-i} \in 2^{\pi_{-i}} (s_{-i}, s_i) \models \varphi_i$$

Nous allons à présent définir la notion d'utilité afin de caractériser les équilibres de Nash sur les jeux booléens.

**Définition 7 (Utilité)** Une fonction d'utilité est une fonction  $u : \Omega \rightarrow \mathbb{R}$ .  $u(S)$  mesure la satisfaction du joueur lorsque le profil de stratégie  $S$  est choisi.

Nous nous situons ici dans un cadre particulier, nous étudions des jeux à utilité binaire, dont voici la définition :

**Définition 8 (Utilité binaire)** Pour chaque joueur  $i$ , la fonction d'utilité binaire induite par  $\varphi_i$  est la fonction  $u_i : \Omega \rightarrow \mathbb{R}$  telle que :

$$u_i(S) = \begin{cases} 0 & \text{si } S \models \neg \varphi_i \\ 1 & \text{si } S \models \varphi_i \end{cases}$$

Nous pouvons à présent définir les équilibres de Nash en stratégies pures des jeux booléens à  $n$  joueurs.

Un équilibre de Nash est un profil de stratégie tel que la stratégie de chaque joueur est une réponse optimale aux stratégies des autres joueurs.

**Définition 9 (Equilibres de Nash)**

Soit  $G = (A, V, \pi, \Phi)$  un jeu booléen, avec  $\Phi = \{\phi_1, \dots, \phi_n\}$  l'ensemble des buts des joueurs, et  $A = \{a_1, \dots, a_n\}$  l'ensemble des joueurs.

$S = \{s_1, \dots, s_n\}$  est un équilibre de Nash en stratégies pures si et seulement si :

$$\forall i \in \{1, \dots, n\}, \forall s'_i \in 2^{\pi_i}, u_i(S) \geq u_i(s_{-i}, s'_i)$$

Étudions un exemple simple.

**Exemple 2** Soit  $G = \{A, V, \pi, \Phi\}$  un jeu booléen avec

- $V = \{a, b, c\}$ ,  $A = \{a_1, a_2, a_3\}$ ,
- $\pi_1 = \{a\}$ ,  $\pi_2 = \{b\}$ ,  $\pi_3 = \{c\}$ ,
- $\phi_1 = (\neg a \vee (a \wedge b \wedge \neg c))$ ,
- $\phi_2 = (a \leftrightarrow (b \leftrightarrow c))$  et
- $\phi_3 = ((a \wedge \neg b \wedge \neg c) \vee (\neg a \wedge b \wedge c))$ .

On peut à présent construire la forme normale de  $G^4$  :

stratégie de $a_3 : \{c\}$		
$a_1 \backslash a_2$	$\emptyset$	$\{b\}$
$\emptyset$	(1, 1, 0)	(1, 0, 1)
$\{a\}$	(0, 0, 0)	(0, 1, 0)

stratégie de $a_3 : \emptyset$		
$a_1 \backslash a_2$	$\emptyset$	$\{b\}$
$\emptyset$	(1, 0, 0)	(1, 1, 0)
$\{a\}$	(0, 1, 1)	(1, 0, 0)

On constate tout d'abord que le joueur  $a_1$  a une stratégie gagnante. En effet, si  $a_1$  choisit d'instancier la variable  $a$  à faux, alors il est sûr de gagner quels que soient les choix de  $a_2$  et  $a_3$ .

Étudions à présent les équilibres de Nash. Pour cela, il faut étudier chaque profil de stratégie.

<sup>4</sup>Les  $n$ -uplets (ici des triplets) donnent le résultat obtenu par les  $n$  joueurs dans l'ordre : (résultat joueur 1, résultat joueur 2, ...). Le 0 (resp. 1) signifie que le joueur concerné perd (resp. gagne).

- $S_1 = \{a, b, c\} : a_1 \text{ préfère } \{b, c\}$
- $S_2 = \{a, b\} : a_2 \text{ préfère } \{a\}$
- $S_3 = \{a, c\} : a_2 \text{ préfère } \{a, b, c\}$
- $S_4 = \{b, c\} : a_2 \text{ préfère } \{c\}$
- $S_5 = \{a\} : a_1 \text{ préfère } \emptyset$
- $S_6 = \{b\} : a_3 \text{ préfère } \{b, c\}$
- $S_7 = \{c\} : \text{Equilibre de Nash}$
- $S_8 = \emptyset : a_2 \text{ préfère } \{b\}$

Le profil de stratégie  $S = \{c\}$  est donc le seul équilibre de Nash du jeu  $G$ .

L'utilisation des jeux booléens permet d'avoir une représentation compacte du problème. Pour illustrer ce propos, nous allons utiliser une variante de l'exemple du dilemme des prisonniers : nous considérons ici  $n$  prisonniers qui ne peuvent bénéficier que d'un seul type de remise de peine afin de simplifier le problème.

**Exemple 3** Dans le jeu du prisonnier à  $n$  joueurs,  $n$  détenus (notés  $p_i$ ,  $i = 1, \dots, n$ ) sont emprisonnés dans des cellules séparées. La police fait à chacun d'eux le même marché :

"Tu as le choix entre trahir tes complices en les dénonçant (noté  $T_i$ ,  $i = 1, \dots, n$ ) ou les couvrir (noté  $C_i$ ,  $i = 1, \dots, n$ ). Si tu les dénonces, tu auras une remise de peine et tes partenaires purgeront le maximum (sauf si l'un d'eux t'a dénoncé aussi, auquel cas il bénéficiera comme toi d'une remise de peine). Mais si vous vous couvrez mutuellement, vous aurez tous une remise de peine."

La représentation de ce jeu en forme normale pour  $n = 3$  est la suivante :

stratégie de $p_3 : C_3$		
$p_1 \backslash p_2$	$C_2$	$T_2$
$C_1$	(1, 1, 1)	(0, 1, 0)
$T_1$	(1, 0, 0)	(1, 1, 0)

stratégie de $p_3 : T_3$		
$p_1 \backslash p_2$	$C_2$	$T_2$
$C_1$	(0, 0, 1)	(0, 1, 1)
$T_1$	(1, 0, 1)	(1, 1, 1)

On constate ici que pour  $n$  prisonniers, on aura une matrice<sup>5</sup> à  $n$  dimensions, chaque dimension étant égale à 2, donc  $2^n$   $n$ -uplets à spécifier.

<sup>5</sup>Ou un arbre binaire à  $2^n$  feuilles si on utilise une représentation sous forme extensive.

Or, ce jeu peut être traduit très simplement par le jeu booléen  $G = \{A, V, \pi, \Phi\}$  suivant :

- $A = \{p_1, p_2, \dots, p_n\}$ ,
- $V = \{C_1, \dots, C_n\}$  (avec  $\neg C_i = T_i, \forall i$ ),
- $\forall i \in \{1, \dots, n\}, \pi_i = \{C_i\}$ , et
- $\forall i \in \{1, \dots, n\}, \phi_i = \{(C_1 \wedge C_2 \wedge \dots \wedge C_n) \vee \neg C_i\}$ .

L'utilisation des jeux booléens permet donc de réduire de manière très significative la taille de la représentation du problème.

Dans le cas de jeux à  $n$  joueurs, il peut être intéressant de définir la notion de coalitions efficaces.

**Définition 10 (Coalitions efficaces)** Une coalition est un sous-ensemble non-vide  $I$  de l'ensemble des joueurs  $A$ .  $I$  est une coalition efficace si et seulement s'il existe une stratégie combinée de  $I$  satisfaisant les buts de tous les joueurs de  $I$ .  $I$  est une coalition efficace minimale si et seulement si elle est efficace et pour tout  $J \subset I$ ,  $J$  n'est pas efficace.

On peut à présent étudier les propriétés des stratégies gagnantes pour un joueur ou pour une coalition de joueurs.

**Propriété 1** Soit  $G = \{A, V, \pi, \Phi\}$  un jeu booléen.

1.  $\forall a_i \in A, a_i$  a une stratégie gagnante ssi

$$PI_{\pi_i}(\phi_i) \neq \emptyset$$

2. Une coalition de joueurs  $I \subseteq A$  est efficace si et seulement si

$$PI_{\bigcup_{i \in I} \pi_i} \left( \bigwedge_{i \in I} \phi_i \right) \neq \emptyset$$

*Preuve* : Intuitivement, si le but qu'un joueur, ou un ensemble de joueurs noté  $I$ , cherche à satisfaire contient un terme (donc une conjonction de littéraux)  $\alpha$  dont tous les littéraux sont contrôlés par le joueur ou par  $I$ , alors le joueur ou  $I$  ont une stratégie gagnante. En effet, ce terme  $\alpha$  peut être satisfait, donc le but qui le contient est aussi satisfait, et le joueur ou l'ensemble des joueurs  $I$  gagnent.

1. Pour un joueur.

- D'après la définition 2, si  $PI_{\pi_i}(\phi_i) \neq \emptyset$ , alors  $\exists \alpha \in 2^V$  tel que :  $\alpha \models \phi_i$  et  $Lit(\alpha) \subseteq \pi_i$ .  
Donc,  $a_i$  possède une stratégie  $s_i \in 2^{\pi_i}$  telle que :  $\forall s_{-i} \in 2^{\pi_{-i}}(s_{-i}, s_i) \models \phi_i$ .

- D'après la définition 6,  $a_i$  possède une stratégie gagnante si et seulement si  $\exists s_i \in 2^{\pi_i}, \forall s_{-i} \in 2^{\pi_{-i}}(s_{-i}, s_i) \models \phi_i$ .

Le joueur  $a_i$  peut donc satisfaire son but quels que soient les choix de ses adversaires. Il existe donc  $\alpha \in 2^{\pi_i}$  tel que :  $\alpha \models \phi_i$  et  $Lit(\alpha) \subseteq \pi_i$ .  $\alpha$  est un  $\pi_i$ -impliquant de  $\phi_i$ . Donc, il existe  $\alpha'$   $\pi_i$ -impliquant premier de  $\phi_i$ . Donc  $PI_{\pi_i}(\phi_i) \neq \emptyset$ .

2. Pour une coalition de joueurs, la preuve est identique en remplaçant  $\pi_i$  par  $\bigcup_{i \in I} \pi_i$  et  $\phi_i$  par  $\bigwedge_{i \in I} \phi_i$ . ■

Une coalition de joueurs aura donc une stratégie gagnante si et seulement si il existe un impliquant premier de la conjonction de tous les buts des membres de la coalition, cet impliquant premier étant composé uniquement de variables contrôlées par les membres de la coalition.

Étudions ce problème sur un exemple simple.

**Exemple 4** Soit  $G = \{A, V, \pi, \Phi\}$  un jeu booléen. On donne

- $V = \{a, b, c\}, A = \{a_1, a_2, a_3\}$ ,
- $\pi_1 = \{a\}, \pi_2 = \{b\}, \pi_3 = \{c\}$ ,
- $\phi_1 = (a \leftrightarrow (b \wedge c))$ ,
- $\phi_2 = (\neg a \vee \neg c)$  et
- $\phi_3 = (a \wedge \neg b)$ .

On constate tout d'abord qu'aucun joueur isolé n'a de stratégie gagnante.

On remarque ensuite que les trois joueurs ne peuvent pas gagner tous ensemble. En effet,  $\phi_1 \wedge \phi_3$  est incohérent. Comme il ne peut pas exister de coalition entre  $a_1$  et  $a_3$ , les seules coalitions possibles sont  $\{a_1, a_2\}$  et  $\{a_2, a_3\}$ .

Étudions-les.

- $\{a_1, a_2\}$

$$(\phi_1 \wedge \phi_2) = (\neg a \wedge \neg b) \vee (\neg a \wedge \neg c)$$

Il existe un impliquant premier,  $(\neg a \wedge \neg b)$ , qui ne contient que des variables contrôlées par  $a_1$  et  $a_2$ . Cette coalition a donc une stratégie gagnante. Comme c'est la plus petite coalition contenant  $a_1$  et  $a_2$ , c'est une coalition efficace minimale.

- $\{a_2, a_3\}$

$$(\phi_2 \wedge \phi_3) = (a \wedge \neg b \wedge \neg c)$$

Cette coalition n'a pas de stratégie gagnante car la variable  $a$ , qui n'est contrôlée par aucun joueur de cette coalition, appartient au seul impliquant premier. A partir du jeu  $G$ , ce n'est donc pas une coalition efficace.

Remarquons par ailleurs que de façon (presque) équivalente, la recherche d'une stratégie

gagnante correspond à une résolution de  $QBF_{2,\exists}$ [12]. En effet, comme on l'a vu dans la propriété 1, la recherche d'une stratégie gagnante correspond à la recherche d'un impliquant premier. Or, cette recherche est du type  $\exists A \forall B \varphi(A, B)$  : pour un but  $\varphi$  donné, qui peut être le but d'un joueur ou d'une coalition  $I$ , on cherche *s'il existe* un impliquant  $\alpha$  de  $\varphi$  ne contenant que des variables contrôlées par  $I$ . Si  $\alpha$  existe, on veut que *pour tout* autre impliquant,  $\alpha$  soit le "plus petit".

Par ailleurs, l'existence d'une stratégie gagnante est une instance du problème de *contrôlabilité* en logique propositionnelle [1, 9] (voir aussi [2] pour une extension de la contrôlabilité à un cadre multi-agents).

Donnons enfi n une caractérisation simple des équilibres de Nash aux stratégies pures.

**Propriété 2** *Soit  $S \in 2^V$ . Alors  $S$  est un équilibre de Nash en stratégies pures pour  $G$  si et seulement si pour tout  $a_i \in A$ , on a :*

- soit  $S \models \varphi_i$ ,
- soit  $s_{-i} \models \neg\varphi_i$ .

*Preuve :*  $S$  est un équilibre de Nash en stratégies pures pour  $G$  si et seulement si pour tout  $a_i \in A$  et pour tout  $s'_i \in 2^{\pi_i}$  on a  $u_i(S) \geq u_i(s_{-i}, s'_i)$ . Or, puisque  $u_i(S)$  et  $u_i(s_{-i}, s'_i)$  ne peuvent prendre que deux valeurs, cette inégalité est équivalente à

$$\text{pour tout } a_i \in A \text{ et pour tout } s'_i \in 2^{\pi_i}, \\ u_i(S) = 1 \text{ ou } u_i(s_{-i}, s'_i) = 0$$

c'est-à-dire

$$\text{pour tout } a_i \in A, \text{ soit } (a) u_i(S) = 1, \\ \text{soit } (b) \text{ pour tout } s'_i \in 2^{\pi_i}, u_i(s_{-i}, s'_i) = 0$$

(a) est équivalente à  $S \models \varphi_i$  et (b) est équivalente à ( $b'$ ) pour tout  $s'_i \in 2^{\pi_i}$ ,  $(s_{-i}, s'_i) \models \neg\varphi_i$ , donc  $s_{-i} \models \neg\varphi_i$ . ■

## 4.2 Cas particulier : Jeux à deux joueurs et à somme nulle

Les jeux booléens étudiés dans [7, 5] sont un cas particulier des jeux booléens à  $n$  joueurs présentés dans la section 4.1. Les définitions sont les mêmes que celles de la section 4.1, mis à part que la notion de coalition n'a aucun sens.

Certains paramètres sont simplifiés. En effet, comme chaque variable de décision de  $V$  ne peut être contrôlée que par un seul des deux joueurs, on a  $\pi_2 = \overline{\pi_1}$ . D'autre part, on a  $\varphi_2 \equiv \neg\varphi_1$ .

*Stricto sensu, les jeux obtenus ne sont pas à somme nulle, mais à somme constante (égale à 1) – ce qui, en pratique, n'a aucune importance – on utilisera donc, par abus de langage, la terminologie "à somme nulle".*

Étudions à présent un exemple simple.

**Exemple 5** *Soit  $V = \{a, b, c, d\}$ ,  $A = \{a_1, a_2\}$ ,  $\pi_1 = \{a, c\}$  et  $\varphi_1 = (a \wedge \neg b) \vee (b \wedge d)$ . Le jeu booléen  $G = \{A, V, \pi_1, \varphi_1\}$  est totalement défini.*

*En effet, on sait que  $\pi_2 = \overline{\pi_1}$ , et que*

$$\varphi_2 = \neg\varphi_1 = (\neg a \vee b) \wedge (\neg b \vee \neg d)$$

*On peut à présent construire la forme normale de  $G$  (1 signifie que  $a_1$  gagne et  $a_2$  perd et vice-versa pour 0) :*

$a_1 \backslash a_2$	$\emptyset$	$\{b\}$	$\{d\}$	$\{b, d\}$
$\emptyset$	0	0	0	1
$\{a\}$	1	0	1	1
$\{c\}$	0	0	0	1
$\{a, c\}$	1	0	1	1

*On constate ici que le joueur  $a_2$ , qui contrôle les variables  $b$  et  $d$  a une stratégie gagnante. En effet, s'il choisit de mettre  $b$  à vrai et  $d$  à faux, il est sûr de gagner quels que soient les choix de  $a_1$ .*

*La propriété 1 nous permet de vérifier ce résultat. En effet, le but de  $a_2$  :*

$$\begin{aligned} \varphi_2 &= (\neg a \vee b) \wedge (\neg b \vee \neg d) \\ &= (\neg a \wedge \neg b) \vee (\neg a \wedge \neg d) \vee (b \wedge \neg d) \end{aligned}$$

*contient un impliquant premier  $(b \wedge \neg d)$  qui est composé uniquement de variables contrôlées par  $a_2$ .*

*Par contre, le but du joueur  $a_1$ ,  $\varphi_1 = (a \wedge \neg b) \vee (b \wedge d)$ , ne contient aucun impliquant premier composé uniquement des variables  $a$  et  $c$ .  $a_1$  n'a donc pas de stratégie gagnante.*

La propriété suivante donne une caractérisation simple des équilibres de Nash en stratégies pures dans les jeux booléens à deux joueurs et à somme nulle qui est obtenue par une simple transposition de résultats issus de la théorie des jeux à somme nulle.

**Propriété 3** si  $G$  est un jeu booléen à deux joueurs et à somme nulle,  $S = (s_1, s_2)$  est un équilibre de Nash en stratégies pures si et seulement si  $s_1$  est une stratégie gagnante pour le joueur 1 ou  $s_2$  est une stratégie gagnante pour le joueur 2.

*Preuve :*

1. Soit  $S = (s_1, s_2)$  un équilibre de Nash en stratégies pures.
  - Supposons que l'on a  $u_1(S) = 1$ . Le jeu étant à somme nulle, on a  $u_2(S) = 0$ . Comme  $S$  est un équilibre de Nash,  $\forall s'_2, u_2(S) \geq u_2(s_1, s'_2)$  ce qui entraîne  $\forall s'_2, u_2(s_1, s'_2) = 0$ . Donc,  $\forall s'_2, (s_1, s'_2) \models \neg\phi_2$ , soit  $\forall s'_2, (s_1, s'_2) \models \phi_1$ .  $s_1$  est donc une stratégie gagnante pour le joueur 1.
  - On a  $u_1(S) = 0$  et  $u_2(S) = 1$ . En faisant le même raisonnement que précédemment, on montre que  $s_2$  est donc une stratégie gagnante pour le joueur 2.
2. Supposons que  $s_1$  est une stratégie gagnante. On a donc  $\forall s_2, u_1(s_1, s_2) = 1$  et  $\forall s_2, u_2(s_1, s_2) = 0$ . Posons  $S = (s_1, s_2)$ . On a bien  $\forall s'_1, u_1(S) \geq u_1(s'_1, s_2)$  et  $\forall s'_2, u_2(S) \geq u_2(s_1, s'_2)$ .  $S$  est donc bien un équilibre de Nash.  
On raisonne de la même façon si l'on suppose que  $s_2$  est une stratégie gagnante. ■

Par conséquent, dans un jeu booléen à deux joueurs et à somme nulle, il existe un équilibre de Nash en stratégies pures si et seulement si l'un des deux joueurs a une stratégie gagnante.

## 5 Complexité algorithmique

La propriété 3 permet de déterminer facilement la complexité algorithmique du problème d'existence d'un équilibre de Nash en stratégies pures. On rappelle que  $\Sigma_2^P = \text{NP}^{\text{NP}}$  est la classe des langages reconnaissables en temps polynomial par une machine de Turing non-déterministe munie d'oracles NP (voir [11]). Les résultats de complexité qui suivent n'étant pas centraux dans cet article, nous ne donnons pas plus de détails.

**Propriété 4** Le problème de l'existence d'un équilibre de Nash en stratégies pures dans un jeu booléen à deux joueurs et à somme nulle est  $\Sigma_2^P$ -complet.

*Preuve :* L'appartenance à  $\Sigma_2^P$  est immédiate ; la difficulté est obtenue par la réduction polynomiale suivante de  $\text{QBF}_{2,\exists}$  : à toute instance  $Q = (\langle A, B, \Phi \rangle$  de  $\text{QBF}_{2,\exists}$ , où  $\Phi$

est une formule propositionnelle de  $L_{A \cup B}$ , avec  $A \cap B = \emptyset$ , on associe le jeu booléen suivant  $G_Q$  à deux joueurs et à somme nulle :  $\phi_1 = (\Phi \leftrightarrow x)$ , où  $x$  est une nouvelle variable (n'apparaissant pas dans  $A \cup B$ ), et  $\pi_1 = A \cup \{x\}$ ,  $\pi_2 = B$ . Il est clair qu'il n'existe pas de stratégie gagnante pour  $a_2$ . Par conséquent, il existe un équilibre de Nash en stratégies pures pour  $G_Q$  si et seulement si il existe une stratégie gagnante pour  $a_1$ , ce qui est équivalent à ce que  $\exists A \forall B \Phi(A, B)$  soit une instance positive de  $\text{QBF}_{2,\exists}$ . ■

On en tire le corollaire suivant, concernant cette fois les jeux booléens à  $n$  joueurs.

**Corollaire 1** Le problème de l'existence d'un équilibre de Nash en stratégies pures dans un jeu booléen à  $n$  joueurs est  $\Sigma_2^P$ -complet.

Ce résultat est à rapprocher de la complexité de la contrôlabilité – également un problème  $\Sigma_2^P$ -complet [9]. Or, le problème de l'existence d'un équilibre de Nash dans un jeu à plusieurs joueurs et à somme non nulle étant bien plus général que la contrôlabilité, le fait qu'il ne soit pas situé plus haut dans la hiérarchie des classes de complexité est plutôt une bonne nouvelle.

On peut expliquer intuitivement le fait que le problème de l'existence d'un équilibre de Nash soit au second niveau de la hiérarchie polynomiale par le fait que la résolution de ce problème comporte deux sources indépendantes de complexité NP-difficiles : la recherche de la "bonne" combinaison de stratégies, et la vérification qu'elle constitue un équilibre de Nash en stratégies pures. Par comparaison, l'existence d'une combinaison de stratégies dont l'utilité cumulée est supérieure à une borne donnée est seulement un problème NP-complet.

De la propriété 4 nous tirons aussi ce second corollaire :

**Corollaire 2** Le problème de l'existence d'une coalition efficace dans un jeu booléen à  $n$  joueurs est  $\Sigma_2^P$ -complet.

*Preuve :* L'appartenance à  $\Sigma_2^P$  est immédiate ; la difficulté est obtenue, dès que  $N = 2$ , par la réduction polynomiale suivante de l'existence d'un équilibre de Nash en stratégies pures dans un jeu booléen à deux joueurs et à somme nulle. Soit  $G$  un tel jeu, où  $\phi_1$  est le but du joueur 1. Comme  $\phi_1 \wedge \phi_2 \equiv \phi_1 \wedge \neg\phi_1 \equiv \perp$ ,  $\{1, 2\}$  ne peut pas être une coalition efficace, les seules coalitions efficaces possibles sont  $\{1\}$  et  $\{2\}$ , d'où le résultat. ■



## 6 Utilités non binaires

Les jeux booléens considérés jusqu'à présent ne permettent d'exprimer que des utilités binaires, et constituent donc un cadre quelque peu simpliste. Cette hypothèse simplifi catrice avait l'avantage de permettre une exposition claire, et de donner des bornes inférieures de complexité algorithmique. Dans cette section, nous montrons que la généralisation à des utilités quelconques peut être faite sans grande difficulté, et, qui plus est, sans saut de complexité.

On a dit en introduction qu'il existait plusieurs langages logiques de représentation compacte des préférences ; par souci de simplicité et de brièveté, nous n'allons en considérer qu'un seul, à savoir celui des *formules pondérées*, où les préférences d'un agent sont décrites par un ensemble de buts, chaque but ayant un poids propre.

La définition des jeux booléens à utilités non binaires est la même que celle de la section 4.1 (cf. définition 3), à une différence près : à la place des buts des joueurs, représentés par les formules propositionnelles  $\phi_1, \dots, \phi_n$ , on a pour chaque joueur un ensemble de formules associées à un poids numérique :

$$\Phi = \langle \phi_1, \dots, \phi_n \rangle$$

et pour chaque joueur  $a_i$ ,

$$\phi_i = \{(\phi_{i,1}, w_{i,1}), \dots, (\phi_{i,p_i}, w_{i,p_i})\}$$

où  $\phi_{i,1}, \dots, \phi_{i,p_i} \in L_V$  et  $w_{i,1}, \dots, w_{i,p_i} \in \mathbb{R}$ .

Chacune des formules  $\phi_{i,j}$  est un "but élémentaire" et  $w_{i,j}$  est sa contribution à l'utilité du joueur  $a_i$ , cette fonction d'utilité étant définie comme la somme des valeurs numériques des buts élémentaires satisfaits : pour toute combinaison de stratégies  $S$ , et tout joueur  $a_i$ ,

$$u_i(S) = \sum \{w_{i,k} \mid S \models \phi_{i,k}\}$$

(avec la convention  $\sum \emptyset = 0$ ).

La notion de stratégie gagnante n'a évidemment plus de sens, mais on peut la généraliser par la notion d'utilité minimale garantie par une stratégie : si le joueur  $a_i$  choisit la stratégie  $s_i$ , son utilité minimale garantie est

$$u_i^*(s_i) = \min_{s_{-i} \in 2^{V \setminus \pi_i}} u_i(s_i, s_{-i})$$

**Exemple 6** Soit  $G = \{A, V, \pi, \Phi\}$  un jeu booléen à utilité non binaire. On donne :

- $V = \{a, b, c\}$ ,  $A = \{a_1, a_2, a_3\}$
- $\pi_1 = \{a\}$ ,  $\pi_2 = \{b\}$ ,  $\pi_3 = \{c\}$ ,
- $\phi_1 = \{((\neg a \wedge \neg b), 2), ((a \wedge c), 1)\}$ ,
- $\phi_2 = \{((\neg a \vee \neg c), 2), ((b \wedge c), 1)\}$ ,
- $\phi_3 = \{((a \wedge \neg b), 2), ((a \wedge b), 1)\}$ .

La table suivante donne l'utilité de chaque joueur pour chaque combinaison de stratégies.

$S$	$u_1$	$u_2$	$u_3$
$(a, b, c)$	1	1	1
$(a, b, \neg c)$	0	2	1
$(a, \neg b, c)$	1	0	2
$(a, \neg b, \neg c)$	0	2	2
$(\neg a, b, c)$	0	3	0
$(\neg a, b, \neg c)$	0	2	0
$(\neg a, \neg b, c)$	2	2	0
$(\neg a, \neg b, \neg c)$	2	2	0

On a  $u_1^*(a) = u_1^*(\neg a) = 0$ ;  $u_2^*(b) = 1$  et  $u_2^*(\neg b) = 0$ ;  $u_3^*(c) = u_3^*(\neg c) = 0$ .

Les résultats de complexité donnés précédemment donnent comme corollaires :

- l'existence pour le joueur  $a_i$  d'une stratégie d'utilité minimale garantie supérieure à un seuil donné est un problème  $\Sigma_2^P$ -complet ;
- l'existence d'un équilibre de Nash en stratégies pures est un problème  $\Sigma_2^P$ -complet.

La généralisation à des utilités non binaires, avec le choix de ce langage de représentation, ne s'accompagne donc pas d'un saut de complexité.

Il reste à étudier de plus près les notions de coalitions efficaces et leur calcul. Une première approche consiste à supposer que les utilités sont transférables, et que l'utilité d'un groupe d'agents est la somme des utilités de chaque joueur du groupe, et à définir une coalition stable comme un sous-ensemble  $I$  permettant d'obtenir une utilité minimale garantie supérieure à la somme des utilités minimales garanties obtenues pour chaque partition de  $I$ . Ainsi, la coalition  $\{1, 2\}$ , en jouant  $(\neg a, \neg b)$ , permet d'obtenir les utilités minimales garanties  $u_1^*(\neg a, \neg b) = u_2^*(\neg a, \neg b) = 2$ , alors que le joueur  $a_1$  (resp.  $a_2$ ), isolé, ne peut avoir qu'une utilité minimale garantie de 0 (resp. 1). On a donc  $u_1^*(\neg a, \neg b) + u_2^*(\neg a, \neg b) > \max_{s_1} u_1^*(s_1) + \max_{s_2} u_2^*(s_2)$ , c'est-à-dire que  $\{1, 2\}$  est stable. Par contre,  $\{1, 2, 3\}$  n'est pas stable, puisque  $\max_{S \in 2^{\{a,b,c\}}} u_1(S) + u_2(S) + u_3(S) = 4$ , alors que  $u_1^*(\neg a, \neg b) + u_2^*(\neg a, \neg b) = 4$ .  $\{1, 2, 3\}$  ne réalise donc pas mieux que  $\{1, 2\}$ . Il faut noter

qu'on s'approche ici de notions étudiées dans les jeux coopératifs, en particulier la notion de cœur. Une étude précise de ces notions dans un cadre booléen, ainsi que de leur complexité algorithmique, sera l'objet de travaux ultérieurs.

## 7 Conclusion

La notion de jeu booléens étudiée par [7, 5] semble prometteuse quand on cherche à modéliser de manière compacte les préférences des joueurs dans le cadre de jeux statiques. Toutefois, ces jeux booléens correspondent à une spécification très particulière (2 joueurs, jeux à somme nulle, utilités binaires) et nous avons donc du étendre cette notion. Cette extension porte sur plusieurs points :

- accepter un nombre quelconque de joueurs (et plus seulement 2) ; la notion de coalition de joueurs prend alors tout son sens ;
- envisager des jeux à somme non nulle ; ainsi le gain d'un joueur n'est plus la perte d'un autre ;
- quantifier les gains et les pertes par des utilités non binaires ; cela nous permet de sortir du cadre trop simple du "perdant ou gagnant" et de proposer des notions de préférence sur les stratégies des joueurs : un joueur gagne plus ou moins suivant la stratégie de jeu qu'il suit et dans ce cadre un problème intéressant consiste à savoir comment optimiser son gain (éventuellement à l'aide d'une coalition).

Bien entendu, les jeux booléens évoqués par Harrenstein et co. dans [7, 5] peuvent être vus comme des cas particulier de ces "jeux booléens étendus".

En parallèle, nous avons aussi établi un lien entre les jeux booléens (étendus ou pas) et les notions d'impliquants et d'impliquants premiers (notions bien connues de tous ceux qui s'intéressent au problème de la satisfiabilité en logique classique). Ainsi, pour un joueur ou une coalition de joueurs, la détection de stratégies gagnantes dans un jeu booléen revient au simple calcul des impliquants premiers liés à l'ensemble des variables contrôlées par ce joueur ou cette coalition de joueurs. Nous avons également identifié la complexité algorithmique de quelques problèmes liés aux jeux booléens.

Ce travail, encore très préliminaire, ouvre de nombreuses perspectives :

- poursuivre l'extension des jeux booléens pour permettre aux joueurs de partager le contrôle

- des variables ; une même variable pourra alors être contrôlée par plusieurs joueurs ;
- affiner la notion "d'utilités non binaires", notamment en considérant d'autres langages de représentation de préférences.
- étudier la notion de préférences entre coalitions, et établir des liens avec des notions bien connues en théorie des jeux coopératifs (valeur de Shapley, cœur etc.) ;
- définir des relations de conséquence sur le modèle des travaux d'Harrenstein [6] à partir de ces jeux booléens étendus ;
- modéliser ces jeux à l'aide de la logique des actions ;
- utiliser ces jeux pour l'étude de processus de raisonnement faisant intervenir  $n$  agents en interaction (argumentation, prise de décision, ...).

## Références

- [1] C. Boutilier. Toward a logic for qualitative decision theory. In *KR-94*, pages 75–86, 1994.
- [2] L. Cholvy and Garion C. Deriving individual goals from goals allocated to a group of agents. *JEDAI*, 2002.
- [3] S. Coste-Marquis, J. Lang, P. Liberatore, and P. Marquis. Expressive power and succinctness of propositional languages for preference representation. In *Proceedings of KR 2004*, pages 203–212, 2004.
- [4] P.E. Dunne and W. van der Hoek. Representation and complexity in boolean games. In *Proc. of JELIA2004*, volume LNCS 3229, pages 347–359. José Júlio Alferes et João Alexandre Leite (eds), 2004.
- [5] P. Harrenstein. *Logic in Conflict*. PhD thesis, Utrecht University, 2004.
- [6] P. Harrenstein. Logical consequence and the theory of games. *Philosophia Scientiae*, 8(2) :179–193, 2004. Special issue on logic and games.
- [7] P. Harrenstein, W. van der Hoek, J.J Meyer, and C. Witteveen. Boolean games. In J. van Benthem, editor, *Proceedings of TARK 2001*, pages 287–298. Morgan Kaufmann, 2001.
- [8] J. Lang. Logical preference representation and combinatorial vote. In *Annals of Mathematics and Artificial Intelligence*. Kluwer, 2004.
- [9] J. Lang and P. Marquis. Two forms of dependence in propositional logic : controllability and definability. In *Proceedings of AAI-98*, pages 268–273, 1998.
- [10] P. Marquis. *Consequence Finding Algorithms*, in *Handbook on Defeasible Reasoning and Uncertainty Management Systems*, volume 5 : Algorithms for Defeasible and Uncertain Reasoning, S. Moral and J. Kohlas, chapter 2, pages 41–145. Kluwer Academic Publisher, 2000.
- [11] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
- [12] L. J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3 :1–22, 1977.