Mémoire de DEA

DEA Systèmes Intelligents et Applications

Jeux à n joueurs : simulations

Présenté le 28 juin 2004

LALLART Thomas - Faculté des Sciences Jean Perrin

Remerciements

Je tiens à remercier toutes les personnes qui m'ont permis de mener ce travail à bien.

Tout d'abord, je remercie mes maîtres de stages Pierre Marquis, Sébastien Konieczny et Bruno Beaufils, qui m'ont toujours encouragé lors de cette étude. Par leur soutien et leurs conseils, il m'ont permis de rester très motivé tout au long du stage et d'aller le plus loin possible sur cette étude.

Un remerciement spécial est adressé à Bruno Beaufils qui s'est énormément impliqué pour me permettre d'avoir accès aux machines de l'IUT de Lille. Sans cette intervention, il m'aurait été impossible d'effectuer les nombreux tests qui font partie de cette étude.

Ensuite, je remercie les enseignants du CRIL qui ont contribué à ma formation universitaire.

Enfin, je tiens à remercier mes parents qui m'ont toujours encouragé à poursuivre mes études le plus loin possible et sans qui ce travail n'aurait sans doute pas pu être réalisé.

Table des matières

Liste des tableaux

Table des figures

Introduction

Le but de ce stage de DEA est l'étude d'un jeu à n joueur grâce à des expérimentations via la conception d'un simulateur. Un jeu est un conflit d'intérêt entre plusieurs participants. L'objectif de chacun d'entre eux est de maximiser ses gains (ce qui représente l'issue du jeu pour chaque joueur) de manière individuelle ou éventuellement collective, en suivant une stratégie et en fonction de règles clairement établies.

Le jeu qui nous concerne est un jeu à n joueurs dans lequel les participants vont pouvoir s'allier contre d'autres. Ce type de comportement forme des coalitions. Le coeur du sujet est donc l'analyse d'un jeu à n joueurs dans lequel les joueurs vont avoir la possibilité de se regrouper au sein de coalitions. De part sa nature ce type de jeu est particulièment complexe à étudier. En effet les possibilités d'alliance entre joueurs sont nombreuses.

Le domaine de recherche principal sur ce sujet est la théorie des jeux mais les recherches sur les jeux proposant ces caractéristiques sont rares. C'est pour cela que la plupart des études sur ce type de jeux reposent sur des simulations. Les stratégies sont testées de manière virtuelle et non pas traitées de manière théorique.

Le simulateur est utilisé pour générer des parties et des tests afin de dégager de bonnes stratégies des simulations réalisées. Un second objectif est d'analyser les coalitions formées lors des expérimentations.

Pour cette étude, des connaissances théoriques préalables issues de différents domaines sont nécessaires.

En premier lieu, j'ai étudié les principes de base de la théorie des jeux. Après une introduction générale de la théorie des jeux et un historique, nous définirons la théorie de l'utilité qui est l'unité de mesure des gains dans un jeu. Ensuite nous verrons comment représenter les jeux de manière théorique.

Je me suis ensuite interessé à la résolution des jeux. Nous définirons formellement les notions de solution, de stratégie et d'équilibre qui permettent de résoudrent certains jeux. Nous mettrons en évidence le fait que ces méthodes de résolution ne permettent pas de trouver une solution pour tous les jeux.

Dans le troisième chapitre, j'exposerai les méthodes experimentales existantes pour trouver la solution de certains jeux ou pour découvrir les meilleures stratégies. Cette partie repose sur le jeu du dilemme du prisonnier pour lequel nous appliquerons ces méthodes de recherche. Nous nous intéresserons plus particulièrement au cas où le jeu est répété un certains nombre de fois.

Dans le quatrième chapitre, je me suis plus particulièrement intéressé aux jeux à n joueurs dans lesquelles les joueurs pouvaient éventuellement communiquer pour coopérer. Après une explication sur ce qui les différencient des jeux à deux joueurs, nous définierons les notions de base permettant de travailler avec n joueurs. Ensuite nous spécifierons la forme coalitionnelle d'un jeu qui est fréquemment utilisée pour étudier les jeux à n joueurs et les propriétés induites par cette représentation.

Le cinquième chapitre détaille le jeu à étudier. Nous y définirons clairement le jeu, ses règles et les notions utilisées en cours de l'analyse. Ensuite nous présenterons le simulateur qui a été codé pour réaliser les tests. Enfin nous détaillerons les stratégies qui ont été définies afin d'effectuer les expérimentations.

Le sixième chapitre décrit les tests effectués et l'analyse des résultats. Les objectifs de chaque test sont détaillés et les résultats sont ensuite interprétés. Nous concluerons

Enfin le dernier chapitre revient sur la résolution d'un jeu à n joueur et la recherche de coalition dans un cadre théorique. Nous présenterons les notions de coalitions stables, de valeur de Shapley, du coeur et de nucleolus et nous montrerons qu'elles sont difficilement applicables lorsque n est grand.

Chapitre 1

Généralités sur la théorie des jeux

1.1 Introduction

La théorie des jeux se propose principalement de trouver ce que doit faire un participant à une situation stratégique (c'est à dire une situation de conflit où l'action d'un participant influe sur les autres) de façon à obtenir un résultat le plus avantageux pour lui (c'est ce qu'on supposera tout au long de ce document). Ce type de comportement pour un joueur est dit **rationnel**. Avec cette première définition, on réalise rapidement l'ampleur de la tâche proposée. En effet, quel que soit le jeu et la stratégie employée par le joueur rationnel, les autres joueurs vont agir de la même façon. Or, comme généralement ce qui est avantageux pour l'un ne l'est pas pour l'autre, les joueurs cherchant chacun leur intérêt, on se trouve devant un problème de circularité. De plus, la part du hasard et des événements exogènes peut être plus ou moins importante dans le jeu et les règles plus ou moins complexes. Après cette rapide analyse, on réalise que la théorie des jeux ne peut pas inclure toutes les situations de conflit d'intérêt. On se rend compte également que les procédures de décision ne peuvent pas toujours être représentées de la même façon que dans la réalité (en particulier en économie). Par exemple, les décisions ne sont pas toujours de simples alternatives mais peuvent nécessiter une spécification de temps qui est particulièment difficile à représenter formellement.

Cependant, nous verrons que “notre analyse mathématique du problème montrera qu'il existe en fait une famille non négligeable de jeux où une solution peut être trouvée [...] Dans ce cas, chaque participant obtient au moins ce qui lui revient s'il agit de façon rationnelle. C'est en fait ce qu'il obtient si les autres participants se comportent aussi de facon rationnelle ; s'il ne le font pas, il peut même obtenir plus.”(Von Neumann & Morgenstern : *Theory* *of Games and Economic Behaviour*).

Les jeux à étudier peuvent être très simples ou au contraire particulièrement complexes. Il peut y avoir de deux à n joueurs (n pouvant même être un nombre infini). Dans les jeux à n joueurs, les joueurs peuvent éventuellement communiquer pour coopérer et former des coalitions afin de maximiser leurs gains. Les joueurs peuvent ou ne peuvent pas avoir accès à certaines informations sur le jeu (historique des coups précédents, stratégies disponibles par les autres joueurs,...). Les joueurs peuvent ou ne peuvent pas savoir quand le jeu se termine. Toutes ces variantes rendent le problème plus ou moins ardue à analyser.

La théorie des jeux englobe un champ d'application très vaste : biologie de l'évolution, sciences économiques, sciences sociales, stratégies militaires, jeux vidéos,... Dans ces différents champs, les mots n'ont pas toujours le même sens et les situations de jeux ne sont pas les mêmes. Les économistes travaillent beaucoup sur la théorie des jeux. Cependant, certaines de ces études ne sont pas à considérer par l'informaticien car certains critères comme la rationalité sont discutables (le lecteur pourra consulter [18, chapitre 2] pour plus d'informations sur les raisons). Par exemple, d'un point de vue pratique aucun joueur ne peut prétendre être totalement rationnel.

1.2 Petit historique

La théorie des jeux trouve pour origine la théorie des probabilités. Les premiers travaux sur le sujet sont dûs â Girolamo Cardano qui exposa le concept mathématique d'anticipation et les lois de répétitions des événements (1663). Antoine Gombault, Blaise Pascal et Pierre de Fermat travaillèrent sur "la façon de répartir entre deux joueurs l'enjeu d'un jeu de hasard, si celui-ci est inachevé et qu'un des joueurs a l'avantage sur l'autre" (1654). Ils déduisirent pour la première fois une solution permettant de prévoir l'avenir en terme de probabilités. En 1657, Huyghens proposa le concept de compétition dans les jeux de chance. Puis au début du XVIIIème siècle, les Bernouilli fîrent de grandes découvertes dans le domaine des probabilités, Cournot définit un équilibre lors des situations de compétition en économie. Enfin De Moivre définit la structure de la loi normale et le concept d'écart-type.

Pour ce qui concerne la théorie des jeux par elle-même, les textes fondateurs sont ceux de Von Neumann et Morgenstern (définition de la notion de fonction d'utilité, théorème du minimax) au milieu et à la fin de la première moitié du XXième siècle suivi de ceux de John Nash. Notons également les travaux de Waldegrave sur les stratégies dans les jeux de hasard. Une fois les bases jetées, les recherches sur la théorie des jeux s'intensifièrent. En effet, en période de guerre froide, les applications pour les négociations et les stratégies militaires de cette théorie ont contribué à son succès. Le développement de la théorie des jeux a abouti à de nombreux problèmes (pas d'équilibre, équilibres multiples, auto-apprentissage de stratégies, jeux coopératifs ou non coopératifs à n joueurs) et à certaines améliorations (définitions de nouveaux équilibres, équilibres non pertinents pour certains jeux, découverte de stratégies dominantes ou dominées). Les recherches progressèrent rapidement avec les travaux des économistes, des mathématiciens et des informaticiens.

Concrétement, on distingue trois grandes périodes de recherche sur la théorie des jeux.

La première se situe entre les années vingt et quarante. C'est la période d'élaboration de la théorie des jeux. On travaille sur de simples jeux de hasard mais les chercheurs s'intéressent à des extensions des jeux de société.

La seconde période de 1944 (Parution du livre de *Theory of Games and Economic Behavior* de Von Neumann et Morgenstern) à la fin des années soixante-dix s'est essentiellement occupée des jeux coopératifs.

La dernière se situe du début des années quatre-vingt à nos jours. Les études reposent plus sur les jeux non coopératifs et sur le rôle des croyances des joueurs dans le jeu.

Les recherches s'orientent selon deux points de vues : normatif et descriptif. L'approche normative s'intéresse à découvrir l'issue d'un jeu avant même que la partie soit jouée (approche la plus étudiée). L'approche descriptive part d'une situation de jeu et cherche à comprendre comment cette situation a été obtenue.

1.3 Théorie de l'utilité

L'utilité est le terme utilisé pour désigner les gains d'un joueur dans un jeu. Cette notion est issue des travaux de Von Neumann & Morgenstern. Cette théorie nous permettra de pouvoir traiter les gains des joueurs et d'évaluer les résultats des stratégies choisies. Cela dit, il est clair que nous ne pouvons pas considérer de la même façon les utilités pour un individu et pour un groupe (ou une coalition) : les motivations et les intérêts d'un groupe ne sont pas toujours les mêmes que celle des individus qui le composent (ils peuvent même avoir des motivations conflictuelles). De plus, les effets d'une prise de décision ne sont pas nécessairement connus. Une classification des effets d'une prise de décision peut être la suivante : certaine, risquée, incertaine. Nous pouvons alors dire que la prise de décision est :

- certaine si chaque action produit invariablement un résultat connu.
- risquée si chaque action conduit à un ensemble de résultats possibles,

chaque résultat arrivant avec une probabilité connue. Ces probabilités sont connues de celui qui prend la décision.
- incertaine si chaque action mène à un ensemble de résultats possibles, chaque résultat arrivant avec une certaine probabilité. Mais dans ce cas, celui qui prend la décision ne connait pas les probabilités associées à ces résultats.

Dans le cadre des actions effectuées avec certitude nous devons maintenant définir formellement ce qu'est une prise de décision d'un joueur c'est-à-dire le choix d'une stratégie (ou d'une action) parmi un ensemble de stratégies applicables. Soit x une action d'un ensemble F d'actions applicables et f une fonction qui associe à tout x une valeur $f(x)$. Typiquement, la prise de décision consiste à trouver $x_0 \in F$ qui produit le meilleur résultat pour le joueur c'est-à-dire tel que $f(x_0) \geq f(x), \forall x \in F$. Le coeur du problème est de trouver les gains associés à chaque action. Dans le contexte économique par exemple, ces valeurs vont être très souvent les profits et les pertes engendrés par l'action choisie. Mais dans d'autres situations, les actions n'engendrent pas nécessairement de telles quantités. Il faut pourtant trouver un moyen d'associer un "gain" à chaque action. Prenons un exemple : supposons qu'un joueur ait le choix entre dix actions, chacune d'entre elles lui permet de gagner un objet différent. Il peut alors "trier" ces objets par ordre de préférence et donc associer une valeur de 1 à celui qu'il aime le moins, 2 au suivant et ainsi de suite jusque 10 pour celui qu'il préfère. Le joueur définit ainsi un préordre total sur l'ensemble des alternatives. Il choisira donc l'action x le conduisant à obtenir l'objet qu'il préfère lui apportant un gain (une utilité) de $f(x) = 10$. Ce premier exemple nous montre une première propriété de l'utilité : la transitivité. Si on compare les objets deux à deux, si l'objet A est préféré à l'objet B et l'objet B est préféré à l'objet C alors l'objet A est préféré à l'objet C (l'utilité associée au gain de l'objet A est supérieure à celle associée au gain de l'objet C). Remarquons également que la valeur associée à chaque objet est sans importance tant qu'elle respecte l'ordre : pour trier les dix objets le joueur pouvait également associer les valeurs 10,14.21,...,137 au lieu de 1,2,3,...,10 sans que cela modifie ses choix. Il est important de ne pas voir en ces nombres les mêmes propriétés que celles qu'on utilise habituelement : la seule propriété numérique signifiante est l'ordre. Il est important de préciser que la comparaison entre utilités (induisant la notion d'ordre) ne peut se faire qu'entre les utilités d'un même joueur. En effet, l'utilité représentant l'intérêt d'un joueur pour un résultat, chaque joueur n'attache pas la même importance à un même résultat. Si on reprend l'exemple, un second joueur ne va pas nécessairement trier les objets selon le même ordre que le premier. Le résultat avec la plus grande utilité pour le premier joueur peut même être celui avec la plus petite pour le second (sur l'exemple, l'objet préféré du premier joueur peut être celui qui est le moins bien classé pour le second).

Si on se place du point de vue de la prise de décision dans le risque, il nous faut modifier la façon d'associer les utilités aux actions. Soient $(s_1, s_2, ..., s_n)$ l'ensembles des actions applicables par un joueur, $(a_1, a_2, ..., a_n)$ des gains associés respectivement à chaque action et $(p_1, p_2, ..., p_n)$ les probabilités (entre 0 et 1) associées respectivement à chaque résultat tel que $\sum_{i=0}^{n} p_i = 1$. Le joueur a connaissance de ces probabilités. Le gain espéré est donc $a_1p_1 + a_2p_2 + ... + a_np_n$.

Prenons un nouvel exemple : un joueur a le choix entre trois alternatives (pris ici dans son acceptation anglo-saxonne de choix possible) A,B ou C. Supposons qu'il préfère A à B et B à C. Maintenant supposons que le résultat de B soit certain mais que les résultats de A et de C soient incertains avec une probabilité de p pour A et $1 - p$ pour C. Il est clair que si la valeur de p est trop faible, le joueur préferera alors l'alternative B. Le joueur devra exprimer la valeur de p à partir de laquelle son choix change de A vers B (ce qui n'est pas forcément évident). Si par exemple A est une information totalement certaine ($p = 1$) et C totalement incertaine ($1 - p = 0$) et que le joueur avait décidé que son choix changait à partir de $p = \frac{2}{3}$, alors les utilités assignées respectivement à A,B et C seront $1, \frac{2}{3}, 0$.

Pour assigner correctement les utilités, Von Neumann & Morgenstern ont proposé des axiomes requis pour une assignation cohérente des utilités.

i. Tout couple d'alternatives doit être comparable c'est-à-dire que pour tout couple, le sujet préferera l'une ou l'autre ou il sera indifférent.

ii. Les relations de préférence et d'indifférence pour les loteries (les alternatives) sont transitives c'est-à-dire qu'étant donné trois loteries A,B et C, si un joueur préfère A à B et B à C alors il préfère A à C ; et s'il est indifférent entre A et B et entre B et C alors il est indifférent entre A et C.

iii. Dans le cas où une loterie a parmi ses alternatives une autre loterie alors la première loterie est décomposable en alternatives plus basiques grâce à l'utilisation de la théorie des probabiltés.

iv. Si deux loteries sont indifférentes pour un joueur alors elles sont interchangeables comme alternatives dans toute loterie qui la compose.

v. Si deux loteries impliquent les deux mêmes alternatives alors l'alternative préférée qui a une plus grande probabilité d'arriver est elle-même préférée.

vi. Si A est préférée à B et B à C alors il existe une loterie impliquant A et C (avec les probabilités associées) qui est indifférente à B.

A partir des axiomes on peut montrer que les nombres peuvent être assignés aux alternatives basiques tel que un coup est préféré à un autre si et seulement si l'utilité attendue du premier est supérieure à l'utilité du second. Si μ est une telle utilité, toute autre utilité est en relation avec elle par une transformation affine, c'est-à-dire qu'il existe une constante positive a et une constante b tel que $au + b$ est la seconde utilité. Une telle fonction est appelée fonction d'utilité linéaire, où linéaire signifie que l'utilité du coup est la valeur attendue des utilités de ses composants. Plus clairement ce résultat montre que l'on peut changer les utilités associées aux joueurs par une transformation affine $au + b$ sans changer leurs préférences, ni l'analyse et l'issue du jeu.

Pour plus d'informations on pourra consulter [24, chapitre 2].

1.4 Représentation des jeux

1.4.1 Notations et définitions

Définition 1 ***Un profil de stratégies*** *$s = \langle s_1, s_2, ..., s_n \rangle$ représente une prise de décision de chaque joueur : le joueur 1 joue la stratégie s_1, le joueur 2 joue la stratégie s_2, etc... On définit $s_{-i} = \langle s_1, s_2, ..., s_{i-1}, s_{i+1}, ..., s_n \rangle$, le profil des stratégies autres que celles du joueur i.*

Une stratégie correspond au choix d'un joueur parmi plusieurs alternatives. On notera $S_i = \langle s_1, s_2, ..., s_n \rangle$, l'ensemble des stratégies (ou alternatives) disponibles pour le joueur i et S l'ensemble de tous les profils de stratégies.

Rappelons que les gains des joueurs à chaque coup sont aussi appelés "utilités". Les utilités ne peuvent pas être comparées entre joueurs. En effet, un joueur peut associer une utilité de 10 au fait de gagner 100 euros et un autre une utilité de 2 au fait d'en gagner 10000. Donc, un joueur s'attache à maximiser son gain sans se soucier de ce que cela procure à son adversaire. On notera $\mu_i : S \to I\!R$ la fonction qui associe l'utilité du joueur i pour le profil s et $\mu : S \to I\!R$ la fonction telle que $\mu(S) = \langle \mu_i(s), ..., \mu_n(s) \rangle$. Cette fonction est appelée **fonction d'utilité**.

1.4.2 Forme normale

La forme normale d'un jeu utilise une matrice des gains pour représenter le jeu. Dans le cas de 2 joueurs, les lignes et les colonnes sont les stratégies applicables pour chaque joueur. Les valeurs de la matrice sont des vecteurs représentant les gains pour chaque joueur. Un exemple est représenté dans le tableau 1.1. Les règles de ce jeu sont les suivantes. Les deux joueurs ont chacun deux cartes en main : un roi et un as. Ils doivent chacun en choisir une simultanément. On notera les gains de la façon suivante : si un joueur a joué un as et l'autre un roi alors le joueur qui a joué l'as gagne 1 et l'autre 0, si les deux jouent l'as alors les deux gagnent 1 enfin si les deux jouent le roi, les deux gagnent 2.

Joueur 1 / Joueur 2	Roi	As
Roi	(2,2)	(0,1)
As	(1,0)	(1,1)

TAB. 1.1 – Un jeu sous forme normale

Définition 2 *Un jeu sous forme stratégique est défini par :*
- *un ensemble* $N = \{1, 2, ..., n\}$ *de joueurs*
- *pour chaque joueur* i*, un ensemble de stratégies* $S_i = \{s_1, s_2, ..., s_{n_i}\}$
- *pour chaque joueur* i*, une fonction d'utilité* $\mu_i : S_1 \times S_2 \times ... \times S_n \to \mathbb{R}$

On remarque que ce type de représentation est très pratique dans le cas de jeu à deux joueurs car il permet une lecture rapide des stratégies et des gains des joueurs.

1.4.3 Forme extensive

La forme extensive d'un jeu utilise un arbre pour représenter le jeux. Ce type de représentation s'appelle aussi l'arbre de Kunh. Chaque branche de l'arbre représente un choix stratégique. Une représentation du jeu défini précédemment sous forme extensive est illustrée figure 1.1.

Définition 3 *Un jeu sous forme extensive est défini par :*
- *un ensemble* $N = \{1, 2, ..., n\}$ *de joueurs*
- *un arbre fini composé de :*
  - *un ensemble de noeuds* $\{A, B, C, ...\}$ *représentant les coups (les instants auxquels les joueurs prennent leur décision)*
  - *un ensemble de branches* $\{x, y, z, ...\}$ *représentant les stratégies*
- *une fonction de nommage qui indique à chaque noeud quel est le joeur qui doit jouer*
- *une fonction de valuation qui associe à chaque feuille un vecteur représentant les utilités de chaque joueur*

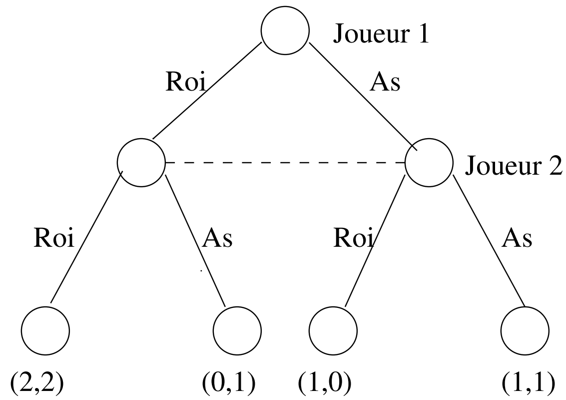

FIG. 1.1 – Un jeu sous forme extensive

La ligne pointillé entre les deux noeuds du joueur 2 est appelée un **ensemble d'information**. Il signifie que le joueur 2 ne sait pas dans quel noeud il se trouve c'est-à-dire qu'il ne connait pas la décision du joueur 1.

Ce type de représentation est trompeur car il laisse penser qu'il y a un ordre d'intervention des joueurs ce qui n'est pas toujours le cas (indiqué par les ensembles d'informations).

1.4.4 Types de jeu et propriétés

On peut définir plusieurs caractéristiques sur les jeux.

Les jeux à somme nulle sont tels que toute perte d'un joueur est redistribuée aux autres. Un jeu fini à somme nulle est également appelé jeu rectangulaire. La redistribution des gains a un rôle important dans ce type de jeu.

Les jeux à information complète supposent que tous les joueurs connaissent les utilités et les stratégies applicables par les autres. C'est ce qui est supposé dans beaucoup de jeux mais ce n'est pas forcément le cas.

Dans un **jeu à information parfaite**, les joueurs connaissent toujours le noeud sur lequel ils se trouvent dans l'arbre.

A l'inverse, dans un jeu à information imparfaite, les joueurs ne connaissent pas tous les choix passés, c'est-à-dire que les joueurs ne voient pas tous les noeuds de l'arbre et ne savent donc pas où ils se trouvent exactement dans l'arbre.

Dans un **jeu coopératif**, les joueurs peuvent se consulter avant de prendre une décision et respectent cette décision. Dans un jeu non coopératif, chaque joueur choisit finalement sa stratégie seul.

Dans ce premier chapitre, nous avons donc défini les bases de travail qui vont nous permettre de poursuivre cette étude. Nous avons introduit la notion de rationalité ainsi que la notion d'utilité. Puis nous avons montré les différents types de représentation d'un jeu. Enfin nous avons détaillé les principales propriétés des jeux.

Nous pouvons donc maintenant voir les différentes notions issues de la recherche permettant de résoudre un jeu.

Chapitre 2

Résolution des jeux

Dans ce chapitre, nous allons essentiellement nous intéresser aux jeux à deux joueurs. Résoudre un jeu, c'est deviner à l'avance quelle en sera l'issue en présence de joueurs rationnels. Cela revient à trouver quelles stratégies seront jouées par les joueurs en les supposant rationnels. Nous allons voir différents concepts permettant de prédire ces stratégies.

2.1 Equilibre de Nash

Les équilibres dans les jeux définissent les issues du jeu vers lesquelles le jeu va tendre si les joueurs sont rationnels. Tous les jeux ne possèdent pas nécessairement d'équilibre ou peuvent en posséder plusieurs. **L'équilibre de Nash (1951)** décrit une situation de jeu dans laquelle aucun joueur n'a d'intérêt à modifier unilatéralement sa stratégie compte tenu des stratégies des autres joueurs. L'équilibre de Nash est une généralisation de l'équilibre de Cournot.

Définition 4 *Soit s_{-i} l'ensemble des profils possibles de stratégies de tous les joueurs sauf i. L'équilibre de Nash se définit formellement de la façon suivante. Un profil $s = \langle s_1, ..., s_n \rangle$ est un équilibre de Nash si et seulement si pour tout joueur i, pour toute stratégie $s' \in S_i$: $\mu_i(s_1, s_{-i}) \geq \mu_i(s', s_{-i})$.*

Considérons l'exemple tableau 2.1 :

Cet exemple représente le jeu de la guerre des sexes : un couple veut sortir. Pour eux le plus important est d'être ensemble. Mais le mari préfère aller voir un match de foot (stratégie F) alors que la femme préfère aller à l'opéra (stratégie O).

Dans ce jeu, (O,O) et (F,F) sont des équilibres de Nash. En effet, pour (O,O) le joueur 1 a une utilité de 2 et s'il dévie vers F alors il aura une utilité

Joueur 1 / Joueur 2	O	F
O	(2,1)	(0,0)
F	(0,0)	(1,2)

TAB. 2.1 – Le jeu de la guerre des sexes

de 0. Le joueur 2 a une utilité de 1 et s'il dévie vers F alors il aura une utilité de 0. Le raisonnement est identique pour l'équilibre (F,F) si l'un des joueurs dévie vers O.

L'équilibre de Nash induit plusieurs problèmes. En particulier, il peut exister plusieurs équilibres de Nash, et dans ce cas, on ne sait pas a priori lequel choisir (c'est le cas du jeu de la guerre des sexes). Au contraire, il n'existe pas toujours d'équilibre de Nash pour un jeu donné en stratégies pures (voir section 2.7.1), et dans ce cas, on ne peut rien déduire sur le jeu grâce à cet équilibre. D'autres techniques, permettant de deviner vers quelle situation le jeu va tendre ont été développées.

2.2 Stratégies dominantes et dominées

2.2.1 Stratégies dominées

Une stratégie est dite **faiblement dominée** pour un joueur s'il existe une stratégie au moins égale à celle-ci dans toutes les situations.

Une stratégie est dite **strictement dominée** pour un joueur s'il existe une stratégie strictement meilleure à celle-ci dans toutes les situations.

Formellement :

Définition 5 *Une stratégie s_i est faiblement dominée pour le joueur i s'il existe une stratégie s'_i telle que pour tous les profils s_{-i} on a*

$$\mu_i(s'_i, s_{-i}) \geq \mu_i(s_i, s_{-i})$$

Définition 6 *Une stratégie s_i est strictement dominée pour le joueur i s'il existe une stratégie s'_i telle que pour tous les profils s_{-i} on a*

$$\mu_i(s'_i, s_{-i}) > \mu_i(s_i, s_{-i})$$

Nous pouvons supprimer d'un jeu les stratégies strictement dominées. En effet, en présence d'agents rationnels, ces stratégies ne seront jamais jouées et nous n'avons donc pas à les considérer pour l'étude de ce jeu.

Notons que la suppression des stratégies strictement dominées peut se faire dans n'importe quel ordre.

2.2.2 Stratégies dominantes

On dit qu'une stratégie s est **dominante** pour un joueur i si, quelles que soient les stratégies choisies par les autres joueurs, s maximise le gain du joueur i.

Formellement,

Définition 7 *La stratégie s est une stratégie dominante pour le joueur i si*

$$\forall s' \in S \forall s_{-i} \in S_{-i}, \mu_i(s, s_{-i}) \geq \mu_i(s', s_{-i})$$

Elle est strictement dominante si la comparaison est faite avec un ordre strict.

Il y a dualité entre les notions de stratégies dominantes et dominées. S'il existe une stratégie dominante alors toutes les autres sont dominées et s'il existe une stratégie dominée alors il existe au moins une stratégie qui la domine.

A partir de cette notion de stratégie dominante, nous pouvons définir la notion d'équilibre en stratégies dominantes. **Un équilibre en stratégie dominante** est un profil où tous les joueurs jouent une stratégie dominante.

Définition 8 *Soit $\tilde{s}_i$ la stratégie dominante pour le joueur i (lorsqu'elle existe). Alors le profil $(\tilde{s}_1, \tilde{s}_2, ..., \tilde{s}_n)$ est un équilibre en stratégies dominantes.*

Quand cet équilibre existe, nous sommes assuré qu'il est unique. De plus, il fournit une prédiction très claire sur l'issue du jeu. Malheureusement, ce type d'équilibre n'existe que pour très peu de jeu.

Une propriété interessante est que tout équilibre en stratégie dominante est aussi un équilibre de Nash (la réciproque est fausse).

2.3 Critère de Pareto et niveau de sécurité

2.3.1 Critère de Pareto

Définition 9 *On dit qu'un profil s* ***domine*** *un profil s' au* ***sens de Pareto*** *(ou* ***Pareto-domine****) si pour tout $s_i \in s$ pour tout $s'_i \in s'$ on a $s_i \geq s'_i$.*

Clairement, si dans un jeu on a plusieurs équilibres de Nash, on choisit un équilibre "préféré" en sélectionnant celui qui maximise les gains de tous les joueurs. Ce critère permet donc de résoudre les problèmes d'équilibre multiples dans certains jeux. Malheureusement, il ne résout pas ce problème dans tous les cas.

Joueur 1 / Joueur 2	S3	S4
S1	(4,4)	(3,1)
S2	(2,3)	(7,5)

TAB. 2.2 – Critère de Pareto : exemple 1

Dans l'exemple 2.2, il existe deux équilibres de Nash : (S1,S3) et (S2,S4). Le premier attribuant les gains (4,4) et le second (7,5). Dans ce cas on peut dire que le profil (S2,S4) Pareto-domine le profil (S1,S3) car pour les deux joueurs le résultat (S2,S4) est plus intéressant que le second.

Définition 10 *On dit qu'un profil s est un* ***optimum de Pareto*** *s'il n'existe pas un autre profil qui le Pareto-domine.*

Dans l'exemple ci dessus, (S2,S4) est un optimum de Pareto.

Définition 11 *On dit que deux profils s et s' ne sont pas Pareto-comparables si $\exists i$ tel que $\mu_i(s) > \mu_i(s')$ et $\exists j \neq i$ tel que $\mu_j(s) < \mu_j(s')$.*

Dans l'exemple de la guerre des sexes (tableau 2.1) les profils (O,O) et (F,F) ne sont pas Pareto-comparables.

Joueur 1 / Joueur 2	S3	S4
S1	(4,4)	(3,1)
S2	(2,3)	(7,3)

TAB. 2.3 – Critère de Pareto : exemple 2

Dans le jeu représenté dans le tableau 2.3 où l'on a juste modifié un gain par rapport à l'exemple précédent, (S1,S3) ne Pareto-domine plus (S2,S4), et donc on ne peut pas deviner quel profil sera préféré par les joueurs. Cet exemple simple montre que le critère de Pareto, bien que plus restrictif que l'équilibre de Nash, ne résout pas le problème de l'existence de plusieurs équilibres.

2.3.2 Niveau de sécurité

Le niveau de sécurité d'une stratégie s pour un joueur i est le gain minimum apporté par cette stratégie quelles que soient celles choisies par les autres joueurs.

Joueur 1 / Joueur 2	S3	S4
S1	(9,9)	(0,8)
S2	(8,0)	(7,7)

TAB. 2.4 – Niveau de sécurité

On pourrait en déduire que les joueurs ont tout intérêt à jouer les stratégies maximisant leurs niveaux de sécurité mais ce n'est pas toujours le cas.

Dans cet exemple donné dans le tableau 2.4, il existe deux équilibres de Nash : (S1,S3) et (S2,S4). Pour le joueur 1, la stratégie S1 apporte un niveau de sécurité de 0 et S2 apporte un niveau de 7 : les joueurs joueront donc (S2,S4). Le joueur 1 jouera donc S2. Pour le joueur 2, le raisonnement est identique, il aura le choix en S3 apportant un niveau de sécurité de 0 et S4 lui apportant un niveau de 7. Le joueur 2 jouera donc S4. Or il est clairement plus intéressant pour les deux joueurs de jouer (S1,S3) qui est le choix indiqué par l'optimum de Pareto (apportant 9 à chacun au lieu de 7).

Formellement,

Définition 12 *Le niveau de sécurité d'un stratégie s_i pour le joueur i est* $\min_{s_{-i}} \mu_i(s_i, s_{-i})$.

Donc, si les joueurs peuvent s'accorder, ils choisiront ensemble de jouer (S1,S3). Dans le cas contraire, il est trop risqué pour un joueur de jouer S1 ou S3 car il peut ne rien gagner. Ils choisiront de minimiser leurs risques en jouant (S2,S4) s'assurant un gain minimal de 7.

2.4 Fonction de meilleure réponse

Définition 13 ***La fonction de meilleure réponse*** *$R_i(s_{-i})$ associe à chaque combinaison de stratégies des autres joueurs s_{-i} une stratégie du joueur i qui maximise son gain.*

Sur l'exemple de la guerre des sexes (tableau 2.1) on a :

$$R_1(s_2) = \begin{cases} O & \text{si } s_2 = O \\ F & \text{si } s_2 = F \end{cases}$$

$$R_2(s_1) = \begin{cases} O & \text{si } s_1 = O \\ F & \text{si } s_1 = F \end{cases}$$

Cette fonction est étroitement liée à l'équilibre de Nash. En effet, si un profil s est un équilibre de Nash alors $R_i(s_{-i}) = s$.

2.5 Sous-jeux et équilibre parfait en sous jeux

Ces notions ont été introduites par Selten en 1975.

2.5.1 Menaces et promesses non crédibles

Menaces non crédibles

Définissons tout d'abord la notion de menace non crédible.

Soit le jeu sous forme extensive suivant :

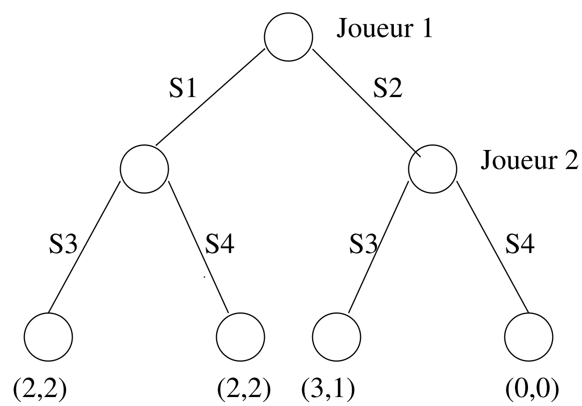

FIG. 2.1 – Exemple de menaces non crédibles

Dans cet exemple, il existe deux équilibres de Nash : (S1,S4) et (S2,S3). Cependant l'équilibre (S1,S4) repose sur la **menace non crédible** du joueur 2 de jouer S4. S3 est une stratégie dominante pour le joueur 2 et donc quelle que soit la décision du joueur 1, le joueur 2 a intérêt à jouer cette stratégie. La menace du joueur 2 de jouer S4 si le joueur 1 joue S2 pour le conduire à un gain de 0 n'est donc pas crédible. L'équilibre (S2,S3) sera donc "préféré" à (S1,S4).

Promesses non crédibles

De façon analogue, on peut définir la notion de promesse non crédible.

Considérons le jeu en forme extensive représenté figure 2.2.

Sur cet exemple, la promesse du joueur 2 de jouer S4 si le joueur 1 joue S2 n'est pas crédible. En effet, une fois que le joueur 1 a pris la décision de jouer S2, le joueur 2 n'a aucun intérêt à jouer S3. Il choisira donc de ne pas respecter sa promesse et de jouer S3. Le joueur 1 connaissant le jeu

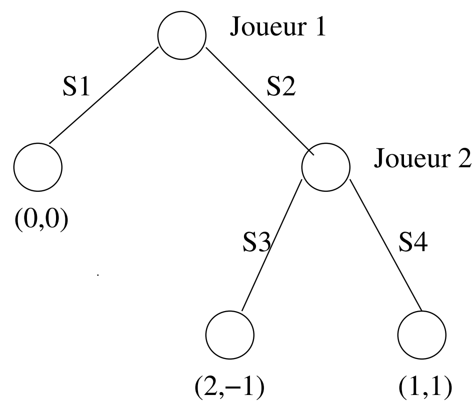

FIG. 2.2 – Exemple de promesses non crédibles

(s'il est à information complète) choisira donc de jouer S1. Ce résultat étant particulièrement mauvais pour les deux joueurs.

2.5.2 Sous-jeux

Avant de parler d'équilibre parfait en sous-jeux, nous devons encore définir ce qu'est un sous-jeu.

Un sous-jeu est l'ensemble formé par un noeud de décision du jeu original et tous les noeuds qui en découlent directement. Quand ce sous-jeu est différent du jeu original on l'appelle sous-jeu propre.

Considérons l'arbre de jeu représenté figure 2.3

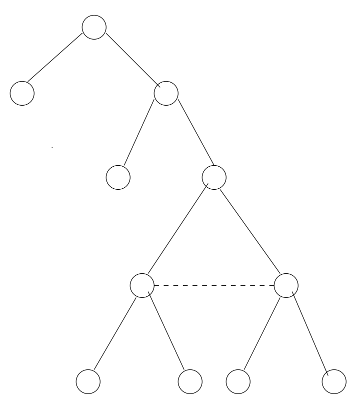

FIG. 2.3 – Un jeu sous forme extensive

Dans ce jeu, il y a trois sous-jeux. Le premier est le jeu lui-même. Les deux autres sous-jeux sont ceux représentés figures 2.4 et 2.5

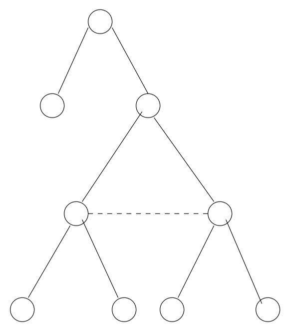

FIG. 2.4 – Premier sous-jeu

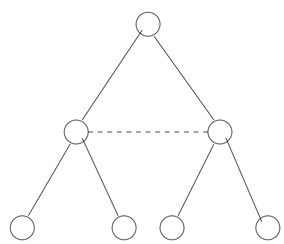

FIG. 2.5 – Second sous-jeu

On remarquera que la partie du jeu contenant l'ensemble d'information n'a pas été décomposé en sous-jeux. En effet on ne peut pas décomposer des noeuds reliés par un ensemble d'information.

2.5.3 Equilibres parfaits en sous-jeux

Une fois ces définitions établies, nous pouvons définir la notion d'équilibre parfait en sous-jeux. Ce concept vise à éliminer les équilibres qui reposent sur des menaces non crédibles qui ne seront jamais adoptées si le joueur est effectivement confronté à ce choix. En fait un résultat est un **équilibre parfait en sous-jeux** s'il correspond à un équilibre de Nash dans chaque sous-jeu. La réciproque est fausse, un équilibre de Nash du jeu original n'est pas nécessairement un équilibre parfait en sous-jeux.

2.6 Récurrence à rebours

La notion d'équilibre parfait en sous-jeux nous permet de définir une nouvelle notion : **la récurrence à rebours** (ou backward induction).

Soit le jeu sous forme extensive représenté figure 2.6.

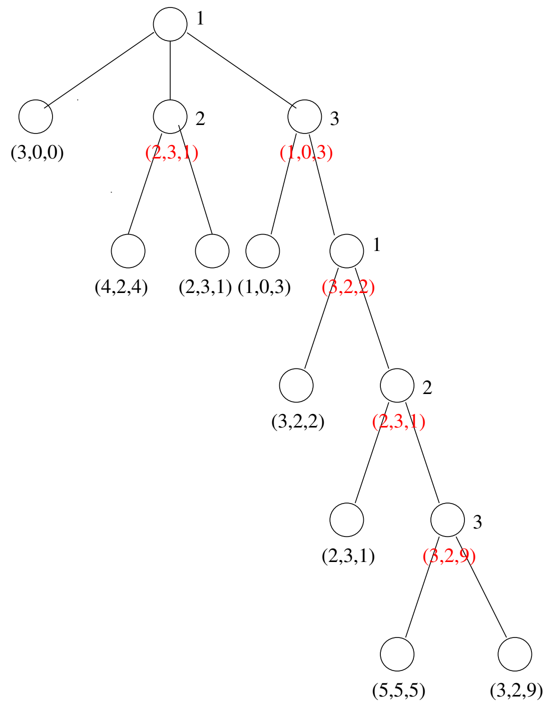

FIG. 2.6 – Récurrence à rebours

Comment prédire l'issue de ce jeu ?

Nous allons considérer récursivement les sous-jeux et "remonter" les valeurs jusqu'au noeud de départ. Les valeurs ainsi trouvées sont indiquées en rouge. Elles se déduisent simplement du sous-jeu en comparant les utilités sur les feuilles ou les valeurs "remontées". Par exemple, pour le dernier noeud, le joueur 3 va préférer l'alternative lui apportant une utilité de 9 que celle lui proposant 5. On place donc à ce noeud les utilités (3,2,9) et on considère maintenant ce noeud comme une feuille. Une fois sur le premier noeud, le joueur 1 a le choix entre trois alternatives lui apportant 3, 2 et 1 : il choisira donc l'alternative résultant sur (3,0,0). C'est donc le principe du niveau de sécurité qui est appliqué ici.

Ce processus décrit le principe du minimax (introduit par Von Neumann et Morgenstern) qui est la base de la résolution des jeux en intelligence artificielle (par un arbre de résolution). Ce principe est lié à la notion de niveau de sécurité et il est très souvent utilisé pour résoudre un jeu.

2.7 Stratégies pures et stratégies mixtes

2.7.1 Stratégies pures

On dit que le stratégie d'un joueur est **pure** si elle consiste à choisir toujours la même décision (c'est-à-dire la choisir avec une probabilité de 1).

Jusqu'alors nous avons toujours considéré des jeux en stratégies pures. Dans certains cas, considérer ces stratégies pose problème, par exemple pour la guerre des sexes (tableau 2.1). Avec ce que nous avons défini jusque-là, il nous est impossible de trouver une stratégie intéressante pour un joueur et encore moins de prédire l'issue du jeu. La solution peut être apportée en considérant le concept de stratégie mixte.

2.7.2 Stratégies mixtes

Les stratégies mixtes sont des distributions de probabilités sur les stratégies. Pour résoudre le jeu de la guerre des sexes, on peut imaginer que le joueur lance une pièce et décide alors de jouer F si elle tombe sur face et O sinon. Grâce à ce lancer le joueur augmente son utilité espérée.

En effet, on a alors une chance sur deux de choisir F et une chance sur deux de choisir O. Donc :

$$\mu_1(F_{\frac{1}{2}}, F) = \tfrac{1}{2} \times 2 + \tfrac{1}{2} \times 0 = 1$$
$$\mu_1(F_{\frac{1}{2}}, C) = \tfrac{1}{2} \times 0 + \tfrac{1}{2} \times 1 = \tfrac{1}{2}$$

Dans ce cas, l'utilité espérée du joueur est de $\frac{1}{2}$ au lieu de 0. Evidemment, dans cet exemple, ce résultat a réellement un sens uniquement si le jeu est répété un certain nombre de fois.

Définition 14 *On note $\sum_i$ l'ensemble des stratégies mixtes d'un joueur. Une stratégie mixte de cet ensemble est noté σ_i. Soit $p_i(s_k)$ la probabilité associée à s_k par σ_i. On définit alors l'utilité d'un profil en stratégies mixtes par :*

$$\mu_i(\sigma) = \sum_{s \in S} (\prod_{j=1}^{n} p_j(s_j))\mu_i(s)$$

Nash a montré qu'il existe toujours un équilibre en stratégies mixtes. Cet équilibre se définit de la même façon que pour les stratégies pures. En effet, les stratégies pures sont un cas particulier des stratégies mixtes.

Dans ce chapitre, nous avons présenté les différentes notions permettant de résoudre un jeu. Au cours de cette présentation, nous avons vu qu'aucune de ces méthodes n'étaient complète. En effet, aucune ne permet de résoudre un jeu dans le cas général.

Il nous faut donc trouver une autre méthode pour découvrir les stratégies que les joueurs devront utiliser. Une technique envisageable est l'approche par simulations.

Chapitre 3

Recherche de bonnes stratégies : approche par simulations

Le chapitre précédent a mis en évidence l'absence de meilleure stratégie pour un joueur donné général (c'est-à-dire lorsqu'il ne possède pas de stratégie dominante). Dans ce cas de figure, il est intéressant de comparer empyriquement les stratégies entre elles. Pour cela nous nous appuierons sur un jeu inventé par Flood et Drescher en 1952 : **le dilemme du prisonnier**. Nous allons d'abord voir sur cet exemple, ce que les notions définies dans le chapitre 2 nous permettent de connaître sur ce jeu. Nous introduirons dans cette partie le concept de jeu itéré. Nous verrons que dans un jeu itéré, on ne déduit pas les mêmes conclusions que s'il ne l'était pas.

Ce chapitre n'a absolument pas la prétention de présenter exhaustivement les recherches effectuées sur le dilemme du prisonnier. Le lecteur pourra trouver plus d'informations dans [10].

3.1 Le dilemme du prisonnier classique

3.1.1 Définition

Le dilemme des prisonniers est un jeu à deux joueurs. Supposons que deux personnes (coupables) soient arrêtées par la police. Nos deux prisonniers ont deux possibilités : nier ou coopérer. Si les deux nient, ils écopent tous deux d'un an de prison. S'ils coopèrent tous les deux, ils écopent de 3 ans de prison. S'il l'un nie et l'autre coopère (il trahit son compère), celui qui coopère est relâché et l'autre écope de 5 ans de prison. Plus généralement, on dit que si les deux nient ils obtiennent tous deux un gain R, s'ils coopèrent, un gain P et si l'un trahit l'autre, celui qui trahit recoit S et l'autre recoit T. Pour être en présence du dilemme du prisonnier, la condition $S < P < R < T$ doit être respectée. Nous pouvons reformuler les stratégies : Coopérer = Nier aux policiers (coopérer avec l'autre joueur) : on notera cette stratégie C (Cooperate) et Trahir = Avouer aux policiers (trahir l'autre joueur) : on notera cette stratégie D (Defect).

Ce jeu est à information imparfaite : les deux prisonniers ne sont pas au courant de la décision prise par l'autre.

Sous forme normale, ce jeu est représenté par le tableau 3.1

Joueur 1 / Joueur 2	Coopérer (C)	Trahir (D)
Coopérer (C)	(R,R)	(S,T)
Trahir (D)	(T,S)	(P,P)

TAB. 3.1 – Le dilemme du prisonnier sous forme normale

et sous forme extensive par la figure 3.1.

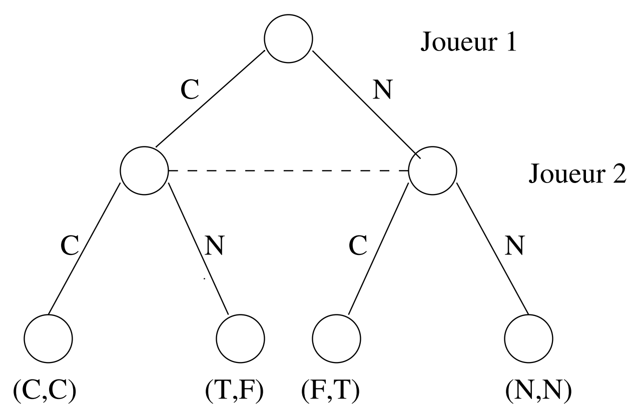

FIG. 3.1 – Le dilemme du prisonnier sous forme extensive

3.1.2 Résolution

Ce jeu possède clairement un équilibre de Nash. Il s'agit de (D,D) apportant un gain de P à chacun des joueurs. (C,C) Pareto-domine (D,D) (car $S < P < R < T$) mais (D,D) maximise le niveau de sécurité des joueurs. Comme les joueurs ne peuvent pas communiquer, ils choisiront de maximiser chacun leurs niveaux de sécurité et choisiront donc de jouer D tous les deux.

3.2 Les jeux itérés

Quand les joueurs jouent le même jeu plusieurs fois de suite, on dit que le jeu est **itéré** (ou répété). Contrairement aux jeux joués une seule fois, les jeux itérés permettent aux stratégies de prendre en compte les coups passés, ce qui autorise les effets de réputation et de rétribution. D'une certaine facon les joueurs vont pouvoir "coopérer". Un joueur pourrait jouer une stratégie à l'avantage de son adversaire si en échange celui-ci joue une qui l'avantage à son tour au coup suivant (rétribution) ou une stratégie désavantageuse pour son adversaire si celui-ci avait joué une stratégie désavantageuse pour lui au coup précédent (notion de réputation). Ces stratégies peuvent donc être qualifiées de "réactive" c'est-à-dire du type : si le(s) coup(s) précédent(s) de mon adversaire était(ent) la stratégie s alors je joue la stratégie s'. Avec ce type de stratégie, ce sont les suites d'événements passés dans le jeu qui déclenchent le choix des coups à jouer par le joueur : la stratégie des joueurs "réagit". Nous pouvons alors introduire des stratégies plus ou moins sévères envers l'adversaire, plus ou moins "coopérative".

Un "problème" apparait rapidement : celui de la dernière itération du jeu. En effet, le dernier tour du jeu peut être vu comme un jeu sans répétition puisqu'il n'a pas de coup à jouer après celui ci. Si une stratégie dominante existe pour un joueur, alors il jouera cette stratégie indépendamment de ce qui s'est passé précédemment dans le jeu. Puisque les joueurs connaissent l'issue du dernier tour, ils peuvent l'ignorer à l'avant dernière itération. Ils joueront donc cette itération comme si c'est la dernière et joueront à nouveau leurs meilleures stratégies. On voit clairement apparaître un phénomène d'induction qui va nous ramener jusqu'à la première itération du jeu. L'itération du jeu n'a donc pas d'intérêt réel si les joueurs connaissent le nombre d'itérations qu'ils vont devoir jouer. Pour étudier les stratégies dans un jeu itéré, les joueurs doivent donc ignorer le nombre d'itérations qu'il comporte.

3.3 Résolution du dilemme itéré du prisonnier

On définit le dilemme itéré du prisonnier avec les mêmes règles, les mêmes alternatives disponibles pour les joueurs mais le jeu est joué plusieurs fois. Ce nombre n'est pas connu des joueurs. Pour que l'étude soit plus intéressante on ajoute la condition $S+T < 2R$ qui permet de favoriser la coopération. Les joueurs accumulent l'utilité gagnée à chaque itération. Le gagnant est celui qui a le gain total le plus important à la fin du jeu. Résoudre ce dilemme revient à trouver une stratégie meilleure que toutes les autres dans le sens où elle gagne toujours contre toutes les autres et permet donc au joueur qui la joue de toujours gagner le jeu en obtenant toujours plus de gain que son adversaire à la fin du jeu, et cela quel que soit le nombre d'itérations après lequel le jeu s'arrête.

3.3.1 Définition des stratégies

Nous devons donc chercher des stratégies qui gagnent contre toutes les autres. Définissons quelques stratégies dont nous allons nous servir par la suite comme éléments de comparaison.

Commencons par les stratégies les plus simples :

- (1) gentille : coopère toujours.
- (2) méchante : trahit toujours.
- (3) lunatique : choisit au hasard (avec un tirage à pile ou face) le coup à jouer.

Définissons maintenant des stratégies jouant alternativement C et D :

- (4) per_CCD : joue successivement C puis C puis D puis recommence un nouveau cycle CCD.
- (5) per_DDC : joue successivement D puis D puis C.

Enfin introduisons des stratégies considérant les coups joués précédemment :

- (6) rancunière : joue toujours C mais dès que l'adversaire a trahi une fois, elle trahit toujours.
- (7) donnant-donnant : joue C au premier tour puis toujours le même coup que celui de l'adversaire à l'itération précédente.
- (8) donnant-donnant_dur : comme donnant-donnant mais joue D au premier tour.
- (9) majoritaire_dur : joue le coup le plus souvent joué par l'adversaire et trahit en cas d'égalité. Le premier coup est considéré comme une égalité.
- (10) majoritaire_gentille : comme majoritaire_dur mais coopère en cas d'égalité.
- (11) graduelle : joue toujours C mais trahit une fois quand l'adversaire trahit la première fois, puis trahit deux fois de suite quand l'adversaire trahit pour la seconde fois, etc...
- (12) sondeur : joue D, C puis C aux trois premier coups, si les deux derniers coups de l'adversaires étaient C alors la stratégie trahit toujours sinon elle joue donnant-donnant.

Une fois ces stratégies définies, il nous faut les comparer pour trouver la meilleure.

3.3.2 Comparaison des stratégies

Tournois

Nous pouvons tout d'abord imaginer de se faire rencontrer toutes les stratégies l'une contre l'autre (y compris le cas où elle se rencontre elle même). Sur 1000 itérations, les stratégies produisent les résultats présentés dans le tableau 3.2 :

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(12)
(1)	3000	0	1519	2001	999	3000	3000	2997	2997	3000	6
(2)	5000	1000	3017	3668	2332	1004	1004	1000	1000	1004	1008
(3)	3987	495	2226	2831	1675	500	2268	2237	1936	2078	2012
(4)	3666	333	1999	2334	1665	343	2667	2664	3663	3666	2664
(5)	4334	667	2470	3335	1666	671	2003	1999	667	671	2006
(6)	3000	999	3024	3663	2331	3000	3000	1003	1000	3000	1007
(7)	3000	999	2264	2667	1998	3000	3000	2500	2500	3000	2999
(8)	3002	1000	2239	2669	1999	1003	2500	1000	1000	2500	3000
(9)	3002	1000	2385	2003	2332	1003	2500	1000	1000	2500	2501
(10)	3000	999	2319	2001	2331	3000	3000	2500	2500	3000	2999
(12)	4996	998	2332	2669	1996	1002	2999	2995	2496	2999	1004

TAB. 3.2 – Tournoi de 11 stratégies

En faisant la somme des gains obtenus par chaque stratégie on déduit le classement suivant :

1. donnant-donnant
2. majoritaire_gentille
3. rancunière
4. sondeur
5. per_CCD
6. gentille
7. lunatique
8. méfiante
9. majoritaire_méchante
10. méchante
11. per_DDC

Si on qualifie de "gentilles" les stratégies ne prenant jamais l'initiative de trahir et de "méchantes" celles qui peuvent trahir sans que l'adversaire ait nécessairement trahi auparavant, on remarque que, dans l'ensemble, les stratégies "gentilles" donnent de meilleures résultats.

On peut remarquer également que la stratégie donnant-donnant arrive en tête bien qu'elle ne gagne jamais (dans le sens où elle finit le jeu avec moins de points que la stratégie adverse) ! Mais au pire elle ne perd que de 5 points quel que soit la stratégie opposée. Notons que ce résultat est correct car le but d'un joueur est toujours de maximiser ses gains. Donc donnant-donnant est une bonne stratégie.

Ce type de comparaison de stratégies est-il réellement pertinent ? La réponse est non. En effet, les stratégies ne font que de se rencontrer les unes contre les autres et cela ne révèle pas la capacité d'une stratégie dans un environnement composé de plusieurs autres. Pour ce faire nous allons devoir considérer une autre méthode de comparaison de stratégie plus robuste : l'évolution écologique.

Evolution écologique

L'évolution écologique comme méthode de comparaison est basée sur les algorithmes évolutionnaires. Nous représentons chaque stratégie par une population de n individus. Si on compare m stratégies, on a donc une population de départ de $m \times n$ individus. Un algorithme évolutionnaire permet, partant d'une population de départ représentant des entités, de progressivement isoler les meilleurs individus (et donc les meilleures stratégies) en supprimant les plus faibles. Ce processus est appelé sélection naturelle. Dans ce cas, à chaque itération (appelé génération) de l'algorithme les stratégies vont s'affronter deux à deux. A la suite de cela, les stratégies ayant les scores les plus faibles vont être éliminées de la population par sélection naturelle selon un certain taux et les plus fortes vont demeurer. Après un certain nombre d'itérations, la population se stabilise et seuls les meilleurs individus y demeurent.

Cette comparaison nous permet de situer le comportement des stratégies au fil du temps et de déduire les plus efficaces contre toutes les autres. Le résultat est représenté par le graphe figure 3.2

Dans ce cas, le classement est :

1. donnant-donnant
2. majoritaire_gentille
3. rancunière
4. gentille
5. méchante

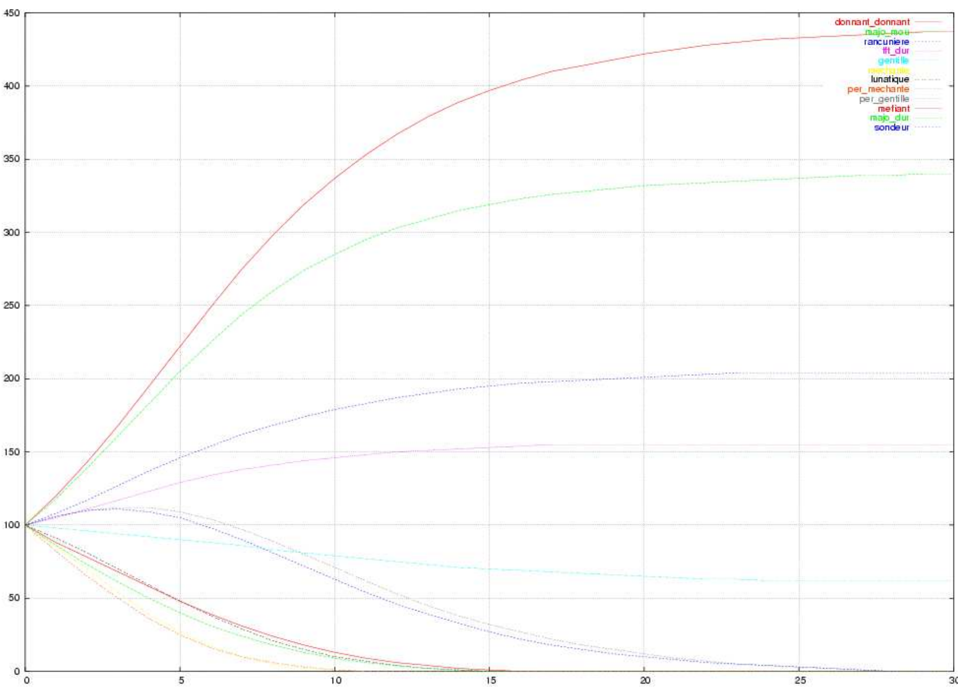

FIG. 3.2 – Comparaison avec évolution écologique

6. lunatique
7. per_DDC
8. per_CCD
9. méfiante
10. majoritaire_dur
11. sondeur

Ce type de comparaison fournit également les meilleures stratégies mais donnant-donnant est toujours en tête, bien qu'on ait montré qu'elle ne gagnait jamais.

Nous pouvons encore critiquer cette comparaison car certaines stratégies nourissent d'autres stratégies. En effet, les stratégies méchantes gagnent beaucoup au début des comparaisons et donc leurs populations vont s'agrandir rapidement. Par la suite, les stratégies gentilles, via les coopérations vont reprendre le dessus. Donc, certaines stratégies ont un classement qui dépend du type de stratégies auxquelles elles sont opposées.

Il existe des variantes de l'évolution écologique : on peut par exemple considérer les évolutions écologiques sur 11 stratégies parmi 12 puis 10 parmi 12,... Ainsi, nous verrons quelles stratégies "se nourissent" des autres. Nous pouvons faire varier la durée, c'est-à-dire arrêter l'évolution avant la stabilisation de la population, pour constater quelles stratégies prennent le dessus au début de la comparaison.

Axelrod, en 1981, a définit les critères de bonnes stratégies (au sens de l'évolution) suivants :

- gentillesse
- réactivité
- pardon
- simplicité

La simplicité est un critère discutable (Beaufils et al., 1996).

3.3.3 Recherche des stratégies

Nous devons trouver une méthode permettant de trouver les meilleures stratégies et qui nous fournit un résultat en un temps fini qui code clairement le comportement adopté par celle ci. Il est possible d'automatiser la recherche de bonnes stratégies, toujours grâce aux algorithmes évolutionnaires. Par exemple, considérons la classe des stratégies se souvenant de leur dernier coup et du dernier coup de l'adversaire, alors le génotype de cette stratégie sera composé de 5 gènes. Soit la stratégie suivante :

Jouer C au premier coup (comportement codé dans le gène 1). Si au coup précédent j'ai joué C et que l'adversaire a joué C alors je joue C (comportement codé dans le gène 2). Si au coup précédent j'ai joué C et que l'adversaire a joué D alors je joue D (comportement codé dans le gène 3). Si au coup précédent j'ai joué D et que l'adversaire a joué C alors je joue C (comportement codé dans le gène 4). Si au coup précédent j'ai joué D et que l'adversaire a joué D alors je joue D (comportement codé dans le gène 5).

Le génotype sera celui représenté figure 3.3

C	C	D	C	D

FIG. 3.3 – Génotype de donnant-donnant

Soit M le nombre de coups qu'une stratégie peut voir de son passé et O le nombre de coups du passé de l'adversaire. Par exemple, pour M=0 et O=1, on cherche la meilleure stratégie parmi celles qui ne se souviennent que du dernier coup de l'adversaire. Nous allons contruire une population constituée d'individus représentant les 8 stratégies possibles (car il y a 3 gènes pour coder cette stratégie) et procéder à une comparaison par évolution écologique.

Ce qui donne le résultat figure 3.4 :

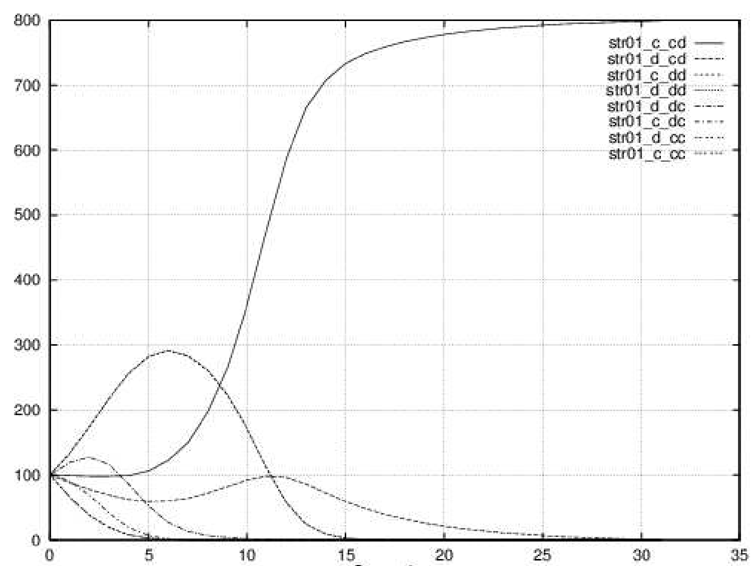

FIG. 3.4 – Recherche de stratégies grâce à l'évolution écologique

On montre ainsi que la meilleure stratégie quand on considère les stratégies qui se souviennent uniquement du dernier coup de l'adversaire, est donnant-donnant.

Dans ce chapitre, nous avons décrit une approche par simulations pour trouver les bonnes stratégies dans un jeu itéré à deux joueurs. Nous avons vu que les résultats obtenus sont pertinents mais que nous ne pouvons pas montrer qu'il n'existe pas d'autres stratégies meilleures que celles obtenues par les simulations.

Nous allons maintenant nous rapprocher du sujet en abordant les jeux à n joueurs.

Chapitre 4

Jeux avec coalitions

4.1 Introduction

Les jeux avec coalitions sont nécessairement des jeux à plus de deux joueurs. Le fait que les joueurs puissent s'allier contre d'autres afin de former des **coalitions** augmente la complexité du jeu mais aussi son intérêt. D'un point de vue économique et vie sociale, ces jeux représentent des situations qu'on retrouve fréquemment et donc pour lesquels les applications sont importantes. Les joueurs peuvent donc éventuellement communiquer et s'accorder sur des coalitions à former afin de maximiser leurs gains. En effet, dans la plupart des jeux à n joueurs, il est possible pour les joueurs d'augmenter l'utilité attendue en collaborant avec d'autres joueurs. Cela dit, dans la plupart des jeux, l'utilité gagnée devra être redistribuée aux membres de la coalitions.

Dans cette partie, nous allons tout d'abord introduire la notion de système multi-agents et effectuer une comparaison avec les jeux à deux joueurs. Nous mettrons ainsi en évidence les motivations et les nouveaux problèmes engendrés par l'introduction de plus de deux joueurs. En particulier, nous montrerons les problèmes de représentation des jeux, de formation des coalitions, de recherches des meilleures stratégies.

4.2 Jeux à n joueurs

4.2.1 Systèmes multi-agents

Introduisons brièvement quelques notions de base sur les systèmes multi-agents. En effet, les joueurs vont pouvoir se comporter de la même façon, il peut donc être intéressant de connaître ces bases.

Un agent est une entité autonome située dans un environnement et capable de communiquer avec lui par le biais de messages et d'agir sur lui. Il peut envoyer ces messages à un autre agent en particulier à un groupe d'agents ou à tous les agents. Ces messages sont généralement des requêtes, les autres agents sont libres d'accepter ou de refuser ce qui leur a été proposé. Dans le cas où plusieurs agents forment un groupe afin de travailler ensemble pour regrouper les capacités de chacun par exemple, on nomme ce groupe système multi-agents. Les agents forment des systèmes afin de pouvoir atteindre un objectif de façon plus efficace que s'ils travaillaient indépendamment les uns des autres. Les agents vont donc se regrouper au mieux dans des systèmes pour coopérer et réaliser un but que l'on peut voir comme une maximisation d'une fonction objectif. Au sein d'un système, il existe des processus de négociation, permettant aux agents de se mettre en accord sur les tâches à accomplir et les éventuels mouvements d'agents entre les systèmes. Dans ce cadre, les agents sont libres de changer de système quand ils le souhaitent ; par exemple, si un autre système répond mieux aux attentes d'un agent ou si les capacités de l'agent sont intéressantes pour ce système, l'agent pourra quitter celui auquel il appartient pour se joindre à l'autre.

Le parallèle avec les jeux à n joueurs est évident : les joueurs vont coopérer au sein de coalition afin de maximiser leurs gains. Mais ils restent libres de négocier entre eux et de changer de coalition si cela est dans leur intérêt. Dans le cadre de la théorie des jeux, les joueurs sont rationnels et leur objectif est donc de maximiser leurs gains.

4.2.2 Comparaison avec les jeux à deux joueurs

D'un point de vue pratique, on ne peut pas restreindre la théorie des jeux aux jeux à deux joueurs. Les jeux à n joueurs représentent un champ d'application important mais bien plus complexe à étudier que la restriction à deux joueurs. Les jeux à n joueurs conservent les plus importantes notions qui ont été définies plus haut pour les jeux à 2 joueurs : l'équilibre et les stratégies mixtes. Comme nous l'avons déjà précisé plus haut, le fait qu'il y ait n participants au jeu permet la formation de coalitions. Cette notion est une différence marquante par rapport aux jeux à deux joueurs. Un premier problème est qu'il n'existe pas dans la théorie de formalisme définissant comment se fait la communication entre les joueurs afin de former les coalitions. Un second est d'ordre pratique : le nombre de coalitions pouvant être formées par n joueurs est de l'ordre de 2^n et donc générer toutes les coalitions pour les comparer est un problème dont le temps de calcul pour le résoudre est exponentiel. De plus, les stratégies de jeu doivent inclure le choix de la coalition à rallier, à quitter ou éventuellement en créer une et les négociations à entamer avec d'autres joueurs ou coalition. A chaque fois qu'il devra jouer, le joueur aura donc le choix entre une quantité considérable de choix à sa disposition. Nous devons donc trouver un autre moyen de mettre en évidence les meilleures coalitions et les meilleures stratégies et définir une structure permettant de travailler avec n joueurs. Pour plus d'informations, le lecteur pourra consulter [24, chapitre 7].

4.2.3 Représentation

Il est clair qu'on ne peut plus représenter de facon aussi simple les jeux à n joueurs qu'à deux joueurs. En effet, pour la représentation en forme normale, il faut créer une matrice de dimension n (où n désigne le nombre de joueurs) avec, pour chaque dimension, autant de lignes et de colonnes qu'il y a de stratégies. Pour trois joueurs, nous devons considérer un hypercube de dimension 3, ce qui est déjà très contraignant. Sous forme extensive, le probléme est similaire. Le nombre d'arcs issus d'un noeud est égal au nombre de stratégies disponibles pour un joueur, ce qui peut être gigantesque. Nous allons donc devoir rester dans un cadre très formel.

Formellement, la représentation d'un jeu à n joueurs s'étend naturellement du cas où n=2.

Définition 15 *Un jeu à n joueurs est représenté de la façon suivante :*
- *un ensemble N de n joueurs.*
- *n ensembles de stratégies $S_1, S_2, ..., S_n$*
- *n fonctions de gain à valeur réelle $\mu_1, \mu_2, ..., \mu_n$ où $\mu_i(s_1, s_2, ..., s_n)$ est l'utilité du joueur i si le joueur 1 joue s_1, 2 joue s_2,..., n joue s_n.*

Les stratégies mixtes (rappellons que les stratégies pures sont un cas particulier des stratégies mixtes) sont définies exactement de la même façon (définition 14).

Nous pouvons également étendre le concept de jeux à somme nulle pour n joueurs. Un jeu à n joueur est à somme nulle si la somme des utilités associées à chaque tuple de stratégies est 0. Formellement, la fonction μ_i peut être choisie telle que pour tout tuple de stratégies $(s_1, s_2, ..., s_n)$ on a $\sum_{i=1}^{n} \mu_i(s_1, s_2, ..., s_n) = 0$.

4.2.4 Equilibres dans les jeux à n joueurs non coopératifs

Dans ce paragraphe, nous supposerons que les joueurs ne peuvent pas communiquer et qu'aucune coalition ne peut se former.

Soit M_i la fonction de gain du joueur i et $(s_1, s_2, ..., s_n)$ un profil de stratégies à n joueurs. Ces stratégies sont en **équilibre** si aucun des joueurs n'a d'interêt à changer de stratégie si les autres joueurs ne changent pas les leurs.

Définition 16 $(s_1, s_2, ..., s_n)$ *est un point d'équilibre en stratégies pures si pour tout joueur i* $M_i(s_1, s_2, ..., s_i, ..., s_n) \geq M_i(s_1, s_2, ..., r_i, ..., s_n)$ *pour toute stratégie* $r_i \in S_i$ *où* S_i *désigne l'ensemble des stratégies disponibles pour le joueur i.*

Tout comme pour les jeux à deux joueurs cet équilibre n'existe pas toujours ou il peut en exister plusieurs. On définit de la même facon l'équilibre en stratégies mixtes.

Définition 17 $(\sigma_1, \sigma_2, ..., \sigma_n)$ *est un point d'équilibres en stratégies mixtes si pour tout joueur i* $M_i(\sigma_1, \sigma_2, ..., \sigma_i, ..., \sigma_n) \geq M_i(\sigma_1, \sigma_2, ..., \sigma'_i, ..., \sigma_n)$ *pour toute stratégie* $\sigma'_i \in S_i$ *où* S_i *désigne l'ensemble des stratégies disponibles pour le joueur i.*

Dans le cas des jeux à n joueurs, Nash a également montré qu'il existe toujours au moins un équilibre en stratégies mixtes. Le lecteur pourra trouver la démonstration dans [24, Annexe 2]. Ces résultats montrent, comme dans le cas des jeux à deux joueurs, que la notion d'équilibre ne permet pas de résoudre tous les jeux.

4.2.5 Stratégies

Pour limiter le domaines des stratégies, on peut réduire la notion de stratégie. Plutôt que de donner des stratégies mixtes sur l'ensemble des alternatives, un joueur peut indiquer pour chacun de ses ensembles d'information une distribution de probabilités sur les alternatives de l'ensemble. Un telle classe de distribution est appelée **stratégie comportementale**. On peut montrer que pour chaque stratégie mixte on peut associer une unique stratégie comportementale. Réciproquement, à chaque stratégie comportementale on peut associer au moins une stratégie mixte. Pour illustrer la différence entre les stratégies mixtes et les stratégies comportementales, considérons le jeu représenté à la figure 4.1.

On pose $\alpha_1 = (S1, S3), \alpha_2 = (S1, S4), \alpha_3 = (S2, S3), \alpha_4 = (S2, S4)$.

Considérons les deux stratégies mixtes suivantes :

$\sigma'_1 = (\frac{1}{2}\alpha_1, 0\alpha_2, 0\alpha_3, \frac{1}{2}\alpha_4)$ et $\sigma''_1 = (\frac{1}{4}\alpha_1, \frac{1}{4}\alpha_2, \frac{1}{4}\alpha_3, \frac{1}{4}\alpha_4)$.

Ces deux stratégies mixtes induisent la même stratégie comportementale. Le **comportement** du joueur 1 sera : jouer S1 ou S2 en premier avec une probabilité de $\frac{1}{2}$ puis S3 ou S4 avec une probabilité de $\frac{1}{2}$ au second coup.

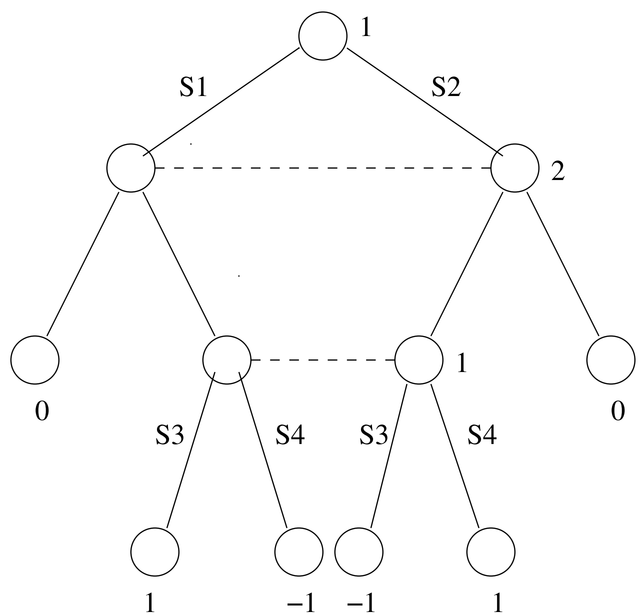

FIG. 4.1 – Comparaison : stratégies mixtes et comportementales

On dit que deux tuples de stratégies mixtes $(\sigma'_1, \sigma'_2, ..., \sigma'_n)$ et $(\sigma''_1, \sigma''_2, ..., \sigma''_n)$ sont comportementalement équivalentes si σ'_i et σ''_i induisent la même stratégie comportementale pour tout i.

Ces stratégies sont intéressantes mais elles supposent que le joueur sache les coups qu'il a effectués précédemment, c'est-à-dire que le jeu soit à information parfaite.

Il serait intéressant d'avoir une propriété entre les stratégies mixtes et comportementales plus forte que l'équivalence comportementale. C'est ce qui a été apporté par Thompson et Kuhn.

Définition 18 *On dit qu'un ensemble d'information U d'un joueur i est un* ***ensemble d'information signalant*** *s'il existe d'autres ensembles d'information, appelé V, plus bas dans l'arbre du jeu qui appartiennent aussi au joueur i et une branche r laissant l'ensemble d'information U tel que :*

- *Il existe au moins un mouvement dans V qui peut être atteint par un chemin partant de la rième branche d'un mouvement dans U.*
- *Il existe au moins un mouvement dans V qui ne peut pas être atteint par aucun chemin partant de la rième branche d'un mouvement dans U.*

Cela signifie que si U est un ensemble d'information signalant pour le joueur i, alors quand ce joueur se trouve à un ensemble d'information V, il est impossible de savoir s'il a choisi ou non la rième alternative à U.

Définition 19 *Un jeu ne possédant pas de tel ensemble d'information est dit à **mémoire parfaite**. Un jeu à information parfaite a une mémoire parfaite mais la réciproque est fausse.*

Kuhn montra que si un jeu a une mémoire parfait alors si $(\beta_1, \beta_2, ..., \beta_n)$ sont des stratégies comportementales induites par les stratégies mixtes - $(\sigma_1, \sigma_2, ..., \sigma_n)$ alors pour chaque joueur i $M_i(\beta_1, \beta_2, ..., \beta_n) = M_i(\sigma_1, \sigma_2, ..., \sigma_n)$.

Peut-on extraire également certaines propriétés pour les jeux sans mémoire parfaite ?

Définition 20 *Notons Δ_i l'ensemble des ensembles d'information signalants pour un joueur i. On appele **stratégies pures signalantes** pour un joueur i les stratégies définies sur l'ensemble Δ_i et stratégies comportementales associées, les stratégies comportementales sur les ensembles d'informations de ce joueur qui ne sont pas signalants.*

Définition 21 ***Une stratégie composite*** *est une paire constituée d'une stratégie mixte signalante et d'une stratégie comportementale associée. Chaque stratégie mixte induit une unique stratégie composite une stratégie composite induit au moins une stratégie mixte.*

Tout comme les stratégies comportementales, les utilités pour le joueur i dans un jeu à mémoire parfaite sont égales à celles des stratégies mixtes.

4.2.6 Coalitions admissibles

Comme nous l'avons vu, le nombre de coalitions pouvant être formées est exponentiel en le nombre de joueurs. D'un point de vue pratique il faut restreindre l'univers des coalitions. Les coalitions sont représentées par une partition notée τ de l'ensemble des joueurs. Par exemple, $(\{1,2,3\}, \{4, ..., n-2\}, \{n-1, n\})$ représente un jeu avec trois coalitions respectivement formées des joueurs (1,2,3), (4,...,n-2) et (n-1,n). Chaque joueur devra considérer les changements potentiels d'alliances induit par sa prise de décision.

On peut imaginer que dans le jeu toutes les coalitions ne sont pas permises, ou encore qu'un joueur ne peut pas joindre, quitter ou créer une coalition comme il le souhaite.

Définition 22 ***Une règle de changement de coalition admissible****, notée ψ, indique pour chaque structure de coalition τ les coalitions des joueurs qui possèdent la liberté de communication avant un coup sans restriction et peuvent agir conjointement. La classe de ce type de coalitions admissibles selon la règle ψ pour la structure τ est notée $\psi(\tau)$.*

En fait, il s'agit d'une règle qui indique quelles coalitions sont autorisées ou non. Par exemple, considérons un jeu à trois joueurs tel que la structure de coalition soit $(\{1,2\},\{3\})$ et la règle ψ stipule que les seuls changements admissibles sont les coalitions formées par ajout d'au plus un joueur dans une coalition existante. $\psi(\tau)$ nous donne donc les coalitions pouvant être formées avec cette règle et est égal à $(\{1,2\},\{3\},\{1,2,3\},\{1,2\},\{2,3\})$. Remarquons que cette notion est très souple : elle permet d'autoriser toutes les coalitions possibles ou aucune.

4.3 Jeux sous forme coalitionelle

4.3.1 Définition

Un jeu sous forme coalitionelle est caractérisé par une fonction de coalition. Ce type de jeu est également appelé jeu sous forme de fonction caractéristique. On définit une fonction de coalition comme une fonction v qui associe une valeur $v(S)$ à chaque coalition S possible. Si on note N l'ensemble des n joueurs et 2^N l'ensemble des 2^n coalitions possibles, $v : 2^N \rightarrow R$. v(S) indique le gain total minimal que les joueurs de S peuvent avoir conjointement (on appelle aussi cette valeur la force ou le pouvoir de la coalition). Cette fonction est souvent appelée fonction caractéristique d'un jeu sous forme coalitionnelle et ce jeu est souvent noté (N,v).

Définition 23 *La forme coalitionnelle d'un jeu à n joueurs est donnée par le couple (N,v) où N est l'ensemble des n joueurs et v est la fonction caractéristique du jeu définie pour chaque coalition S. Généralement, on considère que cette fonction doit respecter les conditions :*

- *(i)* $v(\emptyset) = 0$
- *(ii) si S et T sont deux coalitions disjointes alors* $v(S)+v(T) \leq v(S \cup T)$

On appelle communément la coalition vide, la coalition formée par l'ensemble vide $\emptyset$, et la grande coalition celle formée par l'ensemble des n joueurs. On désigne souvent par le jeu v le jeu défini par la fonction caractéristique v.

Maintenant se pose la question de la méthode de définition de la valeur de v pour chaque coalition. Une façon usuelle de déterminer la valeur de la fonction caractéristique pour une coalition S est de considérer le jeu à deux joueurs à somme nulle formé par la coalition S agissant comme un joueur opposé à $\bar{S} = N - S$. On assigne alors $v(S)$ avec la valeur de ce jeu (symbole Val). Nous avons donc :

$$v(S) = Val(\sum_{i \in S} \mu_i(x_1, ..., x_n))$$

où $\mu_i(x_1, ..., x_n)$ désigne le gain de joueur i lorsque le joueur 1 joue x_1, le joueur 2 joue x_2, etc...

La valeur $v(S)$ est donc analogue au niveau de sécurité. En effet, elle représente le gain minimum de S même si elle est opposée à $\bar{S}$ (qui est la plus forte coalition qui peut se former contre S). On remarque que ce type d'assignement assure une propriété définie plus loin : la superadditivité.

Tout jeu à n joueurs peut être réduit à un jeu sous forme coalitionnelle. Cependant, cette réduction perd des propriétés importantes du jeu comme les menaces.

Pour montrer la méthode de transformation d'un jeu à n joueurs sous forme stratégique, nous allons considérer l'exemple de jeu à trois joueurs sous forme normale donné par le tableau 4.1 :

Si 1 joue S1 :

		3	
		S1	S2
2	S1	(0,3,1)	(2,1,1)
	S2	(4,2,3	(1,0,0)

Si 1 joue S2 :

		3	
		S1	S2
2	S1	(1,0,0)	(1,1,1)
	S2	(0,0,1)	(0,1,1)

TAB. 4.1 – Un jeu à trois joueurs sous forme normale

Tout d'abord, on a automatiquement $v(\emptyset) = 0$. Pour trouver $v(N)$, considérons la coalition des n joueurs contre la coalition vide. Ensemble, les trois joueurs pourront jouer (S1,S2,S1) ce qui rapportera 4+2+3=9 (l'adversaire n'a pas d'alternative disponible : c'est la coalition vide). Donc $v(N) = 9$.

		(2,3)			
		S1,S1	S1,S2	S2,S1	S2,S2
1	S1	(0)	(2)	(4)	(1)
	S2	(1)	(1)	(0)	(0)

TAB. 4.2 – Le joueur 1 contre (2,3)

Pour trouver $v(\{1\})$, il suffit de considérer $S = \{1\}$ opposée à $\bar{S} = \{2,3\}$, que l'on peut représenter sous forme normale par le tableau 4.2.

Au minimum le joueur 1 reçoit donc $v(\{1\}) = Val \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \frac{1}{2}$.

On contruit les matrices de manière similaire afin de trouver $v(\{2\})$ et $v(\{3\})$. On obtient $v(\{2\}) = 0$ et $v(\{3\}) = \frac{3}{4}$.

Enfin, pour trouver les valeurs des coalitions à deux joueurs, nous allons considérer les coalitions composées de deux joueurs opposées au joueur restant seul. Pour $\{1,3\}$ opposée à $\{3\}$, nous obtenons la matrice 4.3 :

		2	
		S1	S2
1,3	S1,S1	(1)	(7)
	S1,S2	(3)	(1)
	S2,S1	(1)	(1)
	S2,S2	(2)	(1)

TAB. 4.3 – 2 joueurs contre 1

La valeur de $v(\{1,3\})$ est donc $Val \begin{pmatrix} 1 & 7 \\ 3 & 1 \end{pmatrix} = \frac{5}{2}$. De manière similaire, on déduit $v(\{1,2\}) = 3$ et $v(\{2,3\}) = 2$ ce qui termine la caractérisation de la fonction caractéristique du jeu sous forme coalitionnelle.

4.3.2 Propriétés et classes de jeux

Plusieurs propriétés ont emergé des jeux sous forme coalitionnelle. Dans cette partie, elles ne sont pas toutes présentées mais uniquement les plus importantes.

Définition 24 *Un jeu sous forme coalitionnelle est dit à somme constante si* $v(S) + v(\bar{S}) = v(N)$ *pour toute coalition* S*. Il est à somme nulle si* $v(N) = 0$.

Définition 25 *Si la valeur $v(N)$ de la coalition formée par tous les joueurs est strictement supérieure à la somme $\sum_{i \in N} v(i)$ des valeurs des coalitions formées par un joueur seul alors le jeu est dit* ***essentiel****.*

Ce type de jeu a la propriété que tous les joueurs ont interêt à intégrer une stratégie de groupe coopérative.

Dans le cas d'un jeu où les utilités doivent être redistribuées aux membres de la coalition, se pose le problème du partage équitable. En effet, rappelons que les utilités ne peuvent pas être comparées entre différents joueurs. Cependant, grâce à une normalisation commune des utilités pour les préférences des joueurs, nous n'effectuons pas de comparaisons directes d'utilités entre les joueurs lors de la redistribution des gains.

Nous pouvons remarquer un cas particulier des jeux à n joueurs avec coalitions : le cas où le jeu ne contient que deux coalitions S et T telles que $S \cup T = N$ et $S \cap T = \emptyset$ (c'est-à-dire S et T sont complémentaires et englobent tous les joueurs du jeu). Dans ce cas, le jeu peut se restreindre sous certains aspects aux cas des jeux à deux joueurs.

On définit la notion de jeu **stratégiquement équivalent** en forme coalitionelle comme suit :

Définition 26 *Soit $a > 0$ et $b_1, b_2, ..., b_n$ des constantes quelconques, alors le jeu w défini pour toute coalition S par $w(S) = a(v(S) - \sum_{i \in S} b_i)$ est stratégiquement équivalent à v.*

Définition 27 *Un jeu est dit* ***superadditif*** *si, pour toutes coalitions disjointes S et T, on a $v(S \cup T) \geq v(S) + v(T)$. Clairement, cette propriété signifie que la fusion de deux coalitions ne peut être que bénéfique pour chacune d'entre elles.*

Il est **sous additif** dans le cas contraire.

Définition 28 *Si $v(S) \geq v(T), \forall S \supset T$ alors le jeu v est dit* ***monotone****.*

Clairement, si une coalition est contenue dans une autre alors la valeur de la première est inférieure ou égale à la valeur de la seconde, et que ceci est vrai pour toutes les coalitions alors le jeu est monotone. On dit qu'il est **zéro-monotone** si sa (0,1)-normalisation est monotone. Un jeu superadditif est zéro-monotone mais la réciproque est fausse.

Définition 29 *Un jeu est* ***convexe*** *si pour toute coalition S et T, $v(S \cup T) + v(S \cap T) \geq v(S) + v(T)$ ou encore si pour tout joueur i et toute coalition $S \supset T$ ne contenant pas i, $v(S \cup i) - v(S) \geq v(T \cup i) - V(T)$.*

Clairement, si on a deux coalitions S et T telles que $S \supset T$, alors l'ajout d'un joueur i dans la coalition S apporte au moins autant à la coalition S que l'ajout du joueur i dans la coalition T, ou encore il est plus intéressant de rejoindre la coalition la plus grande. C'est l'effet "boule de neige" : plus la coalition est grande, plus le gain apporté par un nouveau membre est important.

Définition 30 *Un jeu est à **somme constante** si pour toutes coalitions complémentaires S et $\bar{S}$, $v(S) + v(\bar{S}) = v(N)$.*

Définition 31 *Le **jeu dual** d'un jeu v est un jeu v' définit par $v'(S) = v(N) - v(\bar{S})$ pour tout $S \subset N$.*

Définition 32 ***Un jeu simple** est un jeu monotone représenté par une fonction de coalition à valeur dans {0,1}. Clairement, une coalition S est gagnante si $v(S) = 1$ et perdante si $v(S) = 0$. De plus, la coalition vide est toujours perdante et la coalition contenant l'ensemble des joueurs est toujours gagnante.*

On peut remarquer que le jeu dual d'un jeu v est un jeu v' dans lequel les coalitions gagnante de v' sont les perdantes de v.

Définition 33 *Un jeu simple est **propre** si le complément de chaque coalition gagnante est perdante. Un jeu est **fort** si le complément de chaque coalition perdante est gagnante. Un jeu **décisif** est un jeu propre et fort.*

Un jeu de **vote pondéré** $[q : w_1, ..., w_n]$ est un jeu simple dans lequel une coalition S est gagnante si et seulement si $\sum_{i \in S} w_i \geq q$ où q est une constante positive, c'est-à-dire si la somme des poids des joueurs de S est supérieure à un quota. Chaque joueur a un poids associé qui fait que son vote a plus ou moins d'importance.

Ce type de jeu est souvent utilisé pour définir les méthodes de vote employées lors de certains conseils dans le cas où certains membres ont un pouvoir de vote plus important. C'est le cas, par exemple, du Conseil de l'Union Européenne. En effet, le vote de chaque pays prend en compte la population de ce pays : le vote d'un pays ayant beaucoup d'habitants est plus important qu'un pays qui en compte moins. Illustrons cette notion par un exemple simple : le jeu $[3; 2, 1, 1, 1]$. Dans ce jeu, le quota q est fixé à 3 et il existe quatre joueurs notés a, b, c, d ayant respectivement un poids de $2, 1, 1, 1$. On remarque facilement que ce jeu est propre et fort : il est donc décisif. On remarque également l'importance du vote du joueur a. En effet, on vérifie que la seule coalition gagnante ne contenant pas le joueur a est celle qui contient tous les autres joueurs. La seule coalition perdante contenant a est celle où a est seul. Par simple calcul, nous pouvons déduire en particulier que $v(\{a,b\} = v(\{a,c\}) = v(\{a,d\}) = 1$ quelles que soient les décisions des autres joueurs (ce qui signifie que pour eux v vaut 0).

Définition 34 *Un* ***vecteur comptant*** *d'un jeu simple à n joueur est un n-uplet* $(c_1, ..., c_n)$ *où* c_i *est le nombre de coalitions gagnantes contenant le joueur i. Aucun jeu de vote pondéré ne peut avoir le même vecteur comptant qu'un autre.*

La division des ressources est une partie importante des jeux sous forme coalitionelle ; ces divisions "imputent" un gain pour chaque joueur. L'imputation x divise un gain aux joueurs de la coalition S en donnant à chaque joueur au moins autant que ce qu'il pouvait avoir seul. On note parfois $x(S) = \sum_{i \in S} x_i$. Une imputation est un vecteur de gain qui peut s'écrire $\langle s = (x_1, x_2, ..., x_n\}\rangle : \sum_{i \in N x_i = v(N)}$ et $x_i \geq v(i\}$. Ce vecteur représente une proposition s de redistribution des gains telle que le joueur 1 recoit x_1, le joueur 2 recoit x_2, etc... respectant certaines propriétés.

Les imputations doivent respecter deux propriétés. La première est que le gain total des revenus des joueurs d'une coalition S doit être $v(S)$.

Définition 35 *Un vecteur de gain* $x = (x_1, x_2, ..., x_n)$ *est rationnel pour les groupes (ou efficace) si* $\sum_{i=1}^{n} x_i = v(N)$.

La seconde propriété assure qu'un joueur reçoit au moins autant que s'il était seul.

Définition 36 *Un vecteur de gain* x *est individuellement rationnel si* $x_i \geq v(\{i\}) \forall i = 1, ..., n$.

Définition 37 *Une imputation est un vecteur de gain qui est efficace et individuellement rationnel.*

L'ensemble des imputations peut s'écrire $\{x = (x_1, ..., x_n)$ tels que $\sum_{i \in N} x_i = v(N)$ et $x_i \geq v(\{i\}) \forall i \in N\}$.

Nous reviendrons plus tard sur les imputations pour définir le coeur du jeu.

Une imputation x domine une imputation y s'il existe une coalition S pour laquelle, pour tout joueur i de S, $x_i > y_i$ et $x(S) \leq v(S)$ c'est à dire que tout les joueurs de S préfèrent x à y et que les joueurs ensemble ne "réclament" pas plus que $v(S)$. Deux jeux sont équivalents au sens de la domination s'il existe une bijection entre leurs ensembles d'imputations qui préserve la relation de dominance.

Définition 38 ***La couverture d'un jeu*** *est un autre jeu avec des attributs au moins aussi grands que la valeur de toute coalition.*

Définition 39 ***La couverture superadditive*** *d'un jeu est le jeu v^{sa} défini pour toute coalition S par $v^{sa}(S) = \max \sum_{T \in \mathcal{P}} v(T)$ où le maximum est pris parmi toutes les partitions $\mathcal{P}$ de S.*

Définition 40 *Une coalition S* ***voit*** *une autre coalition $T \supset S$ (et donc T voit S) si la ligne reliant les points $(e^S, v(S))$ et $(e^T, v(T))$ sont sur la limite supérieure du graphe d'extension.*

Cette propriété induit des relations vers d'autres propriétés : un jeu est totalement équilibré si toutes les coalitions peuvent voir la coalition vide. Un jeu est convexe si toutes les coalitions peuvent voir leurs sur-ensembles et sous-ensembles.

Définition 41 ***Le coeur*** *d'un jeu v est l'ensemble des imputations x pour lesquelles $x(S) \geq v(S)$ pour toute coalition S.*

Nous reviendrons en détail sur cette notion dans la section 7.3.

Définition 42 *Le jeu est* ***exact*** *si pour chaque coalition S il existe une imputation x dans le noyau du jeu satisfaisant $x(S) = v(S)$. Clairement un jeu est exact si et seulement si chacune de ses coalitions voit N et la coalition vide.*

Définition 43 ***Un jeu symétrique*** *est caractérisé par une suite de $n+1$ nombres $(0, v_1, ..., v_n)$ où pour chaque coalition S, $v(S) = v_{|S|}$.*

Définition 44 *Un partenariat dans v est une coalition S pour laquelle $v(T) = v(T \setminus S)$ quand $T \cap S \neq S$.*

Cela signifie que la présence des membres d'un partenariat dans une coalition n'apporte rien à la valeur de la coalition sauf si tous les membres du partenariat y sont.

Définition 45 *Une famille est une coalition S pour laquelle $v(T) = v(T \cup S)$ quand $T \cap S \neq \emptyset$.*

Cela signifie que la présence d'un membre quelconque d'une famille dans une coalition est équivalente à la présence de tous les membres de la famille.

Définition 46 ***Un jeu avec structure de coopération*** *est un jeu coalitionnel v et une structure de coopération $\mathcal{P}$ qui associe à chaque coalition S une partition $\mathcal{P}(S)$ de S. Le jeu $v^{\mathcal{P}}$ est défini par $v^{\mathcal{P}}(S) = \sum_{S_j \in \mathcal{P}(S)} v(S_j)$ pour toute coalition S.*

Dans ce chapitre nous avons introduit les jeux à n joueurs et détaillé les différences majeures avec les jeux à deux joueurs. Nous avons également défini la notion de coalition et et la forme coalitionnelle courrament utilisée pour étudier ce type de jeu. Enfin nous avons donné plusieurs définitions et propriétés concernant cette représentation.

Le cadre théorique concernant les jeux à n joueurs étant défini, nous pouvons alors nous intéresser au jeu proposé et détailler le travail effectué sur celui-si.

Chapitre 5

Etude du jeu proposé

Cette partie est consacré à l'étude du jeu propsé proprement dit.

5.1 Définition du jeu

Le jeu est un jeu à n joueurs dans lequel chaque joueur dispose d'un ensemble de ressources initiales. A chaque tour, tous les joueurs jouent un coup simultanément. Un coup se déroule de la façon suivante :

- Le joueur demande aux joueurs de son choix leurs aides pour attaquer un joueur.
- Les joueurs contactés acceptent ou refusent.
- On calcule la somme des ressources de l'ensemble des joueurs ayant acceptés plus les ressources du joueur (donc au pire la somme est simplement les ressources du joueur). Si cette somme est supérieure aux ressources du joueur attaqué, le joueur en question perd x ressources au profit du joueur attaquant (nous poserons $x = 1$).

A la fin d'un tour, si un joueur voit ses ressources réduites à zéro, il est éliminé. Le jeu s'arrête s'il n'y a plus qu'un seul agent, ou si pendant trois tours successifs les décisions des agents ont été les mêmes et si leurs ressources sont identiques.

En ce qui concerne les informations disponibles pour chaque joueur on peut envisager deux cas possibles :

- Informations privées : un agent ne connait que l'historique des coups le concernant, c'est-à-dire les joueurs qu'il a attaqué, ceux qui l'ont attaqué, ceux qui ont soutenu des attaques contre lui, et ceux qui ont accepté ou refusé de le soutenir.
- Informations publiques : Les informations dont dispose chaque joueur sont l'ensemble des attaques qui ont eu lieu aux tours précédents (y

compris les soutiens des attaques) et bien sur l'ensemble des propositions d'attaques qu'il a refusé ou qui lui ont été refusées.

Au niveau d'un joueur il y a donc trois types de décisions à prendre :
– décider quel joueur attaquer
– décider à quels joueurs demander du soutien
– décider quelles attaques soutenir

Nous avons choisi de considérer le cas des informations publiques : le joueur a connaissance de tout ce qui s'est passé au tour précédent. Il connait les attaques, les demandes de soutien et les décisions prises par tous les joueurs au tour précédent. De plus, il a connaissance des attaques des joueurs avant de prendre les décisions sur les attaques qu'il va soutenir ou non c'est-à-dire que le juge lui indique quelles sont les attaques et les demandes de soutien des autres joueurs avant qu'il prenne sa décision sur les attaques à supporter ou non.

5.2 Analyse générale

Tout d'abord, il est clair que ce jeu est à information complète. En effet, chaque joueur a une parfaite connaissance du jeu et de ses règles. En particulier, il connait les stratégies disponibles pour les autres joueurs et les utilités associées. En revanche, il est à information imparfaite car les coups sont joués de manière simultanée.

Clairement il s'agit d'un jeu à n joueurs à somme nulle. En effet, ce qui est perdu par un joueur est gagné par un autre et réciproquement.

Des coalitions pourront se former au sein de ce jeu car il est à n joueurs. Notre étude doit donc s'orienter sur la théorie des jeux à n joueurs ; en particulier sur la forme coalitionnelle du jeu.

Avant de pouvoir travailler sur les coalitions dans le jeu, il faut définir la notion de coalition. En effet, il nous faut une définition pour savoir quand deux joueurs sont dans une même coalition ou non. Plusieurs définitions peuvent être envisagées.

Deux joueurs sont dans une même coalition si :

1. Aucun des deux n'a attaqué l'autre.
2. Aucun des deux n'a attaqué l'autre et n'a soutenu d'attaque contre l'autre.
3. Aucun des deux n'a attaqué l'autre et n'a soutenu d'attaque contre l'autre et chacun a accepté de soutenir l'attaque de l'autre.
4. Aucun des deux n'a attaqué l'autre et n'a soutenu d'attaque contre l'autre et ils ont tout deux attaqué directement le même joueur.

5. Aucun des deux n'a attaqué l'autre et n'a soutenu d'attaque contre l'autre et chacun a accepté de soutenir l'attaque de l'autre et ils ont tout deux attaqué directement le même joueur.

La première définition n'est pas assez contraignante. En effet, si les joueurs s'attaquent par voie de support on ne peut pas dire qu'ils sont dans une même coalition. La deuxième et la troisième définition semblent plus envisageables. La troisième correspond plus à la réalité mais est peu être un peu trop contraignante. Les définitions 4 et 5 ajoutent une condition sur le joueur attaqué par rapport aux définitions 2 et 3. Le choix de la définition qui sera préférée devrait se faire après quelques expérimentations.

Pour simplifier l'écriture, on écrira qu'un joueur *attaque directement* un autre s'il le choisit comme joueur à attaquer et qu'il *attaque indirectement* un autre joueur s'il accepte de soutenir une attaque contre lui.

Intuitivement, une coalition plus grande garantit généralement de plus grandes chances de succès qu'une autre de taille moindre.

Nous pouvons remarquer que le jeu est superadditif. En effet, $v(S) + v(T) \leq v(S \cup T)$ quelles que soient les coalitions S et T car l'union de deux coalitions présente plus de ressources et donc plus de chances de succès lors des attaques que si les coalitions sont séparées. Cela confirme notre intuition. Ce jeu n'est pas essentiel mais il est monotone car $v(S) \geq v(T), \forall S \subset T$.

5.3 Programme de simulation

Afin de réaliser des simulations de ce jeu, un programme a été créé en Java. Il permet d'éditer un jeu à partir d'une interface graphique en saisissant tous les paramètres de la partie (figure 5.1).

On peut saisir le nombre de joueurs de la partie et le nombre de points (de ressources) dont ils disposent au début du jeu. On sélectionne la valeur de la graine, le niveau de détail de l'historique, et la définition d'une coalition. La graine est le nombre utilisé pour initialiser le générateur de nombre pseudo-aléatoire. Cela nous permet de pouvoir exécuter des parties qui produiront toujours les mêmes résultats même si elles utilisent les nombres aléatoires. Il faut tester les parties en faisant varier cette graine pour avoir des résultats pertinents. On sélectionne ensuite les fichiers dans lesquels l'historique et le graphe du jeu seront sauvegardés. Les derniers paramètres permettent de créer des variantes du jeu en modifiant certaines règles. En effet, on peut saisir le nombre de points perdus par le joueur attaqué (en cas de succès de l'attaque, loosing points sur la fenêtre), le pourcentage de ces points rétribué au joueur attaquant (winning points) et au joueurs ayant soutenus cette attaque (coalition points). Enfin, on peut sélectionner si la retribution des points à

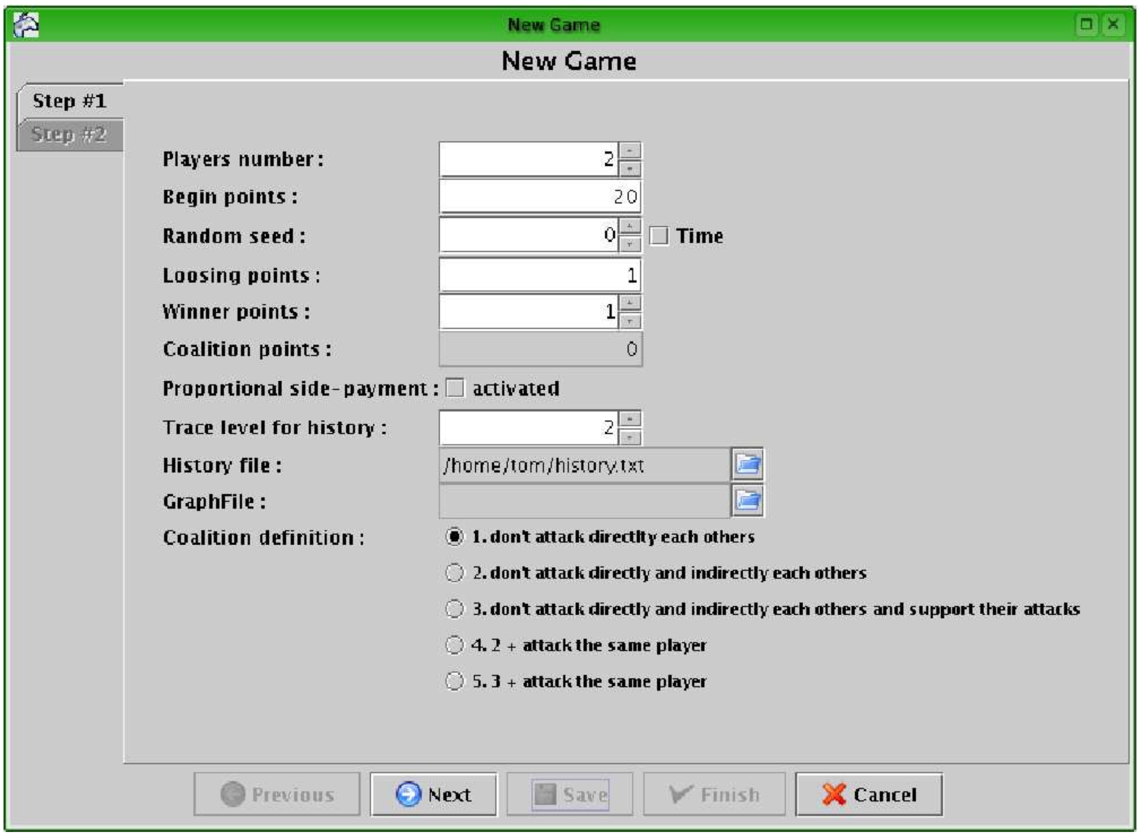

FIG. 5.1 – La première fenêtre d'édition d'un nouveau jeu

ces derniers est identique pour tous ou proportionnelle aux ressources qu'ils possèdent (proportional side payment). Nous n'utiliserons pas ces variantes lors des tests, nous utiliserons les paramètres "par défaut" définis lors de la description du jeu au début de ce chapitre.

La dernière étape consiste à affecter une stratégie à chaque joueur (figure 5.2).

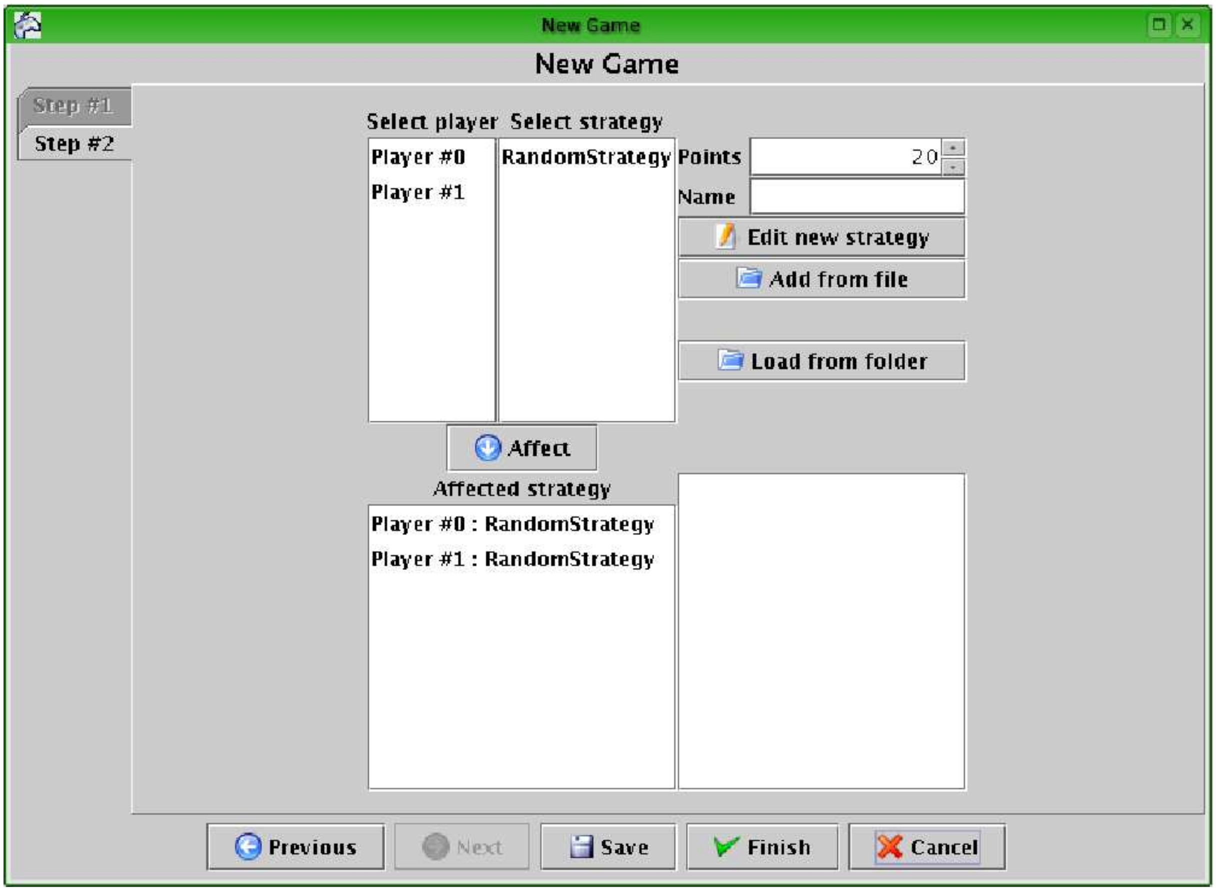

FIG. 5.2 – La seconde fenêtre d'édition d'un nouveau jeu

Il est possible d'éditer une stratégie à partir des sous-stratégies (voir section suivante) déjà disponibles dans le programme pour créer une stratégie de son choix (figure 5.3).

De plus les stratégies ainsi construites peuvent être sauvegardées dans des fichiers (les objets sont serialisables) afin de permettre à l'utilisateur de se créer une banque de stratégies réutilisables par la suite.

Ces parties peuvent être sauvegardées dans un fichier XML permettant

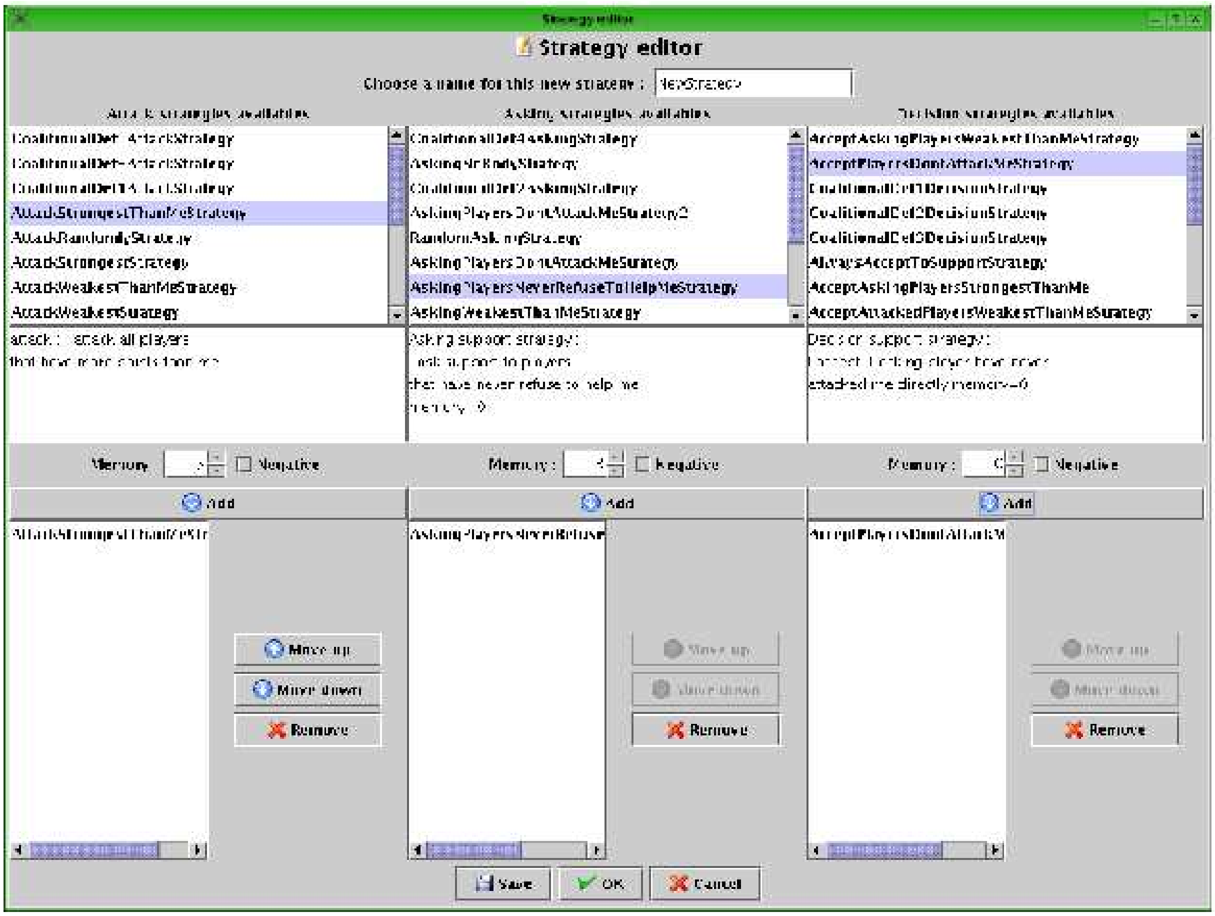

FIG. 5.3 – La fenêtre d'édition d'une stratégie

un lancement du jeu en mode console. Toutes les informations sur le jeu sont mémorisées dans plusieurs fichiers générés par le programme stockant l'historique des coups joués, le résultat du jeu, le graphe du jeu et les coalitions formées. Ces fichiers contiennent des tableaux synthétisant les résultats ; on pourra trouver un exemple du fichier de classement des joueurs en annexe avec le tableau 7.3. Ce tableau contient, pour chaque joueur, le classement du joueur, la stratégie qu'il utilise et le dernier tour auquel il a participé. On trouvera également en annexe avec le tableau 7.4 un exemple du fichier de classement des coalitions. Ce tableau contient le classement de la coalition et les joueurs qu'elle contient, le nombre de fois où elle est apparue dans le jeu, le nombre de séquences dans laquelle elle est apparue et la longueur maximum de ces séquences. Ces fichiers nous permettront d'analyser ce qui s'est passé dans le jeu.

L'exécution d'une partie s'effectue donc en mode graphique après l'édition d'un jeu depuis l'interface ou depuis un fichier XML. L'exécution peut également se dérouler en mode tour par tour ; il suffit de cliquer sur un bouton pour passer au tour suivant. Avec ce mode de fonctionnement, il est possible de visualiser l'historique du coup passé, les coalitions formées et le graphe du jeu simultanément. L'exécution en mode console s'effectue depuis un fichier XML.

Une fois la partie terminée, il est possible de visualiser l'historique, le graphe du jeu, le graphe représentant le nombre de coalitions formées par tour, le classement de la partie, le classement des coalitions et les statistiques des joueurs (on peut voir qui a attaqué qui et combien de fois, qui a demandé du soutien à qui et combien de fois, et qui a accepté de soutenir une attaque de qui et combien de fois).

Une seconde partie du programme permet de récuperer les données des parties testées depuis les fichiers et de les insérer dans une base de données pour en simplifier l'analyse. Le serveur de base de données choisie est MySQL et les données y sont insérées depuis le programme grâce à JDBC. On récupère ainsi les données d'un jeu à partir de simples requêtes et non plus à partir d'une multitude de fichiers. Quelques fenêtres ont été créées pour visualiser les résultats contenus dans la base. Cette base peut également servir de point de départ pour générer dynamiquement des fichiers HTML afin d'afficher ou publier les résultats de son choix.

Globalement, le programme complet repose sur plus de 150 classes ce qui représente plus de 30000 lignes de code.

5.4 Stratégies

Afin de pouvoir effectuer des expérimentations sur des stratégies très variées, la stratégie du jeu a été décomposée en trois sous-stratégies.

- *La stratégie d'attaque* qui décide quel joueur attaquer.
- *La stratégie de demande de soutien* qui décide les joueurs à qui demander de soutenir l'attaque.
- *La stratégie d'acceptation de soutien* qui décide d'accepter ou de refuser de soutenir l'attaque d'un joueur contre un autre.

Une stratégie complète du jeu est donc composée de ces trois sous-stratégies.

Pour affiner encore les tests, chaque sous-stratégie est construite à partir d'une liste de stratégies d'attaque, de demande de soutien ou d'acceptation de soutien. Cette liste est triée par ordre de priorité. Pour les stratégies d'attaque et de demande de soutien, chaque entrée dans la liste agit comme un filtre. Pour la stratégie d'acceptation de soutien, chaque sous-stratégie de la liste doit accepter le soutien pour que la stratégie elle même l'accepte également.

Certaines stratégies possèdent un paramètre appelé la mémoire représenté par un entier m. Cet entier désigne le nombre de tours pendant lequel les informations sont mémorisées par la stratégie. Si cet entier vaut 0 alors la stratégie garde en mémoire les informations passées depuis le début du jeu.

Enfin, toutes les stratégies possèdent leur complémentaire. Ainsi, par exemple, pour une stratégie d'attaque qui attaque les joueurs qu'ils l'ont attaqué au tour précédent, la stratégie complémentaire retourne les joueurs qui ne l'ont pas attaqué au tour précédent.

5.4.1 Stratégies d'attaque

AttackRandomlyStrategy J'attaque aléatoirement un joueur.

AttackStrongestStrategy J'attaque le(s) joueur(s) ayant le plus de ressources.

AttackWeakestStrategy J'attaque le(s) joueur(s) ayant le moins de ressources.

SpitefulAttackStrategy J'attaque le(s) joueur(s) qui m'a(ont) attaqué lors des m derniers tours.

SpitefulAttackStrategy2 J'attaque le(s) joueur(s) qui m'a(ont) attaqué ou qui ont soutenu une attaque contre moi lors des m derniers tours.

AttackAttackedPlayersStrategy J'attaque les joueurs que j'ai attaqué lors des m derniers tours.

AttackAttackedPlayersStrategy2 J'attaque les joueurs que j'ai attaqué ou contre lesquels j'ai soutenu une attaque lors des m derniers tours.

AttackTheMostAttackedPlayerStrategy J'attaque le joueur qui a été le plus souvent attaqué lors des m derniers tours.

AttackTheMostSuccessfulyAttackedPlayerStrategy J'attaque le joueur qui a été le plus souvent attaqué avec succès lors des m derniers tours.

AttackWeakestThanMeStrategy J'attaque les joueurs qui ont strictement moins de points que moi.

AttackStrongestThanMeStrategy J'attaque les joueurs qui ont strictement plus de points que moi.

Les cinq stratégies suivantes sont à différencier des autres car elles utilisent explicitement l'aspect coalitionnel du jeu.

CoalitionalDef1AttackStrategy J'attaque les joueurs qui ne sont pas dans ma coalition (définition 1).

CoalitionalDef2AttackStrategy J'attaque les joueurs qui ne sont pas dans ma coalition (définition 2).

CoalitionalDef3AttackStrategy J'attaque les joueurs qui ne sont pas dans ma coalition (définition 3).

CoalitionalDef4AttackStrategy J'attaque les joueurs qui ne sont pas dans ma coalition (définition 4).

CoalitionalDef5AttackStrategy J'attaque les joueurs qui ne sont pas dans ma coalition (définition 5).

5.4.2 Stratégies de demande de soutien

AskingEveryoneStrategy Je demande du soutien à tous les autres joueurs.

AskingNoBodyStrategy Je ne demande du soutien à personne.

RandomAskingStrategy Je choisi un nombre x de manière aléatoire et je choisis aléatoirement x joueurs auxquels je demande du soutien.

AskingPlayersDontAttackMeStrategy Je ne demande du soutien qu'à tous ceux qui ne m'ont pas attaqué directement lors des m derniers tours.

AskingPlayersDontAttackMeStrategy2 Je ne demande du soutien qu'à tous ceux qui ne m'ont pas attaqué directement et qui n'ont pas soutenu d'attaque contre moi lors des m derniers tours.

AskingPlayersNeverRefuseToHelpMeStrategy Je ne demande du soutien à tous ceux qui n'ont jamais refusé de m'aider lors des m derniers tours.

AskingPlayersWeakestThanMeStrategy Je demande de l'aide aux joueurs qui ont strictement moins de points que moi.

AskingPlayersStrongestThanMeStrategy Je demande de l'aide aux joueurs qui ont strictement plus de points que moi.

Les cinq stratégies suivantes sont à différencier des autres car elles utilisent explicitement l'aspect coalitionnel du jeu.

CoalitionalDef1AskingStrategy Je demande du soutien à tous ceux qui sont dans ma coalition (définition 1).

CoalitionalDef2AskingStrategy Je demande du soutien à tous ceux qui sont dans ma coalition (définition 2).

CoalitionalDef3AskingStrategy Je demande du soutien à tous ceux qui sont dans ma coalition (définition 3).

CoalitionalDef4AskingStrategy Je demande du soutien à tous ceux qui sont dans ma coalition (définition 4).

CoalitionalDef5AskingStrategy Je demande du soutien à tous ceux qui sont dans ma coalition (définition 5).

5.4.3 Stratégies d'acceptation de soutien

AcceptOneByTwoStrategy Je réponds oui une fois sur deux.

AlwaysAcceptStrategy J'accepte toujours.

NeverAcceptStrategy Je refuse toujours.

RandomSupportStrategy J'accepte avec une probabilité de 0.5

AcceptPlayersDontAttackMeStrategy J'accepte d'aider tous ceux qui ne m'ont pas attaqué directement lors des m derniers tours.

AcceptPlayersDontAttackMeStrategy2 J'accepte d'aider tous ceux qui ne m'ont pas attaqué directement et qui n'ont pas soutenu d'attaque contre moi lors des m derniers tours.

AcceptPlayersNeverRefuseHelpMeStrategy J'accepte d'aider tous ceux qui n'ont jamais refusé de m'aider lors des m derniers tours.

AcceptAskingPlayersWeakestThanMeStrategy J'accepte d'aider les joueurs qui ont moins de points que moi.

AcceptAskingPlayersStrongestThanMeStrategy J'accepte d'aider les joueurs qui plus de points que moi.

AcceptAttackedPlayersWeakestThanMeStrategy J'accepte de soutenir les attaques si elles sont contre des joueurs qui ont moins de points que moi.

AcceptAttackedPlayersStrongestThanMeStrategy J'accepte de soutenir les attaques si elles sont contre des joueurs qui ont plus de points que moi.

Les cinq stratégies suivantes sont à différencier des autres car elles utilisent explicitement l'aspect coalitionnel du jeu.

CoalitionalDef1DecisionStrategy J'accepte d'aider les attaques des joueurs qui sont dans ma coalition contre ceux qui n'y sont pas (définition 1).

CoalitionalDef2DecisionStrategy J'accepte d'aider les attaques des joueurs qui sont dans ma coalition contre ceux qui n'y sont pas (définition 2).

CoalitionalDef3DecisionStrategy J'accepte d'aider les attaques des joueurs qui sont dans ma coalition contre ceux qui n'y sont pas (définition 3).

CoalitionalDef4DecisionStrategy J'accepte d'aider les attaques des joueurs qui sont dans ma coalition contre ceux qui n'y sont pas (définition 4).

CoalitionalDef5DecisionStrategy J'accepte d'aider les attaques des joueurs qui sont dans ma coalition contre ceux qui n'y sont pas (définition 5).

5.4.4 Stratégies complètes

Ces stratégies sont les stratégies directement utilisables par les joueurs.
RandomStrategy
Cette stratégie est composée de :
Attaque : AttackRandomlyStrategy.
Demande de soutien : RandomAskingStrategy
Acceptation de soutien : RandomSupportStrategy
PowerStrategy
Cette stratégie est composée de :
Attaque :
1 : AttackWeakestStrategy
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingEveryBodyStrategy
Acceptation de soutien :
1 : AlwaysAcceptStrategy

StupidStrategy
Cette stratégie est composée de :
Attaque :
1 : AttackStrongestStrategy
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingNoBodyStrategy

Acceptation de soutien :
1 : NeverAcceptStrategy

SpitefulStrategyMemory0
Cette stratégie est composée de :
Attaque :
1 : SpitefulAttackStrategy (mémoire 0)
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersDontAttackMeStrategy (mémoire 0)
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy (mémoire 0)

SpitefulStrategyMemory5
Cette stratégie est composée de :
Attaque :
1 : SpitefulAttackStrategy (mémoire 5)
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersDontAttackMeStrategy (mémoire 5)
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy (mémoire 5)

SpitefulStrategyMemory10
Cette stratégie est composée de :
Attaque :
1 : SpitefulAttackStrategy (mémoire 10)
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersDontAttackMeStrategy (mémoire 10)
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy (mémoire 10)

SpitefulStrategy2Memory0
Cette stratégie est composée de :
Attaque :
1 : SpitefulAttackStrategy2 (mémoire 0)
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersDontAttackMeStrategy2 (mémoire 0)
Acceptation de soutien :

1 : AcceptPlayersDontAttackMeStrategy2 (mémoire 0)

SpitefulStrategy2Memory5
Cette stratégie est composée de :
Attaque :
1 : SpitefulAttackStrategy2 (mémoire 5)
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersDontAttackMeStrategy2 (mémoire 5)
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy2 (mémoire 5)

SpitefulStrategy2Memory10
Cette stratégie est composée de :
Attaque :
1 : SpitefulAttackStrategy2 (mémoire 10)
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersDontAttackMeStrategy2 (mémoire 10)
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy2 (mémoire 10)

SpitefulStrategy3Memory0
Cette stratégie est composée de :
Attaque :
1 : SpitefulAttackStrategy (mémoire 0)
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersNeverRefuseToHelpMeStrategy (mémoire 0)
Acceptation de soutien :
1 : AcceptPlayersNeverRefuseToHelpMeStrategy (mémoire 0)

SpitefulStrategy3Memory5
Cette stratégie est composée de :
Attaque :
1 : SpitefulAttackStrategy (mémoire 5)
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersNeverRefuseToHelpMeStrategy (mémoire 5)
Acceptation de soutien :

1 : AcceptPlayersNeverRefuseToHelpMeStrategy (mémoire 5)

SpitefulStrategy3Memory10
Cette stratégie est composée de :
Attaque :
1 : SpitefulAttackStrategy (mémoire 10)
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersNeverRefuseToHelpMeStrategy (mémoire 10)
Acceptation de soutien :
1 : AcceptPlayersNeverRefuseToHelpMeStrategy (mémoire 10)

PersistentStrategyMemory0
Cette stratégie est composée de :
Attaque :
1 : AttackAttackedPlayerStrategy (mémoire 0)
2 : AttackTheMostAttackedPlayerStrategy (mémoire 0)
3 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingEveryOneStrategy
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy (mémoire 0)

PersistentStrategyMemory5
Cette stratégie est composée de :
Attaque :
1 : AttackAttackedPlayerStrategy (mémoire 5)
2 : AttackTheMostAttackedPlayerStrategy (mémoire 5)
3 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingEveryOneStrategy
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy (mémoire 5)

PersistentStrategyMemory10
Cette stratégie est composée de :
Attaque :
1 : AttackAttackedPlayerStrategy (mémoire 10)
2 : AttackTheMostAttackedPlayerStrategy (mémoire 10)
3 : AttackRandomlyStrategy
Demande de soutien :

1 : AskingEveryOneStrategy
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy (mémoire 10)

PersistentStrategy2Memory0
Cette stratégie est composée de :
Attaque :
1 : AttackAttackedPlayerStrategy2 (mémoire 0)
2 : AttackTheMostAttackedPlayerStrategy (mémoire 0)
3 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingEveryOneStrategy
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy2 (mémoire 0)

PersistentStrategy2Memory5
Cette stratégie est composée de :
Attaque :
1 : AttackAttackedPlayerStrategy2 (mémoire 5)
2 : AttackTheMostAttackedPlayerStrategy (mémoire 5)
3 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingEveryOneStrategy
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy2 (mémoire 5)

PersistentStrategy2Memory10
Cette stratégie est composée de :
Attaque :
1 : AttackAttackedPlayerStrategy2 (mémoire 10)
2 : AttackTheMostAttackedPlayerStrategy (mémoire 10)
3 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingEveryOneStrategy
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy2 (mémoire 10)

BadStrategy
Cette stratégie est composée de :
Attaque :
1 : AttackWeakestThanMe

2 : AttackWeakestStrategy
3 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingStrongestThanMeStrategy
Acceptation de soutien :
1 : AcceptAttackedPlayersWeakestThanMeStrategy

AltruistStrategy
Cette stratégie est composée de :
Attaque :
1 : AttackStrongestThanMe
2 : AttackStrongestStrategy
3 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingWeakestThanMeStrategy
Acceptation de soutien :
1 : AcceptAttackedPlayersStrongestThanMeStrategy

SimpleCoalitionalDef1Strategy
Cette stratégie est composée de :
Attaque :
1 : CoalitionalDef1AttackStrategy
2 : AttackRandomlyStrategy
Demande de soutien :
1 : CoalitionalDef1AskingStrategy
Acceptation de soutien :
1 : CoalitionalDef1DecisionStrategy

SimpleCoalitionalDef2Strategy
Cette stratégie est composée de :
Attaque :
1 : CoalitionalDef2AttackStrategy
2 : AttackRandomlyStrategy
Demande de soutien :
1 : CoalitionalDef2AskingStrategy
Acceptation de soutien :
1 : CoalitionalDef2DecisionStrategy

SimpleCoalitionalDef3Strategy
Cette stratégie est composée de :
Attaque :

1 : CoalitionalDef3AttackStrategy
2 : AttackRandomlyStrategy
Demande de soutien :
1 : CoalitionalDef3AskingStrategy
Acceptation de soutien :
1 : CoalitionalDef3DecisionStrategy

<u>SimpleCoalitionalDef4Strategy</u>
Cette stratégie est composée de :
Attaque :
1 : CoalitionalDef4AttackStrategy
2 : AttackRandomlyStrategy
Demande de soutien :
1 : CoalitionalDef4AskingStrategy
Acceptation de soutien :
1 : CoalitionalDef4DecisionStrategy

<u>SimpleCoalitionalDef5Strategy</u>
Cette stratégie est composée de :
Attaque :
1 : CoalitionalDef5AttackStrategy
2 : AttackRandomlyStrategy
Demande de soutien :
1 : CoalitionalDef5AskingStrategy
Acceptation de soutien :
1 : CoalitionalDef5DecisionStrategy

<u>AdvancedCoalitionalDef1Strategy</u>
Cette stratégie est composée de :
Attaque :
1 : CoalitionalDef1AttackStrategy
2 : SpitefulAttackStrategy2 (Memory 5)
3 : AttackWeakestStrategy
4 : AttackRandomlyStrategy
Demande de soutien :
1 : CoalitionalDef1AskingStrategy
Acceptation de soutien :
1 : CoalitionalDef1DecisionStrategy

<u>AdvancedCoalitionalDef2Strategy</u>
Cette stratégie est composée de :

Attaque :
1 : CoalitionalDef2AttackStrategy
2 : SpitefulAttackStrategy2 (Memory 5)
3 : AttackWeakestStrategy
4 : AttackRandomlyStrategy
Demande de soutien :
1 : CoalitionalDef2AskingStrategy
Acceptation de soutien :
1 : CoalitionalDef2DecisionStrategy

AdvancedCoalitionalDef3Strategy
Cette stratégie est composée de :
Attaque :
1 : CoalitionalDef3AttackStrategy
2 : SpitefulAttackStrategy2 (Memory 5)
3 : AttackWeakestStrategy
4 : AttackRandomlyStrategy
Demande de soutien :
1 : CoalitionalDef3AskingStrategy
Acceptation de soutien :
1 : CoalitionalDef3DecisionStrategy

AdvancedCoalitionalDef4Strategy
Cette stratégie est composée de :
Attaque :
1 : CoalitionalDef4AttackStrategy
2 : SpitefulAttackStrategy2 (Memory 5)
3 : AttackWeakestStrategy
4 : AttackRandomlyStrategy
Demande de soutien :
1 : CoalitionalDef4AskingStrategy
Acceptation de soutien :
1 : CoalitionalDef4DecisionStrategy

AdvancedCoalitionalDef5Strategy
Cette stratégie est composée de :
Attaque :
1 : CoalitionalDef5AttackStrategy
2 : SpitefulAttackStrategy2 (Memory 5)
3 : AttackWeakestStrategy
4 : AttackRandomlyStrategy

Demande de soutien :
1 : CoalitionalDef5AskingStrategy
Acceptation de soutien :
1 : CoalitionalDef5DecisionStrategy

Dans ce chapitre, nous avons tout d'abord décrit le jeu à étudier. Nous avons défini les notions et les termes utilisés pour l'étude de ce jeu puis nous avons décrit le simulateur qui a été codé et les possibilités qu'il offre. Enfin nous avons détaillé les stratégies implémentées qui seront utilisées lors des expérimentations.

Chapitre 6

Expérimentations

Maintenant que les notions et définitions sont posées et les stratégies définies, nous pouvons procéder aux tests pour réaliser nos objectifs. Ces tests ont été décomposés en séries qui apportent les informations utilisées permettant de confirmer ou d'infirmer certaines hypothèses.

Les expérimentations ont été effectuées sur les machines de l'IUT de Lille grâce à un accès SSH et FTP sur ces machines. J'avais à ma disposition 40 postes sur lesquels les parties s'exécutaient. Au total, plus de 12000 rencontres ont été effectuées.

6.1 Objectifs et hypothèses

Nos objectifs sont principalement de découvrir s'il existe une stratégie imbattable, sinon s'il existe de smeilleures stratégies et dans ce cas, il faut les définir. Nous allons aussi essayer de découvrir l'importance des coalitions dans le jeu.

Nous pouvons nous poser la question de savoir si des coalitions se forment naturellement au sein du jeu. Les bonnes stratégies utilisent-elles l'aspect coalitionnel du jeu ou sont-elles plus "indépendantes" ? Les coalitions se maintiennent-elles longtemps dans le jeu ou sont-elles rapidement démantelées ?

Enfin, on peut chercher après des parties qui ne se terminent jamais (appelées des *oscillateurs*).

6.2 Série 1

Cette première série se compose de 1460 parties composées de 30 joueurs. Chaque joueur possède une stratégie distincte et les stratégies reposant sur les définitions de coalition ne sont pas utilisées. En enlevant ces stratégies, il reste 9 stratégies d'attaques (dont 6 utilisant le paramètre mémoire), 11 stratégies de demande de soutien (dont 3 utilisant le paramètre mémoire) et 11 stratégies d'acceptation de soutien (dont 3 utilisant le paramètre mémoire). Les stratégies utilisant le paramètre mémoire seront générées trois fois avec les valeurs 0 (depuis le début du jeu), 1 et 10. Chacune des stratégies possedant un complémentaire, nous avons donc $((6 * 3 + 3) * 2) * ((3 * 3 + 5) * 2) * ((3 * 3 + 5) * 2) = 43792$ stratégies disponibles. Le test se compose donc de 1460 parties de 30 joueurs, chacun jouant avec une stratégie distincte et possédant 50 points de ressources au départ. Le nombre de tours maximum est fixé à 20000.

Les 1460 parties sont testées avec 6 valeurs de graines différentes ce qui porte à 8760 le nombre de parties.

On calcule les meilleures stratégies pour une partie en faisant la moyenne de leur ordre d'arrivée sur les instances de cette partie avec les 6 valeurs de graines. On récupère les trois meilleures stratégies de chaque partie que l'ont fait rejouer entre elles sous forme de tournoi. On sélectionne donc 4380 $((8760*3)/6)$ stratégies et on reforme 146 parties de 30 joueurs que l'on teste avec 6 valeurs de graines (ce qui donne 876 parties). De la même manière, après sélection des trois meilleures stratégies de chaque partie, on reforme 438 $((876*3)/6)$ stratégies pour recréer 14 parties de 30 joueurs et une de 18. On les teste toutes sur 6 graines ce qui nous donne 90 parties. De façon similaire, après avoir gardé les meilleures stratégies (35 au total), il nous reste 1 partie de 30 joueurs et une de 15. Enfin, après sélection des meilleures stratégies (6 au total) on construit le dernier jeu de 6 joueurs qui est également testé pour 6 valeurs de graines. Les 3 meilleures stratégies sélectionnées parmi les 6 ont ensuite été également testées directemment entre elles pour n'en garder qu'une.

Ce premier test représente plus de 500 heures de temps CPU et 1.6 Go de données.

La définition de coalition choisie est la deuxième c'est à dire que deux joueurs sont dans la même coalition s'ils ne se sont attaqués ni directement ni indirectement.

Ce premier test va nous permettre de vérifier si des coalitions se forment naturellement et si des bonnes stratégies émergent. Nous appelerons "bonnes stratégies" les stratégies terminant régulièrement dans les premières places des jeux. De plus, nous allons pouvoir vérifier si les critères de bonnes stratégies posés par Axelrod sont encore valables dans ce jeu. Ces critères sont :

- La gentillesse
- La réactivité
- L'indulgence
- La simplicité

6.3 Série 2

Cette série de tests se compose de stratégies plus complexes que l'on oppose directement l'une contre l'autre. Chaque partie est un duel d'une stratégie contre une autre : 15 joueurs jouant avec la première et 15 autres avec la seconde. Les stratégies utilisées sont les 29 stratégies décrites dans le paragraphe 5.4.4 décrivant les stratégies complètes. Ce test se compose 435 parties de 30 joueurs et il est réalisé pour 6 valeurs de graines différentes. On a donc 2610 parties testées.

Ce second test représente plus de 300 heures de temps CPU et plus de 300 Mo de données.

La définition de coalition choisie est également la seconde.

Ce test va nous permettre de voir si parmi ces stratégies il en existe une qui gagne contre toutes les autres. Nous pourrons également voir si les stratégies utilisant pleinement la notion de coalition produisent de bons résultats.

6.4 Analyse des résultats

6.4.1 Série 1

Dans la première série de tests, nous comparions toutes les stratégies constituées de sous-stratégies avec des filtres composés d'un seul élément.

Tout d'abord nous pouvons dire que les résultats de chaque partie sont très différents en fonction de la valeur de la graine. Pour un jeu donné, les résultats sont toujours différents lors des tests avec les 6 valeurs de graines. Cette première remarque montre l'importance des nombres aléatoires lors de ces tests. En effet, le premier coup d'attaque de chaque joueur est nécessairement donné par la stratégie AttackRandomlyStrategy et donc les premiers joueurs attaqués dépendent entièrement de la valeur de la graine. Comme les premiers tours du jeu sont différents et que les stratégies dépendent des tours précédents alors la suite du jeu est également complètement modifiée. C'est principalement pour cette raison que la valeur de la graine est très importante et que les tests doivent être effectués avec plusieurs valeurs.

Cette première série de tests, montre aussi l'importance de l'environnement dans lequel la stratégie évolue. Une stratégie peut finir parmi les premières d'un jeu donné mais si on modifie quelques joueurs de ce jeu, cette même stratégie va terminer beaucoup plus loin au classement. Par exemple j'ai exécuté un jeu de trente joueur puis je l'ai exécuté à nouveau en ajoutant un 31ième joueur. Les stratégies qui finissaient dans les trois premières positions pour le premier finissent respectivement 12ième, 25ième et 30ième dans le second.

Après avoir effectué la sélection des meilleures stratégies jusqu'à ne plus avoir qu'une seule partie il en ressort que les trois meilleures stratégies sont les suivantes :

Strategy #7082

Attaque :

1 : AttackAttackedPlayersStrategy mémoire = 1

2 : AttackRandomlyStrategy

Demande de soutien :

1 : AskingPlayersDontAttackMeStrategy2 mémoire = 0

Acceptation de soutien :

1 : NeverAcceptToSupportStrategy mémoire = 0 NEGATIVE

Strategy #4904

Attaque :

1 : AttackTheMostAttackedPlayerStrategy mémoire = 0 NEGATIVE

2 : AttackRandomlyStrategy

Demande de soutien :

1 : AskingNoBodyStrategy mémoire = 0 NEGATIVE

Acceptation de soutien :

1 : AcceptPlayersNeverRefuseHelpMeStrategy mémoire = 10 NEGATIVE

Strategy #32289

Attaque :

1 : SpitefulAttackStrategy mémoire = 10

2 : AttackRandomlyStrategy mémoire = 0

Demande de soutien :

1 : AskingWeakestThanMeStrategy mémoire = 0 NEGATIVE

Acceptation de soutien :

1 : AcceptPlayersNeverRefuseHelpMeStrategy mémoire = 0 NEGATIVE

On remarque que ces trois stratégies demandent assez facilement du soutien aux autres joueurs. En particulier, la stratégie #4904 demande même du soutien à tous les autres joueurs. Nous pouvons donc en déduire que les bonnes stratégies ne doivent pas hésiter à demander du soutien aux autres joueurs pour soutenir leurs attaques. De plus, les stratégies composées des sous-stratégies ne demandant du soutien à personne (c'est-à-dire AskingNoBodyStrategy et le complémentaire de AskingEveryoneStrategy) ne sont plus qu'une trentaine après la première sélection (c'est-à-dire parmi les 4380 meilleures stratégies) et il n'y en a plus aucune après la seconde (c'est-à-dire parmi les 438 meilleures stratégies).

La stratégies 7082 est une stratégie "persistente" c'est-à-dire une stratégie qui attaque des joueurs qui ont déjà été attaqués auparavant. De manière générale, ces stratégies produisent de bons résultats.

On remarque également que l'on retrouve une stratégie qui contient une stratégie d'attaque rancunière (SpitefulAttackStrategy) que l'on a déjà vu dans le dilemme des prisonniers et qui donnait de bons résultats. De plus cette stratégie a une mémoire de 10 tours. Cela montre que les critères de gentillesse, de réactivité et d'indulgence sont respectés. En effet, cette stratégie n'attaque pas de joueurs tant qu'elle n'a pas été attaqué : elle respecte donc le critère de gentillesse et de réactivité. Quand elle est attaquée, elle réagit. Elle respecte également le critère d'indulgence car elle est paramétrée avec une mémoire de 10. En effet, elle ne se souvient des attaques contre elle que pendant 10 tours et ensuite elle pardonne les joueurs qui l'ont attaqué.

Pour être exact, ces critères ne sont que partiellement respectés car ces stratégies peuvent soutenir des attaques contre des joueurs qui ne l'ont encore jamais attaqué. De plus, elles ne réagissent pas lorsque des joueurs les attaques indirectement (par voie de support). Cependant, elles correspondent globalement à ces critères.

Nous pouvons donc dire que les 3 premiers critères d'Axelrod sont globalement vérifiés après cette première série de tests. Toutes ces stratégies sont simples car elles n'ont toutes qu'un seul élément comme filtre. Pour vérifier si le critère de simplicité est respecté, il faudrait effectuer une série de tests comparant ces stratégies avec d'autres plus complexes ; c'est-à-dire des stratégies avec des filtres plus fournis comme par exemple celles définies dans les stratégies complètes.

Nous pouvons donc préciser que les bonnes stratégies doivent globalement respecter les critères d'Axelrod et ne pas hésiter à demander du soutien aux autres joueurs pour soutenir leurs attaques.

Pour information, l'opposition entre ces trois stratégies donne le classement suivant :

1. Strategy #32289
2. Strategy #4904
3. Strategy #7082

Qu'en est-il de l'aspect coalitionnel du jeu ?

Les expérimentations ont été faites en utilisant la deuxième définition de coalition : deux joueurs sont dans une même coalition s'ils ne se sont pas attaqué ni directement (i) ni indirectement (ii). D'après ce que nous venons de voir, les bonnes stratégies doivent être gentilles et réactives, donc elles ne devront pas attaquer n'importe quel joueur. La condition (i) sera respecté puisque les joueurs ne s'attaqueront pas l'un l'autre sans raison. On peut donc penser que l'aspect coalitionnel du jeu sera utilisé par ces stratégies.

Les résultats montrent en effet que les stratégies finissant parmi les premières dans un jeu sont celles qui apparaissent le plus souvent parmi les coalitions formées dans la partie. Ce résultat vient aussi du fait que ces stratégies jouent plus de tours que les autres, il est donc normal qu'elles apparaissent plus souvent dans les coalitions.

Les tests nous laissent donc penser également que les bonnes stratégies utilisent l'aspect coalitionnel du jeu.

En moyenne, les parties se terminent après 2000 tours mais il existe des parties qui se terminent en moins de 1000 tours et d'autres qui ne sont pas encore terminées après 20000 tours. On peut se poser la question de savoir s'il existe une partie qui ne se termine jamais.

Cette recherche a été faite et un jeu qui semble ne pas s'arrêter a été trouvé. La descrition de ce jeu est fournie en annexe page 95. Le graphe du jeu jusque 100000 tours est en annexe figure 7.2. On remarque que le dernier joueur qui quitte la partie perd au tour 38000 environ est qu'ensuite les joueurs évoluent dans une fourchette de 20 points mais aucun ne se démarque des autres. J'ai tenté de tester ce jeu sur 1000000 tours mais même en augmentant la taille mémoire de la machine virtuelle au maximum, le programme s'est arrêté après environ 200000 tours. Mais après ces 200000 tours, la situation de jeu était encore exactement la même. Malheureusement, je n'ai détecté aucun cycle dans cette partie. Nous verrons dans le test suivant qu'il existe des parties qui ne s'arrêtent jamais.

6.4.2 Série 2

Dans cette seconde série d'expérimentations, nous testons des stratégies plus complexes directement l'une contre l'autre.

Tout d'abord, les résultats montrent que la stratégie purement aléatoire perd toujours contre toutes les autres. Ce résultat était prévisible mais il est important de le préciser.

On remarque dans ces tests les bons résultats des stratégies persistentes. Ces stratégies attaquent toujours des joueurs qui ont déjà été attaqué auparavant. Comme ces joueurs ont déjà été plus ou moins souvent attaqués, il y a de fortes chances pour qu'ils aient un peu moins de ressources que les autres. Donc ces stratégies réussissent plus régulièrement leurs attaques et donc récupèrent des ressources.

Les stratégies basées sur les coalitions produisent également des bons résultats. En effet, dès que quelques tours ont été joués, les stratégies jouent avec les coalitions formées et arrivent à les maintenir. Ces stratégies réussissent à gagner lorsque la taille de la coalition qu'elle a réussi à maintenir est suffisamment grande.

On remarque également que les stratégies rancunières parviennent à se grouper sous forme de coalitions. En effet, elles ne s'attaquent pas, elles demandent du soutien aux joueurs qui ne l'ont pas attaqué et elles attaquent directement et indirectement les joueurs qui les ont déjà attaqué. On voit que les premières conditions des définitions de coalitions sont respectées par ce comportement et nous voyons donc des coalitions émerger.

Enfin, la stratégie PowerStrategy qui est la plus offensive, produit elle aussi de bons résultats. Cette stratégie attaque le plus faible des joueurs, demande de l'aide à tout le monde et accepte toujours de supporter les attaques. Il est donc clair qu'elle contribue aux succès des attaques et donc à faire baisser les ressources de ses adversaires.

J'aurais aimé pouvoir présenté un tableau représentant les résultats de chaque stratégie en indiquant le nombre de victoires de chacune mais le manque de temps m'en a empêché.

Ce qui ressort de ces expérimenations est également le coût en terme de temps de calcul. En effet, pour que les tests soient pertinents, il faut en effectuer énormément. Cela implique un temps CPU très important et une quantité de données très volumineuse à analyser par la suite.

Beaucoup de jeux infinis qui ont été trouvés lors de ces tests. En particulier les oppositions entre stratégies basées sur les coalitions ne se terminent pas toujours. Le phénomène s'explique facilement : au premier tour c'est la stratégie d'attaque aléatoire qui est employée et comme il n'existe pas encore de coalition, ces stratégies ne demandent du soutien à aucun joueur et même n'accepteraient aucune demande de soutien. Au total, 10 parties infinies ont été trouvées quelle que soit la valeur de la graine. Nous avons donc montré qu'il existait des parties pour lesquelles ce jeu ne se termine jamais.

Ces jeux infinis sont particuliers car aucune attaque ne réussit mais on peut en imaginer pour lesquels les attaques réussissent. En effet, si une stratégie attaque le premier joueur de la liste puis le second, etc... de manière cyclique et demande de l'aide au second joueur puis au troisième,etc... de manière cyclique également et qu'elle accepte toujours de supporter les attaques, alors les instances de jeu composées uniquement de cette stratégies ne se termineront jamais. Ce type de jeu peut être appelé des oscillateurs.

Il existe une relation forte entre ces jeux infinis et le domaine de la vie artificielle.

Le jeu de la vie, introduit par Conway, se présente sous la forme d'une grille ; chaque cellule possède deux états : activé et désactivé (vivant ou mort). Des règles permettent de savoir, après chaque génération, si une cellule active devient inactive ou non et si une cellule désactivé devient active ou pas. La partie se termine lorsqu'il ne reste plus aucune cellule vivante. Dans ce jeu, il existe également des oscillateurs ; c'est-à-dire une configuration de départ telle que la partie ne se termine jamais. De plus, à chaque oscillation, une ou plusieurs nouvelles cellules peuvent être créées et activées. L'existence de telles configurations montre que l'on peut simuler des comportements émergents à partir de simulations. Le parallèle avec les jeux infinis est clair : à partir d'une configuration de départ, il est possible de générer des simulations dans lesquelles de nouvelles actions sont effectuées à chaque tour et qui sont infinies. De plus, le jeu de la vie a la propriété d'être un système chaotique. C'est-à-dire qu'une très légère modification de la configuration de départ peut modifier radicalement l'issue de la partie. Il semble que ce soit également une propriété de ce jeu. En effet, on a vu que le fait de modifier la graine change radicalement le résultat du jeu. Il est vrai que la modification de la graine n'est pas un changement mineur pour tous les jeux. Mais nous avons également vu que le fait de rajouter un joueur supplémentaire dans une partie de 30 joueurs a changé complètement l'issue du jeu. D'autres tests seraient nécessaires pour pouvoir affirmer ou non que ce jeu est un système chaotique. J'aurais souhaité étudier plus en profondeur cette relation et établir les liens entre la théorie des jeux et la vie artificielle mais le temps m'a manqué.

Dans ce chapitre, nous avons détaillé les objectifs que nous voulions atteindre et les expérimentations qui ont été réalisées pour les vérifier. Ensuite nous avons expliqué les résultats obtenus. Nous avons montré l'importance de l'aspect coalitionnel et nous avons défini ce que pouvaient être les bonnes stratégies. Enfin, nous avons mis en évidence l'existence de jeux infinis.

Dans le chapitre suivant, nous allons revenir à un cadre théorique sur les coalitions. En effet, les recherches dans ce domaine peuvent nous aider à mieux comprendre les résultats obtenus grâce les expérimentations sur les coalitions.

Chapitre 7

Structures de coalition

Dans ce chapitre, nous allons plus particulièrement nous intéresser aux concepts de solutions des jeux à n joueurs. D'une manière générale, un concept de solution nécessite une stabilité par rapport aux déviations des joueurs au sein des coalitions. Nous verrons en particulier les notions d'équilibre et de stabilité et les concepts de solutions du coeur, de la valeur de Shapley et du nucleolus.

7.1 Définition

Les structures de coalition désignent la façon dont les coalitions sont formées. Le but est de trouver des structures dites "stables" c'est-à-dire une formation de coalitions telle qu'aucun des joueurs n'a d'intérêt à quitter celle à qui il appartient.

Définition 47 *Une structure de coalition est une partition* $B = \{B_1, B_2, ..., B_n\}$ *des n joueurs et tel que* $B_i \cap B_j = \emptyset, \forall i \neq j$ *(c'est-à-dire qu'un joueur ne peut pas faire partie de deux coalitions).*

Notons $A(S)$ l'ensemble des actions "faisables" pour la coalition S. L'utilité espérée par un joueur i d'une coalition S pour l'action $a \in A(S)$ est donné par $\mu_i(a, S)$. La fonction caractéristique est $V(S) = \{x \in I\!R^N$ tel que $\exists a \in A(S)$ tel que $x^i \leq \mu_i(a, S), \forall i \in S\}$. Le problème de cette structure pour une coalition S est qu'elle est totalement indépendante des coalitions de $N \setminus S$ et de leurs choix, ce qui la rend inapropriée pour certains jeux. En effet, les actions disponibles $A(S)$ par une coalition S peuvent dépendre des coalitions adverses.

Les gains possibles pour une structure de coalition B peuvent s'exprimer $X(B) = \cap\{x \in V(S)$ tel que $S \in B\}$ et $X(B) = \{x \in I\!R^N$ tel que $\forall S \in$

$B, \sum_{i \in S} x_i \leq v(S)\}$ pour les jeux avec tranfert d'utilité. Cependant, cette définition pose problème pour certains jeux (en particulier lorsqu'on autorise le transfert d'utilité entre coalitions).

7.2 Stabilité

Considérons (pour le moment) le cas où seulement un joueur est autorisé à joindre une coalition dans la structure. Dans cette restriction, on peut définir un **équilibre individuellement stable**. Cet équilibre est atteint si aucun joueur ne peut changer de coalition tel que ce soit bénéfique pour lui ou pour tous les membres de la coalition qu'il joint.

Formellement :

Définition 48 *Si (N, V) est un jeu sous forme coalitionelle, (x, B) où B est une partition et $x \in X(B)$ est un équilibre individuellement stable s'il n'existe pas de coalition $S \in B \cup \{\emptyset\}$ et $i \in N \setminus S$ tel que x appartient à l'ensemble $V(S \cup i)$.*

Donc si un jeu est superadditif alors il existe toujours un équilibre individuellement stable. En fait, tout couple composé d'un optimum de Pareto et d'un gain individuel rationnel x, constitue un équilibre individuellement stable.

Une extension de cet équilibre est **l'équilibre contractuel individuellement stable** qui requiert qu'un joueur ne peut changer de coalition quand cela est bénéfique pour lui, pour tous les membres de la coalition qu'il joint et celle qu'il quitte. Les contrats (qui empêchent un joueur de quitter une coalition si les membres s'y opposent) et les transferts (autorisant une compensation positive ou négative aux membres restants) génèrent une stabilité individuelle des partitions. Pour donner un exemple de ce type de changement de coalition, considérons un employé qui souhaite changer d'entreprise pour une autre proposant une offre plus attractive. Dans ce cas, le changement est bénéfique pour lui et pour l'entreprise qu'il joint. Considérons maintenant un joueur de football qui souhaite quitter son club pour un autre meilleur et qui propose un salaire supérieur. Le premier ne souhaitant pas voir ce joueur quitter le club se voit proposer un compensation de la part du second. Dans ce cas, le changement est bénéfique pour le joueur, le club qu'il quitte et celui qu'il rejoint.

7.3 Coeur d'une structure de coalition

Avant de définir formellement le concept de solution du coeur, il est nécessaire d'introduire d'autres notions.

Nous avons vu que les imputations sont des vecteurs de gains pour les joueurs. Or, il existe une infinité d'imputations possibles. Il est évident que toutes ne satisfont pas chaque joueur : certaines vont avantager certains joueurs et en dénigrer d'autres. Le concept de coeur vise précisement à définir des imputations qui n'en désavantagent aucun.

Définition 49 *Une imputation x est instable pour une coalition S si $v(S) > \sum_{i \in S} x_i$. On dit que x est instable si il existe une coalition S tel que x soit instable pour la coalition S. x est stable sinon.*

Nous pouvons alors définir le coeur :

Définition 50 *L'ensemble C des imputations stables est appelé le coeur. Posons ζ l'ensemble des coalition. Donc*

$$C = \{x = (x_1, ..., x_n) \text{ tel que } \sum_{i \in N} x_i = v(N) \text{ et } \sum_{i \in S} x_i \geq v(S) \forall S \in \zeta\}$$

Le coeur est un concept de solution intéressant mais il peut être vide. Cela signifie qu'il n'existe aucune imputation satisfaisant tous les joueurs. Une condition nécessaire et suffisante pour que le jeu (N, V) ait un coeur de structure de coalition est que la couverture super-additive du jeu $(N, \bar{V})$ a un coeur non vide.

Comment trouver le coeur d'un jeu ? Il existe une méthode basée sur l'algorithme du simplexe. L'ensemble des imputations peut être représenté par une figure convexe.

Considérons, le jeu de l'exemple 7.1, défini par sa fonction caractéristique.

$$\begin{array}{llll} & v(\{1\}) = 1 & v(\{1,2\}) = 4 & \\ v(\emptyset) = 0 & v(\{2\}) = 0 & v(\{1,3\}) = 3 & v(\{1,2,3\}) = 8 \\ & v(\{3\}) = 1 & v(\{2,3\}) = 5 & \end{array}$$

TAB. 7.1 – Exemple de jeu sous forme coalitionnelle

Dans ce jeu, les imputations sont les points (x_1, x_2, x_3) tels que $x_1 + x_2 + x_3 = 8$ et $x_1 \geq 1$, $x_2 \geq 0$ et $x_3 \geq 0$. Nous pouvons contruire un triangle basé sur les points $(7, 0, 1)$, $(1, 6, 1)$ et $(1, 0, 7)$ qui représente l'ensemble des points satisfaisant ces conditions. On place ces points avec des coordonnées barycentriques. On pose que le plan est défini sur l'équation $x_1 + x_2 + x_3 = 8$ et en donnant à chaque point du plan des coordonnées dont la somme est égale à 8.

On obtient le triangle représenté figure 7.1

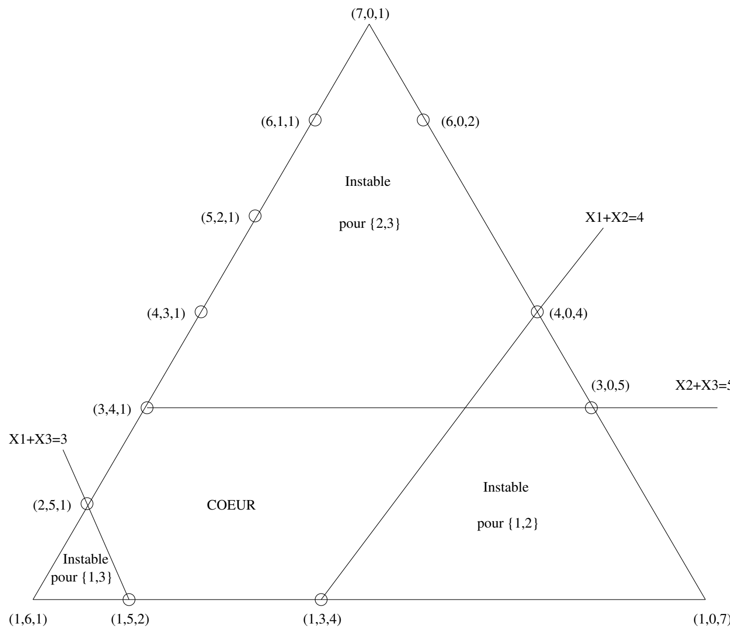

FIG. 7.1 – Recherche du coeur

La méthode consite à éliminer dans le plan les points appartenant à l'ensemble des imputations instables. Tout d'abord, la coalition $\{2,3\}$ est garantie d'obtenir $v(\{2,3\}) = 5$ donc les points tels que $x_2 + x_3 < 5$ représentent des imputations instables : on peut les supprimer. Ensuite, on a $v(\{1,2\}) = 4$ donc nous supprimons les points du plan tels que $x_1 + x_2 < 4$. Enfin, comme $v(\{1,3\}) = 3$, on supprime les points tels que $x_1 + x_3 < 3$.

Ainsi, nous il ne reste que les points du plan représentant des imputations stables : nous avons défini le coeur du jeu.

On notera, en outre, ce théorème :

Théorème 1 *Le coeur d'un jeu essentiel à somme constante est vide.*

Le lecteur pourra trouver la démonstration dans [12].

Une classe de jeux pour lesquelles il existe un coeur est celle des **jeux consécutifs**. Les joueurs d'un tel jeu peuvent être numérotés de telle manière qu'un groupe de joueurs peut bénéficier d'une coopération si et seulement si il est consécutif, c'est-à-dire si tous les joueurs qui sont entre le plus petit et le plus grand joueur (selon la numérotation) du groupe sont des membres du groupe.

Formellement :

Définition 51 *Une coalition S est consécutive si pour tout triplet de joueurs i, j, k tel que $i < j < k$ si i et $k \in S$ alors $j \in S$. La structure consécutive* *de S, noté $C(S)$, est l'unique partition de S dans les coalitions consécutives qui sont maximales, c'est-à-dire l'unique partition de S dans les coalitions consécutives qui contient le nombre minimum de coalitions. Donc $C(S) = \{S\}$ si et seulement si S est consécutive. Le jeu consécutif (N, V_c) d'un jeu (N, V) est donné par : $V_c(S) = \cap\{V(T) \setminus T \in C(S)\}$ c'est-à-dire $x \in I\!R^N \in V_c(S)$ si $x \in V(T)$ pour toute coalition consécutive $T \in C(S)$.*

Mais il existe également des jeux non consécutifs pour lesquels il existe un coeur.

Supposons que les n joueurs ont des préférences sur un ensemble d'alternatives disponibles noté Ω et qu'ils peuvent voter pour celles qu'ils préfèrent. Alors les groupes de joueurs vont pouvoir élire leur(s) stratégie(s) préférée(s). Pour faire ce choix, on peut considérer une méthode avec standard fixé (si les candidats dépassent une certaine valeur fixée alors ils sont élus) ou une méthode à nombre fixé. Cette dernière est plus utilisée : on définit un certain nombre (K) de sièges disponibles (de candidats élus).

Dans le cas du vote avec une méthode avec standard fixée, on pose $m \leq n$ le quota (le nombre de votes) que doit atteindre une alternative pour être élue.

Définition 52 *Un m-**équilibre** est un ensemble d'alternatives tel que quand les votants doivent choisir parmi elles et que chacun vote pour celle qu'il préfère, chaque alternative recoit au moins m votes et aucune autre alternative ne peut en avoir autant.*

Définition 53 *Soit $A \subset \Omega$, le **support d'une alternative** $a \in \Omega$ étant donné A et noté $S(a, A)$, est l'ensemble des agents qui préfèrent strictement a à toute autre alternative.*

On a donc $S(a, A) = \{i \in N \setminus \mu_i(a) > \mu_i(b), \forall b \in A, b \neq a\}$. $A \subset \Omega$ est un m-équilibre si pour tout $a \in \Omega, |S(a, A)| \geq m$ si et seulement si $a \in A$. Pour chaque société N, chaque ensemble fini d'alternatives Ω et tout quota $m \leq n$, il existe un $m-$équilibre (Greenberg & Weber). Tout $m-$équilibre appartient à un coeur de structure de coalition.

Définition 54 *Dans le cas du vote avec une méthode à nombre fixé, un ensemble $A \in \Omega$ d'alternatives est un $K-$équilibre si A contient exactement K alternatives et $\forall b \in \Omega \setminus A, |S(b, A \cup \{b\})| \leq |S(a, A \cup \{b\})|, \forall a \in A$.*

Toutes les sociétés N (pour $K \geq 2$) n'ont pas nécessairement de $K-$équilibre, cependant si on pose $\varphi(2) = 7$ et $\varphi(K) = 2K + 1$ si $K \geq 3$, alors la société N a un $K-$équilibre si et seulement si $n \leq \varphi(K)$.

7.4 Valeur de Shapley

La valeur de Shapley (ou Shapley-Shubik (1953)) cherche à donner une valeur pour chacun des joueurs, qui représente son pouvoir de négociation ou le gain qu'il peut obtenir dans un jeu coopératif. Supposons que l'on ait deux joueurs : $N = \{1, 2\}$: les gains dûs à la coopération sont donc $v(\{1,2\}) - v(\{1\}) - v(\{2\})$. Soit $Sh_i(N)$ la valeur de Shapley pour le joueur i parmi les N joueurs, alors $Sh_i(N) = v(\{i\}) + \frac{1}{2}[v(\{1,2\}) - v(\{1\}) - v(\{2\})]$. Plus généralement, la contribution apportée par le joueur i lorsqu'il joint une coalition S est $v(S) - v(S \setminus \{i\})$. Cette contribution apparait pour tout ordre dans lequel i est précédé par les $s-1$ème autres joueurs de S et suivi par les $n-s$ autres qui ne sont pas dans S où s est la taille de S. Le nombre d'ordres pour lesquels ces conditions sont respectées est $(s-1)!(n-s)!$. Si par exemple, on donne $Sh_i(\{i\}) = v(\{i\})$ alors le joueur 1 gagne autant par la coopération avec 2 que le joueur 2 avec 1.

Définition 55 *La valeur de Shapley pour un joueur i dans un jeu à n joueurs est donné par* $Sh_i(N) = \frac{1}{n!}\sum_{i \in S}(s-1)!(n-s)!(v(S) - v(S \setminus \{i\})$

La valeur de Shapley est donc la moyenne des contributions marginales de i sur tous les ordres possibles. On peut préciser que dans le cas des jeux à somme pondérée, la somme des valeurs de Shapley pour chaque joueur est égale à 1.

Considérons, un premier exemple : soient quatre joueurs possédant respectivement 10, 20, 30 et 40% des actions dans une société. Les décisions sont prises en respectant la majorité en terme d'actions. Nous définissons le jeu de vote pondéré correspondant : $[50 : 10, 20, 30, 40]$. Les coalitions gagnantes sont : $\{2,4\}$, $\{3,4\}$, $\{1,2,4\}$, $\{1,3,4\}$, $\{3,2,4\}$, $\{1,2,3\}$ et $\{1,2,3,4\}$. Les valeurs de Shapley des joueurs sont :

$$Sh_1(v) = \frac{1}{12}, Sh_2(v) = \frac{1}{4}, Sh_3(v) = \frac{1}{4}, Sh_4(v) = \frac{5}{12}$$

Nous avons bien $\sum_{i=1..4} Sh_i = 1$. On remarque que les joueurs 2 et 3 ont exactement le même pouvoir de négociation bien que 3 ait plus d'actions. En effet, le nombre de coalitions gagnantes contenant 2 est le même que le nombre de coalitions gagnantes contenant 3.

Maintenant reconsidérons l'exemple 7.1.

Dans ce jeu, après calcul, on trouve :

$$Sh_1(v) = \frac{14}{6}, Sh_2(v) = \frac{17}{6}, Sh_3(v) = \frac{17}{6}$$

Que peut-on déduire de cet exemple et de la comparaison avec les valeurs du coeur trouvées précedemment ?

Tout d'abord, la valeur de Shapley existe toujours contrairement au coeur qui peut être vide. Dans cet exemple, les valeurs de Shapley définissent un point $(\frac{14}{6}, \frac{17}{6}, \frac{17}{6})$ qui appartient au coeur du jeu. Cependant, ce n'est pas une généralité : la valeur de Shapley n'appartient pas nécessairement au coeur du jeu, et ce, même si le coeur est non vide.

La valeur de Shapley est également appelée indice de Shapley. Mais il existe d'autres notions d'indices permettant de mesurer le pouvoir de négociation des joueurs. Ils sont essentiellement utilisés dans le cadre des jeux de votes : on peut citer les indices de Banzhaf, de Johnston, de Deegan-Packel, de Holler-Packel, de Curiel, de Colomer-Martinez, d'Andjiga-Berg et de Chakravarty.

Tous sont similaires, car de la forme $\sum_{S \in X, i \in S} f(i,S)(v(S) - v(S \setminus \{i\})$ où X est l'ensemble des coalitions admissibles et $f(i,S)$ est une fonction dépendant de la coalition S et du joueur i induisant un poids. Les avis sur la qualité de ces indices sont partagés et les comparaisons de ces différents indices sur plusieurs types de jeux sont encore à l'étude.

Il existe une extension de la valeur de Shapley pour une structure de coalition B appelée $B - valeur$.

Définition 56 *La $B - valeur$ pour un joueur $i \in B_k \in B$ coïncide avec la valeur de Shapley du joueur i dans le sous jeu (B_k, V). Donc, le gain redistribué par la $B - valeur$ aux membres de B_k est $v(B_k)$ (Aumann & Dreze).*

Une structure de coopération est un graphe où chaque noeud est un joueur et tel qu'il existe un arc entre ces noeuds si les deux joueurs peuvent communiquer. Dans ce modèle, une structure de coalition est donc un graphe dans lequel il existe un arc entre deux joueurs si et seulement s'ils appartiennent à la même coalition B_k. Notons G une structure de coopération, Myerson a étendu la valeur de Shapley à une structure de coopération quelconque, appelée **valeur de Myerson** et noté $\mu(G,V)$. Soit $S(G)$ la partition de $S \subset N$ induite par G et (N,V) un jeu (avec utilité transférable). On définit le jeu (N, V_G) où $V_G(S) = \sum_{\tau \in S(G)} V(T)$ alors $\mu(G,V)$ est la valeur de Shapley du jeu (N, V_G).

Définition 57 ***La valeur coalitionnelle de Shapley*** *assigne à chaque structure de coalition B un vecteur de gains $\varphi(B)$ dans $I\!R^N$ tel que $\sum_{i \in N} \varphi^i(B) = v(N)$.*

7.5 Notion de nucleolus

Un autre concept a été introduit par Schmeidler en 1969 comme concept de solution d'un jeu à n joueurs : le nucleolus. Dans ce concept, on essaie de trouver une imputation qui minimise une certaine inégalité. Chaque coalition tente de minimiser son insatisfaction maximum. Pour cela, on définit l'excès comme mesure d'inégalité d'une imputation pour une coalition.

Définition 58 *L'excès d'une imputation x pour une coalition S est donné par*

$$e(x, S) = v(S) - \sum_{j \in S} x_j$$

L'excès mesure la quantité perdue par une coalition S par une imputation x par rapport à son potentiel $v(S)$. Comme le coeur est défini par les imputations telles que $\sum_{i \in S} x_i \geq v(S)$ pour toute coalition S, on déduit qu'une imputation x est dans le coeur si et seulement si tous les excès sont négatifs ou nuls. Le but du nucleolus est de réduire au maximum cet excès pour toutes les coalitions.

Pour illustrer ce concept, on va considérer l'exemple suivant.

Une société doit de l'argent à trois créditeurs : elle doit 10000 au créditeur A, 20000 à B et 30000 à C. Mais la société n'a que 36000 pour les rembourser. Comment doit-elle partager cet argent.

Pour simplifier, nous allons indiquer les valeurs en milliers. Ainsi 6 signifiera 6000 par exemple. Le pro rata indique qu'elle devrait donner 6 à A, 12 à B et 18 à C, ce qui défini l'imputation $x = (6, 12, 18)$.

Tout d'abord, il faut trouver la fonction caractéristique v du jeu. Trivialement, on a $v(\emptyset) = 0$ et $v(ABC) = 36$. A peut ne rien recevoir s'il agit seul, donc $v(A) = 0$. De même, B agissant seul peut ne rien recevoir donc $v(B) = 0$. Quant à C, au pire il recoit 6 (car A et B ne peuvent avoir que 30), donc $v(C) = 6$. On définit facilement les valeurs des coalitions de deux joueurs : c'est la somme restante lorsque le troisième recoit l'intégralité de ce qui lui est dû : $v(AB) = 36 - 30 = 6$, $v(BC) = 36 - 10 = 26$ et $v(AC) = 36 - 20 = 16$.

Posons (x_1, x_2, x_3), l'imputation par laquelle 1 recoit x_1, B recoit x_2 et C recoit x_3 telle que $x_1 + x_2 + x_3 = 36$.

Nous allons procéder de manière similaire à celle de l'algorithme du simplexe. Nous allons partir d'un point arbitraire et maximiser la valeur des x_i. Pour nous aider, nous allons récapituler les valeurs trouvées dans le tableau 7.2.

Tout d'abord, le "point de départ" choisi est le point de pro rata $x = (6, 12, 18)$. Le vecteur définissant les excès est $(-6, -12, -12, -12, -8, -4)$

S	$v(S)$	$e(x,S)$	(6,12,18)	(5,12,19)	(5,10.5,20.5)	(6,11,19)
A	0	$-x_1$	-6	-5	- 5	-6
B	0	$-x_2$	-12	-12	-10.5	-11
C	6	$6-x_3$	-12	-13	-14.5	-13
AB	6	$6-x_1-x_2$	-12	-11	-9.5	-11
AC	16	$16-x_1-x_3$	-8	-8	-9.5	-9
BC	26	$26-x_2-x_3$	-4	-5	-5	-4

TAB. 7.2 – Tableau récapitulatif

représentant les excès respectivement pour A, B, C, AB, AC et BC. La coalition BC est la moins satisfaite par ce partage (-4 est l'excès le plus élevé). Alors nous allons chercher à maximiser $x_2 + x_3$ ce qui revient à minimiser x_1. Or, en diminuant l'excès de BC, on augmente celui de A et ces excès coïncident quand $x_1 = 5$ ce qui donne la valeur -5 pour A. Donc on fixe $x_1 = 5$. Il reste x_2 et x_3 qui peuvent varier tels que $x_2 + x_3 = 36 - 5 = 31$. Si on pose $x = (5, 12, 19)$ alors l'excès le plus élevé (en ne considérant plus A et BC que nous avonc traitées précedemment) est -8 pour AC. Nous devons donc diminuer x_2 (ou augmenter x_3) ce qui a pour effet d'augmenter les excès de B et de AB. On a donc $e(x, AB) = e(x, AC) = 9.5$ pour $x_2 = 10.5$ et $x_3 = 20.5$.

Le nucleolus est donc $(5, 10.5, 20.5)$.

Pour comparer cette solution avec la valeur de Shapley, voici les valeurs de Shapley pour ce jeu.

$$Sh_A = 6, Sh_B = 11, Sh_C = 19$$

Les excès pour les valeurs de Shapley sont indiqués à la dernière colonne du tableau.

Pour définir le nucleolus de manière formelle, il faut définir un ordre sur les vecteurs d'excès.

Soit $\Theta(x)$ le vecteur d'excès de x ordonné dans l'ordre décroissant. Par exemple, pour l'imputation du pro rata définie plus haut $x = (6, 12, 18)$ nous avons $(-6, -12, -12, -12, -8, -4)$ et $\Theta(x) = (-4, -6, -8, -12, -12, -12)$. Sur les vecteurs $\Theta(x)$ on utilise l'ordre lexicographique. On dit qu'un vecteur $y = (y_1, ..., y_k)$ est lexicographiquement plus petit qu'un vecteur $z = (z_1, ..., z_n)$ (noté $y <_L z$) si $y_1 < z_1$ ou $y_1 = z_1$ et $y_2 < z_2$, ou $y_1 = z_1$ et $y_2 = z_2$ et $y_3 < z_3$, etc... L'ordre large est défini de manière classique : $y \leq_L z$ si $y <_L z$ ou $y = z$. Le nucleolus est une imputation qui minimise $\Theta(x)$ selon l'ordre lexicographique.

Définition 59 *Soit $X = \{x \text{ tels que } \sum_{j=1}^{n} x_j = v(N)\}$ l'ensemble des imputations. On dit qu'un vecteur $v \in X$ est un nucleolus si pour tout $x \in X$ on* a $\Theta(v) \leq_L \Theta(x)$.

Le nucleolus possède plusieurs propriétés intérressantes.

Tout d'abord, le nucleolus d'un jeu sous forme coalitionnelle existe toujours et est unique. Ensuite, il est rationnel pour les groupes et individuellement rationnel. Enfin, si le coeur du jeu est non vide alors le nucleolus en fait partie.

Le principal inconvénient de ce concept est la relative difficulté de calcul. En effet, il s'agit de résoudre un problème de minimisation du maximum d'un ensemble de fonctions linéaires soumises à des contraintes linéaires. Il s'agit donc d'un problème de programmation linéaire que nous pouvons résoudre par l'algorithme du simplexe.

L'exemple étudié était un jeu à trois joueurs ce qui rend le calcul encore relativement simple. Cependant, quand le nombre de joueurs devient plus grand, les calculs deviennent trop complexes. On peut alors avoir recours à un programme de résolution par l'algorithme du simplexe.

Les expérimentations montrent qu'aucun des concepts de solution théorique n'est complétement convaincant, mais que les solutions données peuvent être considérées comme "bonnes".

Dans ce dernier chapitre, nous avons vu les méthodes théoriques issues de la recherche pour résoudre les jeux à n joueurs. Comme pour les jeux à deux joueurs, nous avons mis en évidence le fait qu'aucune d'entre elles ne donne la solution du jeu dans le cas général. Le coeur n'existe pas toujours, la valeur de Shapley n'est qu'une mesure (parmi d'autres) de la "force coalitionnelle" des coalitions et enfin le nucleolus ne donne des résultats que pour les imputations.

Il n'existe donc pas de méthode permettant de résoudre les jeux à n joueurs et, dans ce cas, les expérimentations semblent être une méthode de recherche de bonnes stratégies et de meilleures coalitions de laquelle on peut déduire des résultats applicables de manière plus sûre. Cependant, nous avons vu qu'il faut beaucoup de tests et beaucoup de stratégies pour que les conclusions soient pertinentes.

Conclusion

Au cours de ca travail nous avons donc étudié l'aspect théorique de la théorie des jeux puis l'aspect pratique à travers un jeu à n joueurs.

Dans un premier temps, nous avons introduit le théorie des jeux. Nous avons définit les notions de bases utilisées par cette théorie : la rationnalité, la théorie de l'utilité et la représentation des jeux. Nous avons vu qu'elle englode un champ d'application très vaste et que les recherches dans ce domaine sont nombreuses.

Dans un second temps, nous nous sommes intéressé aux méthodes de résolutions appliquées aux jeux à deux joueurs. L'équilibre de Nash est au centre de ces méthodes mais nous avons vu qu'aucune ne permettait de résoudre un jeu dans le cas général.

Dans le troisième chapitre, nous avons décrit une méthode recherche de bonnes stratégies basée sur des simulations. Il s'agit de tester les stratégies sur ordinateur avec un simulateur et c'est l'interprétation des résultats qui permet de conclure quelles sont les meilleures stratégies.

Dans le quatrième chapitre, nous avons défini les notions utilisées pour travailler sur les jeux à n joueurs. Après une comparaison avec les jeux à deux joueurs, nous avons défini les stratégies basées sur le comportement mais nous avons surtout la forme coalitionnelle. Cette représentation est en effet très souvent utilisée lors des études sur les jeux à n joueurs. Nous avons ensuite énoncer plusieurs définitions et propriétés liées à la forme coalitionnelle.

Dans le cinqième chapitre, nous décrivons le jeu proposé et nous avons définit les notions utilisées lors de l'analyse. Nous avons ensuite décrit le simulateur qui a été codé et les stratégies implémentées qui ont été utilisées lors des expérimentations.

Dans le sixième chapitre, nous avons détallé les moyens utilisés pour réaliser les expérimentations puis nous avons expliqué les objectifs de celles ci. Ensuite nous avons décrit les séries de tests effectuées puis nous avons analysé les résultats obtenus. Nous avons expliqué le rôle des coalitions et établit des liens entre les meilleures stratégies pou ce jeu et les critères d'Axelrod.

Enfin, dans le septième et dernier chapitre, nous sommes revenu dans le cadre théorique pour définir les méthodes existantes permettant de résoudre les jeux à n joueurs. Nous avons étudié le coeur, la valeur de Shapley et le nucleolus. Nous avons montré qu'aucune ne permettait de résoudre un jeu dans le cas général.

Ce stage a donc permis de commencer l'étude d'un jeu à n joueurs complexe et de concevoir un simulateur. Ce simulateur a été utilisé pour réaliser des expérimentations et apporter les premiers résultats sur ce jeu. En particulier, nous avons posé les bases de ce que devait être une bonne stratégie et le travail a permis de montrer qu'elles correspondaient globalement aux critères d'Axelrod. Nous avons également montré l'importance des coalitions dans le jeu et pour les stratégies. Nous avons aussi effectué quelques recherches sur les jeux infinis. Enfin, nous avons commencé à discuter des liens entre la théorie des jeux et la vie artificielle.

Perspectives

Le travail restant à effectuer est important.

Tout d'abord le travail sur le simulateur peut être continué. En effet, on peut imaginer une quantité de petites améliorations. On peut créer d'autres courbes représentant des statistiques sur le jeu, par exemple sur le nombre de demandes de soutien ou sur les acceptations de soutien. On peut également sauvegarder plus de données sur le jeu à chaque partie sous forme de fichier par exemple, sur les demandes de soutien. L'analyse de ces données pourraient nous apporter des informations sur les alliances qui se forment au sein du jeu.

Il y a une infinité de stratégies définissables et il faudrait pouvoir en tester un maximum. Pour cela il faudrait définir d'autres sous-stratégies pour créer de nouvelles stratégies. En effet, plus il y aura de stratégies testées et plus les tests seront pertinents.

Il y aurait énormément plus de tests à effectuer pour découvrir les meilleures stratégies et les définir. Des tests sur chaque définition de coalitions seraient également utiles. De cette façon, on pourrait voir quelle définition est la plus interressante et en tirer des conclusions. De plus, des tests supplémentaires nous permettraient de découvrit plus précisemment le rôle des coalitions dans le jeu. D'autres tests peuvent être réalisés pour découvrir la présence d'autres d'oscillateurs ou de parties infinies. Il serait interressant de trouver le plus petit oscillateur (en terme du nombre de joueur ou en terme de fréquence d'oscillations) ou encore celui qui oscille le plus tôt dans le jeu.

Une recherche plus approfondie sur les liens qui peuvent exister entre ce jeu et le domaine de la vie artificielle serait vraiment interressante. En effet, il est possible que certaines applications de la vie artificielle puissent s'appliquer en théorie des jeux et qu'elles soient utilisables dans notre jeu et réciproquement. Le travail peut s'orienté sur les oscillateurs ou sur le principe d'émergence. On peut également effectuer quelques recherches pour montrer si le système est chaotique ou non.

L'analyse d'une telle quantité de données sous forme de fichiers est vraiment laborieuse ; donc la conception et la maintenance d'une base de données robuste serait un outil appréciable. La récupération des données sur un jeu est simplifié, le calcul de statistiques entre les parties également. En particulier l'étude de coalitions redondantes sur plusieurs parties serait interessante. La recherche des stratégies qui s'unissent régulièrement sous forme de coalitions serait aussi une information précieuse qui ne peut être découverte rapidement qu'à partir d'une base. Enfin, cette base peut être utilisée pour générer d'autres fichiers proposant des moyennes, des statistiques et des résumés sur des parties au format HTML par exemple pour les publier facilement.

Annexes

Déscription d'un jeu infini ?

Player #0 Strategy #42270
Attaque :
1 : AttackTheMostAttackedPlayerStrategymémoire = 1 NEGATIVE
2 : AttackRandomlyStrategy
Asking :
1 : AskingPlayersNeverRefuseToHelpMeStrategy mémoire = 1 NEGATIVE
Acceptation de soutien :
1 : AcceptOneByTwoStrategy

Player #1 Strategy #42271
Attaque :
1 : AttackTheMostSuccessfulyAttackedPlayerStrategy mémoire = 1
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersDontAttackMeStrategy2 mémoire = 10 NEGATIVE
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy2 mémoire = 0 NEGATIVE

Player #2 Strategy #42272
Attaque :
1 : AttackTheMostAttackedPlayerStrategy mémoire = 10 NEGATIVE
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingStrongestThanMeStrategy mémoire = 10 NEGATIVE
Acceptation de soutien :
1 : AlwaysAcceptToSupportStrategy NEGATIVE

Player #3 Strategy #42273

Attaque :
1 : SpitefulAttackStrategy2 mémoire = 0
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingNoBodyStrategy
Acceptation de soutien :
1 : AcceptAskingPlayersWeakestThanMeStrategy mémoire = 0 NEGATIVE

Player #4 Strategy #42274
Attaque :
1 : AttackWeakestStrategy NEGATIVE
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersDontAttackMeStrategy2 mémoire = 0 NEGATIVE
Acceptation de soutien :
1 : AcceptOneByTwoStrategy

Player #5 Strategy #42275
Attaque :
1 : AttackStrongestStrategy
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingEveryoneStrategy
Acceptation de soutien :
1 : AcceptOneByTwoStrategy

Player #6 Strategy #42276
Attaque :
1 : AttackAttackedPlayersStrategy2 mémoire = 10 NEGATIVE
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingStrongestThanMeStrategy NEGATIVE
Acceptation de soutien :
1 : AcceptAttackedPlayersStrongestThanMeStrategy NEGATIVE

Player #7 Strategy #42277
Attaque :
1 : AttackAttackedPlayersStrategy mémoire = 10 NEGATIVE
2 : AttackRandomlyStrategy
Demande de soutien :

1 : AskingPlayersDontAttackMeStrategy2 mémoire = 0
Acceptation de soutien :
1 : AcceptOneByTwoStrategy

Player #8 Strategy #42278
Attaque :
1 : AttackAttackedPlayersStrategy mémoire = 0
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersDontAttackMeStrategy mémoire = 0
Acceptation de soutien :
1 : AcceptAskingPlayersStrongestThanMe

Player #9 Strategy #42279
Attaque :
1 : SpitefulAttackStrategy2 mémoire = 10
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingNoBodyStrategy NEGATIVE
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy mémoire=10 NEGATIVE

Player #10 Strategy #42280
Attaque :
1 : AttackAttackedPlayersStrategy mémoire = 0 NEGATIVE
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersDontAttackMeStrategy2 mémoire = 1
Acceptation de soutien :
1 : AcceptPlayersNeverRefuseHelpMeStrategy mémoire=10 NEGATIVE

Player #11 Strategy #42281
Attaque :
1 : AttackAttackedPlayersStrategy2 mémoire = 0 NEGATIVE
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersNeverRefuseToHelpMeStrategy mémoire = 0
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy2 mémoire = 0 NEGATIVE

Player #12 Strategy #42282

Attaque :
1 : AttackTheMostSuccessfulyAttackedPlayerStrategy mémoire = 0 NEGATIVE
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersDontAttackMeStrategy2 mémoire = 10 NEGATIVE
Acceptation de soutien :
1 : AcceptPlayersNeverRefuseHelpMeStrategy mémoire = 0 NEGATIVE

Player #13 Strategy #42283
Attaque :
1 : AttackStrongestStrategy
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersDontAttackMeStrategy2 mémoire = 0 NEGATIVE
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy2 mémoire = 0

Player #14 Strategy #42284
Attaque :
1 : AttackTheMostSuccessfulyAttackedPlayerStrategy mémoire = 10
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersNeverRefuseToHelpMeStrategy mémoire = 10
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy2 mémoire = 1

Player #15 Strategy #42285
Attaque :
1 : AttackWeakestStrategy NEGATIVE
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersDontAttackMeStrategy2 mémoire = 1
Acceptation de soutien :
1 : AlwaysAcceptToSupportStrategy

Player #16 Strategy #42286
Attaque :
1 : AttackAttackedPlayersStrategy mémoire = 10
2 : AttackRandomlyStrategy
Demande de soutien :

1 : AskingPlayersDontAttackMeStrategy mémoire = 10
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy2 mémoire = 10 NEGATIVE

Player #17 Strategy #42287
Attaque :
1 : SpitefulAttackStrategy2 mémoire = 10 NEGATIVE
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingWeakestThanMeStrategy
Acceptation de soutien :
1 : AcceptAskingPlayersStrongestThanMe

Player #18 Strategy #42288
Attack :
1 : AttackTheMostSuccessfulyAttackedPlayerStrategy mémoire = 1 NEGATIVE
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersNeverRefuseToHelpMeStrategy mémoire = 10 NEGATIVE
Acceptation de soutien :
1 : AcceptAttackedPlayersStrongestThanMeStrategy

Player #19 Strategy #42289
Attaque :
1 : AttackAttackedPlayersStrategy2 mémoire = 10
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersDontAttackMeStrategy mémoire = 1
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy2 mémoire = 10

Player #20 Strategy #42290
Attaque :
1 : AttackAttackedPlayersStrategy mémoire = 0 NEGATIVE
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingEveryoneStrategy NEGATIVE
Acceptation de soutien :

1 : AcceptPlayersDontAttackMeStrategy mémoire = 1 NEGATIVE

Player #21 Strategy #42291
Attaque :
1 : SpitefulAttackStrategy2 mémoire = 0
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersDontAttackMeStrategy mémoire = 0
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy2 mémoire = 10 NEGATIVE

Player #22 Strategy #42292
Attaque :
1 : AttackAttackedPlayersStrategy2 mémoire = 0
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingStrongestThanMeStrategy
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy2 mémoire = 10

Player #23 Strategy #42293
Attaque :
1 : SpitefulAttackStrategy2 mémoire = 10 NEGATIVE
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingNoBodyStrategy
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy2 mémoire = 1 NEGATIVE

Player #24 Strategy #42294
Attaque :
1 : AttackStrongestThanMeStrategy
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersDontAttackMeStrategy2 mémoire = 0
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy2 mémoire = 1 NEGATIVE

Player #25 Strategy #42295
Attaque :
1 : AttackTheMostAttackedPlayerStrategy mémoire = 10 NEGATIVE

2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersDontAttackMeStrategy mémoire = 10 NEGATIVE
Acceptation de soutien :
1 : AcceptPlayersDontAttackMeStrategy2 mémoire = 0

Player #26 Strategy #42296
Attaque :
1 : AttackAttackedPlayersStrategy mémoire = 0
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersDontAttackMeStrategy mémoire = 10
Acceptation de soutien :
1 : AcceptAttackedPlayersWeakestThanMeStrategy mémoire = 0

Player #27 Strategy #42297
Attaque :
1 : AttackAttackedPlayersStrategy2 mémoire = 10 NEGATIVE
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersDontAttackMeStrategy mémoire = 0
Acceptation de soutien :
1 : AcceptPlayersNeverRefuseHelpMeStrategy mémoire = 1 NEGATIVE

Player #28 Strategy #42298
Attaque :
1 : AttackAttackedPlayersStrategy mémoire = 0
2 : AttackRandomlyStrategy
Demande de soutien :
1 : RandomAskingStrategy
Acceptation de soutien :
1 : AcceptPlayersNeverRefuseHelpMeStrategy mémoire = 1

Player #29 Strategy #42299
Attaque :
1 : AttackTheMostAttackedPlayerStrategy mémoire = 0
2 : AttackRandomlyStrategy
Demande de soutien :
1 : AskingPlayersNeverRefuseToHelpMeStrategy mémoire = 0
Acceptation de soutien :
1 : AcceptAskingPlayersWeakestThanMeStrategy NEGATIVE

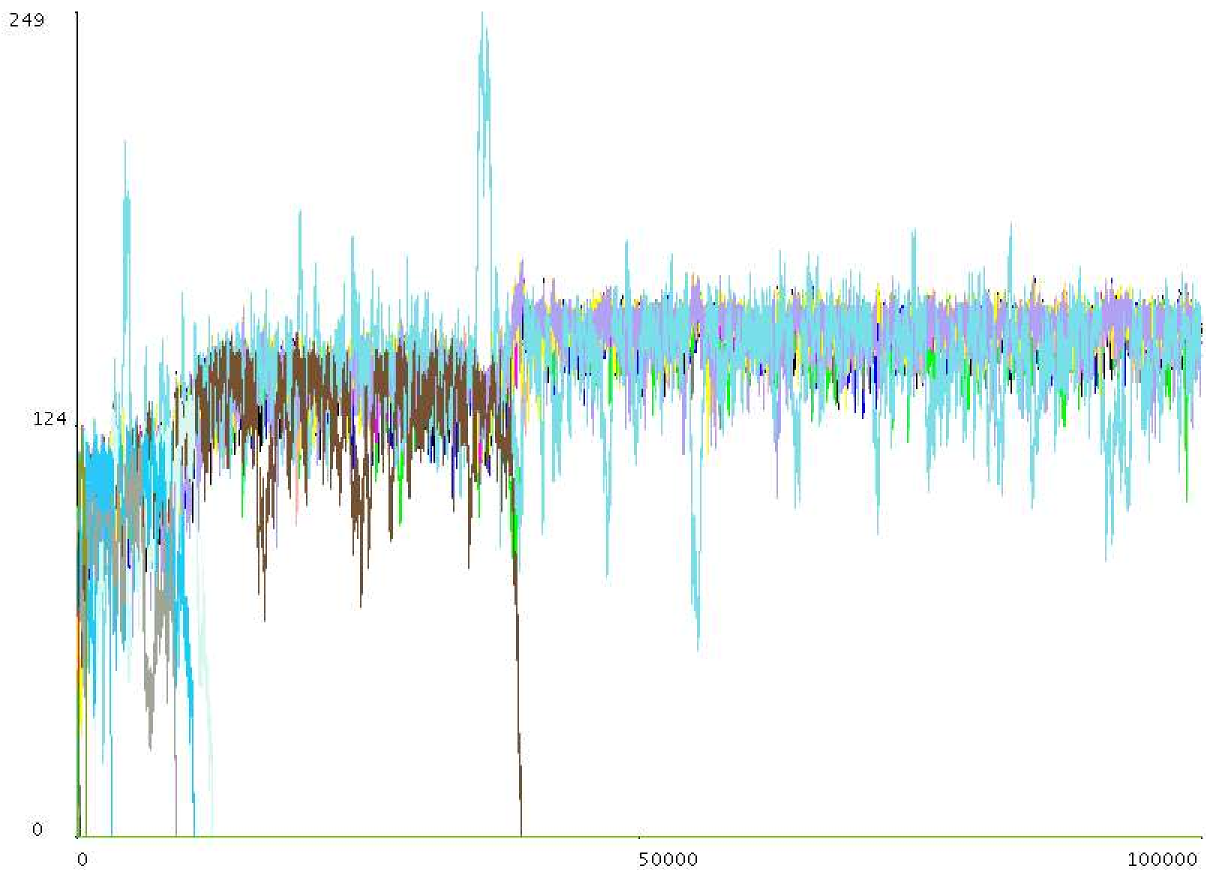

FIG. 7.2 – Un jeu infini ?

Rank	Player	Strategy	Last turn
#1	Player #10	Strategy #2950	2325
#2	Player #16	Strategy #2956	2325
#3	Player #26	Strategy #2966	1935
#4	Player #11	Strategy #2951	1483
#5	Player #14	Strategy #2954	1278
#6	Player #21	Strategy #2961	1042
#7	Player #4	Strategy #2944	907
#8	Player #1	Strategy #2941	667
#9	Player #8	Strategy #2948	625
#10	Player #6	Strategy #2946	486
#11	Player #9	Strategy #2949	457
#12	Player #29	Strategy #2969	438
#13	Player #19	Strategy #2959	346
#14	Player #2	Strategy #2942	329
#15	Player #17	Strategy #2957	264
#16	Player #24	Strategy #2964	241
#17	Player #15	Strategy #2955	201
#18	Player #18	Strategy #2958	172
#19	Player #23	Strategy #2963	159
#20	Player #27	Strategy #2967	130
#21	Player #5	Strategy #2945	107
#22	Player #3	Strategy #2943	92
#23	Player #22	Strategy #2962	82
#24	Player #25	Strategy #2965	74
#25	Player #7	Strategy #2947	64
#26	Player #20	Strategy #2960	53
#27	Player #28	Strategy #2968	50
#28	Player #0	Strategy #2940	43
#29	Player #13	Strategy #2953	24
#30	Player #12	Strategy #2952	14

TAB. 7.3 – Classement des joueurs et des stratégies

Rank	Coalition	Nb occ	Nb seq occ	Lg max
1	[Player #16, Player #10]	269	160	9
2	[Player #14, Player #11]	210	115	9
3	[Player #10, Player #26]	187	133	5
4	[Player #26, Player #16]	112	93	3
5	[Player #4, Player #11]	92	73	3
6	[Player #26, Player #11]	91	84	4
7	[Player #11, Player #4, Player #14]	81	67	3
8	[Player #16, Player #11]	81	70	3
9	[Player #4, Player #14]	73	47	6
10	[Player #21, Player #14]	73	61	3
11	[Player #10, Player #16, Player #26]	70	68	2
12	[Player #8, Player #11, Player #14]	66	40	6
13	[Player #10, Player #14]	56	50	3
14	[Player #21, Player #11]	53	43	5
15	[Player #26, Player #11, Player #16]	53	46	3
16	[Player #26, Player #11, Player #14]	50	45	2
17	[Player #10, Player #14, Player #26]	47	43	2
18	[Player #10, Player #11]	39	38	2
19	[Player #8, Player #11]	38	35	2
20	[Player #1, Player #11, Player #14]	37	30	5
21	[Player #6, Player #11, Player #14]	33	24	4
22	[Player #10, Player #11, Player #26]	32	30	2
23	[Player #16, Player #14, Player #26]	30	28	2
24	[Player #21, Player #11, Player #14]	30	28	2
25	[Player #6, Player #11]	29	23	3
26	[Player #10, Player #11, Player #14]	29	28	2
27	[Player #16, Player #21, Player #26]	28	24	2
28	[Player #1, Player #14]	27	24	2
29	[Player #16, Player #11, Player #14]	25	24	2
30	[Player #29, Player #11, Player #14]	24	18	5

Tab. 7.4 – Classement des coalitions

Bibliographie

[1] Nicolas-Gabriel Andjiga and Frédéric Chantreuil et Dominique Lepelley. La mesure du pouvoir de vote, 2003.

[2] Suryapratim Banerjee, Hideo Konoshi, and Tayfun S'onmez. Core in a simple coalition formation game, 1999.

[3] Bruno Beaufils. *Modèles et simulations informatiques des problèmes de coopération entre agents*. PhD thesis, LIFL, 2000.

[4] Bruno Beaufils, Jean-Paul Delahaye, and Philippe Mathieu. Complete classes of strategies for the classical iterated prisonner's dilemma, 1996.

[5] J. M. Bilbao, J. R. Fernandez, and J. J. Lopez. Complexity in cooperative game theory, 1999.

[6] Jean Pierre Briot. Les systèmes multi-agents, 2000-2003. http ://www-poleia.lip6.fr/˜briot/cours/.

[7] Joseph George Caldwell. Conflict, negociation and general-sum game theory, 2001.

[8] Fréderic Colard. Histoire de la théorie des jeux. http ://perso.wanadoo.fr/frederic.colard/theojeux/theojeux.html.

[9] Bruno Beaufils et Jean-Paul Delahaye et Philippe Mathieu. Le dilemme itéré des prisonniers. http ://www.lifl.fr/IPD.

[10] Jean Paul Delahaye et Philippe Mathieu. Experiences su le dilemme itéré des prisonniers, 1996. Rapport interne IT-233.

[11] D. Fiaschi et P.M. Pacini. Formation of coalitional structures and genetic algorithms, 1999.

[12] Thomas Ferguson. Game theory - part iv : Games in coalitional form, 2000.

[13] Olivier Fourdrinoy. Création d'une librairie de balises pour l'insertion d'agents jade pour des pages jsp, 2002. Mémoire de DEA - http ://www.cril.univ-artois.fr/˜leberre/TER/terjade.pdf.

[14] Gael Giraud. *La théorie des jeux*. Flammarion, 2000.

[15] Olivier Gossner. Jeux coopératifs et valeur de shapley. http ://www.thema.u-paris10.fr/gossner/ENPC03-04/cooperatifs.pdf.

[16] Olivier Gossner. The value of information in zero-sum games, 2001.

[17] Joseph Greenberg. *Handbook of Game Theory*, chapter 37 - Coalitional structures. Elsevier Science, 1994.

[18] Bernard Guerrien. *La théorie des jeux (3ème édition)*. Economica, 2002.

[19] Sébastien Konieczny. Introduction à la théorie des jeux. http ://www.irit.fr/recherches/RPDMP/persos/Konieczny/enseignement/TheorieDesJeux.pdf.

[20] Sébastien Konieczny. Le dilemme itéré des prisonniers. http ://www.irit.fr/recherches/RPDMP/persos/Konieczny/enseignement/DIP.pdf.

[21] David M. Kreps. *Théorie des jeux et modélisation économique*. Dunod, 1990.

[22] Ana Mauleon and Vincent Vannetelbosch. Coalitional negociation, 1999.

[23] Thierry Pénard. Théorie des jeux répétés, 2003. http ://perso.univ-rennes1.fr/thierry.penard/biblio/dea_jeux.htm.

[24] Howard Raiffa R. Duncan Luce. *Games and decisions*. Dover, 1957.

[25] Anatol Rapoport and Melvin Guyer. A taxonomy of 2 x 2 games, 1966.

[26] Jean-Philippe Rennard. *Vie artificielle*. Vuibert Informatique, 2002.

[27] Paul Walker. An outline of the history of game theory. http ://william-king.www.drexel.edu/top/class/histf.html.

[28] Robert J. Weber. *Handbook of game theory*, chapter 36 - Games in coalitional form. Elsevier Sciences, 1994.

[29] Murat Yildizoglu. *La théorie des jeux*, chapter 3 - Les jeux non-coopératifs avec information incomplète. Dunod, 2003. http ://beagle.montesquieu.u-bordeaux.fr/yildi/livretj/chapitre3.pdf.

Index