Apprentissage Multi-Agent (AMA)

Bruno Bouzy
Université Paris 5, UFR de mathématiques et d'informatique, C.R.I.P.5,
45, rue des Saints-Pères 75270 Paris Cedex 06 France,
tél: (33) (0)1 44 55 35 58, fax: (33) (0)1 44 55 35 35,
email: bouzy@math-info.univ-paris5.fr,
http: www.math-info.univ-paris5.fr/∼bouzy/

October 5, 2007

Abstract

Ce document répertorie les articles importants sur l'apprentissage multi-agent. Il résume les apports et effectue un état de l'art. Il classe les articles suivant la théorie des jeux, l'apprentissage par renforcement et l'apprentissage bayésien.

1 Introduction

Initialement, ce document résumait ma compréhension des articles sur l'association de "Reinforcement Learning" (RL) et de "Multi-Agent System" (MAS) pour donner le "Apprentissage par Renforcement Mult-Agent" (ARMA) ou "Multi-Agent Reinforcement Learning" (MARL). Maintenant, réalisant que l'apprentissage par renforcement n'est pas la seule forme d'apprentissage dans les systèmes multi-agent, ce document traite de "Multi-Agent Learning" (MAL) ou "Apprentissage Multi-Agent" (AMA) et dépasse le cadre de RL. Cependant la connotation RL reste forte dans ce document car le RL est le point de départ de mon intérêt pour le MAL.

D'une part, dans les "Processus Décisionnel de Markov" (PDM), le RL suppose la présence d'un unique agent apprenant [67]. L'agent traverse de nombreux états. Dans chaque état, l'agent choisit une action qui influe sur l'état suivant et le retour immédiat de l'agent. A long terme, l'agent cherche à maximiser le cumul de ses retours. D'autre part, dans les jeux matriciels, il y a un ou plusieurs joueurs mais un seul état. A chaque unité de temps d'un jeu matriciel répété, le joueur choisit une action; la récompense de chaque agent est spécifiée par une matrice associée. Un jeu matriciel avec deux joueurs dont une

matrice est l'opposée de l'autre matrice est un jeu à somme nulle. Un jeu matriciel avec deux joueurs dont les matrices sont identiques est un jeu d'équipe ou jeu de coordination. Quand les joueurs ont deux matrices indépendantes, c'est un "general-sum game". A long terme, chaque joueur du jeu matriciel répété maximise le cumul de ses retours. L'AMA est un domaine issu de Machine Learning (ML), surtout RL, et de MAS dans lequel plusieurs joueurs jouent à des jeux matriciels répétés et maximisent leur retours propres. Lorsque le jeu matriciel auquel on joue à un instant t dépend de l'état de l'agent, il s'agit d'un "jeu de Markov" ou "jeu stochastique" [59]. Un jeu stochastique est un PDM à plusieurs agents et plusieurs états dans lesquels les agents jouent à des jeux matriciels pour déterminer les retours. En pratique, un jeu matriciel répété peut être modélisé comme un jeu stochastique en définissant l'état de l'agent comme étant un nuple des actions passées de tous les joueurs depuis un temps donné.

L'état de l'art est structuré pour identifier les articles importants en AMA. Les articles fondateurs de la théorie des jeux n'utilisant pas spécialement d'apprentissage sont mis dans la section 2. Ensuite arrive la section 3 traitant de l'AMA proprement dit, c'est la section importante. Puis arrive la section 4 contenant des articles de RL mono-agent référencés par l'AMA. Et enfin arrive la section 5 sur l'apprentissage bayésien qui m'intéresse moins. Elle est néanmoins non négligeable car les hypothèses sur l'information disponible pour apprendre varient d'un article à l'autre: un apprenant ne connait-il que ses propres retours ? Les retours de tous les agents ? les stratégies des autres agents ? Y a-t-il communication d'information inter-agents ?

2 Theorie des jeux

La théorie des jeux est dominée par les travaux de John Nash. Il y a 4 articles fondateurs de John Nash [49, 50, 51, 52], l'article fondateur de Shapley sur les jeux stochastiques [59] et un article intéressant de Julia Robinson [57]. Tous ces articles constituent le point de départ de l'état de l'art du point de vue de la théorie des jeux. Ensuite, il y a le livre de Fudenberg et Levine et des articles sur les procédures "no-regret", intéressantes elles aussi.

2.1 The bargaining problem (Nash 1950)

En 1950, John Nash publie "The bargaining problem" [49]. C'est une (tentative un peu maladroite selon moi de) formalisation d'un problème concret: la négociation. L'article est parfois peu clair et inégal. Des points sont mis en avant et d'autres absents. L'article contient des petites erreurs, sur les figures notamment. L'article commence par un "no action taken by one of the individuals without the consent of the other can affect the well-being of the other one". Bon, d'accord, mais la suite de l'article ne parle pas de cette contrainte initiale. L'article oscille entre description intuitive du problème de la négociation et description formelle. On y parle d'"anticipation" à une personne ou deux personnes, de "contingencies" et de probabilités, puis de fonction d'utilité. Des

hypothèses sont formulées. On suppose que la fonction d'utilité globale des deux négociateurs est nulle si il n'y a pas de coopération entre eux. On dessine un plan avec les fonctions d'utilité $u1$ et $u2$ comme dimensions. On cherche les solutions dans un ensemble compact et convexe. A un moment on lit que $u1 = u2$, que l'on va maximiser le produit $u1u2$. Après, figure un exemple avec Jack et Bill qui ont des fournitures dont l'utilité dépend de Bill ou Jack. Jack et Bill n'ont pas d'argent à échanger. Donc Bill et Jack se répartissent les fournitures pour maximiser le produit de leurs utilités, ce qui implique effectivement qu'ils coopèrent globalement. La solution se trouve là où $u1u2$ est maximal, why not. L'optimisation du produit des utilités devrait être une hypothèse de départ, mais pas une déduction (décrite comme étant logique) à partir d'une situation cognitive de négociation. Finalement, l'article laisse un impression bizarre parce que alternant entre intuition et formalisation.

2.2 Equilibrium points in N-person games (Nash 1950)

En 1950, John Nash publie "equlilibrium points in N-person games" [50]. Cet article laisse à nouveau un impression très bizarre. [50] est comme une pièce juridique publiée pour que John Nash soit la personne auteur du théorème disant qu'il existe au moins un equilibre (dit de Nash) dans un jeu à plusieurs joueurs. L'article est textuel sans aucun formalisme mathématique et fait à peine une page de long. Incroyable. Cet article est cité par beaucoup d'articles de l'ARMA.

2.3 Méthode itérative pour minmax (Robinson 1951)

En 1951, Julia Robinson publie une méthode itérative pour résoudre un jeu matriciel a somme nulle [57]. Au contraire de [49] et de [50], [57] est bien formalisé, clair et démontré. Il donne une méthode pour trouver la solution d'un jeu à somme nulle. C'est une suite de couples de vecteurs $(U(t), V(t))$, appelé système de vecteurs, vérifiant $minU(0) = maxV(0)$ et tel que $U(t+1) = U(t) + Ai.$ et $V(t+1) = V(t) + A.j$ avec i et j tels que $vi(t) = maxV(t)$ et $uj(t) = minU(t)$. Le théorème dit que $minU(t)/t$ et $maxV(t)/t$ tendent vers v, la valeur minimax du jeu. J'ai implémenté cette méthode, la valeur minimax est approchée. De plus, l'algorithme est constructif (deux joueurs jouent un jeu répété), donc les probabilités empiriques sont calculées. Dans les années 90, cette méthode a inspiré la conception des algorithmes "sans regret" [36].

2.4 Non-cooperative games (Nash 1951)

En 1951, John Nash publie "Non-cooperative games" [51]. Cet article est la référence cité par presque tous le articles de l'ARMA. En effet, "non-cooperative" signifie sans communication autre que celle passant par les actions effectuées par les joueurs, ce qui est l'hypothèse de départ de l'ARMA. L'article est formel: définition de solution tout court, solution forte, sous-solution, points d'équilibre, jeu résolu ou pas. Exemples avec bouclage (table 1), équilibre de Nash dominé

(dilemme du prisonnier) (table 1), jeu insoluble avec plusieurs points d'équilibres (table 2), solutions neutres où tout marche (table 3) équilibres instables (table 3).

	α	β		α	β
a	5 , -3	-4 , 4	a	1 , 1	-10 , 10
b	-5 , 5	3 , -4	b	10 , -10	-1 , -1

Table 1: Exemples 1 et 2 de [51]. Le jeu 1 avec des stratégies pures est un bouclage de période 4. Il y a une stratégie mixte, équilibre de Nash $(9/16, 7/17)$. Pour le jeu 2, il y a une stratégie pure, équilibre de Nash pur $(1, 1)$. Le jeu est dit "fortement résolu". Le jeu 2 est une forme de dilemme du prisonnier.

	α	β		α	β
a	1 , 1	-10 , -10	a	1 , 2	-1 , -4
b	-10 , -10	1 , 1	b	-4 , -1	2 , 1

Table 2: Exemples 3 et 5 de [51]. Le jeu 3 est dit insoluble. Il y a 3 points équilibres: $(0, 0)$, $(1, 1)$, $(1/2, 1/2)$. Le jeu 5 est insoluble. Il y a 3 points équilibres: $(0, 0)$, $(1, 1)$, $(1/4, 3/8)$.

	α	β		α	β
a	1 , 1	0 , 1	a	1 , 1	0 , 0
b	1 , 0	0 , 0	b	0 , 0	0 , 0

Table 3: Exemples 4 et 6 de [51]. Jeu 4: solution forte: toutes les stratégies mixtes marchent. Jeu 6: il y a 2 points équilibres: $(0, 0)$ normal et $(1, 1)$ instable.

Après ces exemples, [51] traite le cas du poker simplifié à trois personnes.

En 1953, John Nash publie "Two-person cooperative games" [52]. Cet article est une extension de "The bargaining problem" [49]. Il est composé de 2 parties. La première traite de représentation et définition d'un jeu "coopératif" avec une première forme de résolution du jeu similaire à [49]. La seconde partie est une approche "axiomatique" confirmant les résultats de la première partie. Attention, "coopératif" signifie "avec communication". Dans le jeu coopératif considéré (celui de la négociation), il y a 2 étapes: d'abord une phase de choix de "menace" par chaque joueur, avec information de l'autre joueur sur la menace choisie. La menace choisie sera exécutée dans la phase suivante si le joueur n'obtient pas sa demande. Ensuite, une phase non coopérative de type somme nulle dans laquelle les joueurs effectuent une "demande" et dans laquelle les règles du jeu donnent les retours en fonction des demandes. Ensuite, l'article développe la résolution du jeu du type de celle existant dans [49]. Puis, la seconde partie donne des "axiomes" et une "démonstration" confirmant la résolution précédente.

2.5 Equilibres corrélés (Aumann 1974)

Un équilibre corrélé est un équilibre atteint dans un jeu matriciel modifié. Dans le jeu modifié, un arbitre tire une action jointe au hasard suivant une distribution de probabilités, ensuite il informe chaque joueur de son action et lui demande s'il va changer son action. Si aucun des joueurs ne change son action, alors la distribution de probabilités est un "équilibre corrélé". Le concept d'équilibre corrélé est plus général que celui d'équilibre de Nash: tout équilibre de Nash est un équilibre corrélé. Ce concept a été inventé par Aumann en 1974 [2]. Ce que je ne comprends pas très bien dans la procédure en deux temps permettant de définir un équilibre corrélé est que le jeu n'est plus vraiment le même. Il y a une sorte de communication ou partage d'information entre les deux joueurs: tous les joueurs connaissent - la distribution de probabilités de l'équilibre corrélé et - l'action jointe tirée au hasard ? Cette connaissance est-elle une information partagée ? La notion d'équilibre corrélé sort-elle du cadre des jeux non-coopératifs? En fait, la procédure en deux temps n'existe que dans la définition d'un équilibre corrélé, et pas dans le jeu réellement joué dans lequel les joueurs peuvent choisir n'importe quelle action. Donc tout va bien, on ne sort pas du cadre des jeux non-coopératifs.

2.6 Theory of learning in games (Fudenberg & Levine1998)

En 1998, Drew Fudenberg et David Levine publient le livre "The Theory of Learning in Games" [29]. Le début est intéressant, notamment l'idée du teaching, symétrique de celle du learning. Laisser passer du temps.

2.7 Algorithmes "sans regret" (Hart & Mas-Colell 2000)

En 2000, Sergiu Hart et Andreu Mas-Colell publient une méthode simple, basée sur le regret, et convergente vers les équilibres corrélés [36]. Le regret d'une action B par rapport à la dernière action jouée A est la différence entre, d'une part le cumul des retours obtenus si, depuis le début du jeu répété, on avait joué l'action B au lieu de jouer l'action A chaque fois que celle-ci a été jouée, et d'autre part le cumul effectif des retours. (Le regret de la dernière action est donc nul). A l'instant suivant, la probabilité de jouer une action est une fonction linéaire de son regret. Avec une telle méthode, Hart et Mas-Colell montrent que le regret tend vers zéro au bout d'un temps infini. Dans le domaine MAL issu de l'IA, cette approche est en vogue depuis 2004. En 1997, Dean Foster et Rakesh Vohra ont présenté une méthode utilisant l'apprentissage "calibré" et convergente vers les équilibres corrélés [27]. En 1999, ils ont présenté une approche aussi basée sur le regret [28], mais moins simple que [36].

3 Apprentissage multi-agent

Cette partie cite des références sur l'apprentissage multi-agent en les commentant.

3.1 Agents indépendants ou coopérants (Tan 1993)

En 1993, Ming Tan a étudié expérimentalement des agents agissant "indépendamment" ou "coopérativement" [68]. Cet article est précurseur: le formalisme des PDM (qui est très utilisé dans l'état de l'art ARMA) est absent de ce papier, mais les expériences menées sont néanmoins très intéressantes. Le support des expériences est le jeu des proies et des prédateurs. Les prédateurs peuvent partager ou pas des perceptions, partager ou pas des politiques de capture et effectuer ou pas des tâches complémentaires. Les perceptions d'un agent sont les cases de la fenêtre 5x5 centrée sur l'agent. Dans l'étude de cas numéro 1, il y a une proie aléatoire, un chasseur apprenant et un observateur aléatoire ("scouting agent"). Le chasseur utilise les perceptions de l'observateur. Ensuite, il y a une proie aléatoire et deux chasseurs apprenant utilisant leurs propres perceptions et les perceptions de l'autre chasseur ("mutual scouting"). Dans cette étude, les expériences montrent que le scouting est bénéfique et que le scouting mutuel aussi. Dans l'étude de cas numéro 2, les agents apprenants partagent des séquences de perceptions d'actions avec les retours associés. Lorsque la séquence d'un chasseur aboutit à une capture, il envoie la séquence à l'autre chasseur qui rejoue cette séquence "mentalement" pour mettre à jour sa politique. Ensuite, un chasseur expert chasse avec un chasseur novice en utilisant cette technique pour démontrer l'efficacité du principe. Dans l'étude de cas numéro 3, la tâche est complémentaire: pour capturer une proie, les deux chasseurs doivent être simultanément sur la case de la proie ou sur une case adjacente. Dans cette tâche plus difficile, les chasseurs apprenants indépendamment arrivent à réussir mais sont moins efficaces que les chasseurs apprenants coopératifs en partageant leurs perceptions. Vocabulaire: dans [68], "coopératif" signifie "communicant" ou "partageur". Dans la littérature qui suit, les agents ne sont généralement pas coopératifs en ce sens (communicants), ils jouent à des jeux sans communiquer et dans lesquels la coopération est nécessaire et survient éventuellement après apprentissage.

3.2 Minimax-Q (Littman 1994)

En 1994, Michael Littman a proposé d'utiliser les jeux de Markov comme cadre d'étude de l'apprentissage multi-agent [41]. Il part de Q learning [71] et propose Minimax-Q. La différence avec Q learning réside dans l'utilisation d'un opérateur minimax dans la formule de mise à jour à la place de l'opérateur max. Minimax Q s'applique au jeux à somme nulle. L'idée d'utiliser minimax vient du contexte sur l'équilibre de Nash [51] dans la théorie des jeux. L'exemple de test est le football sur une grille 5x5 avec 2 joueurs et une balle.

3.3 1995-1996

En 1995, Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund et Robert Shapire [1] présentent "Hedge", un algorithme utilisé pour résoudre le Multi-Armed Bandit Problem. "Hedge" est cité plusieurs faois par les papiers sur les algorithmes

experts utilisés plus tard. En 1996, David Carmel et Shaul Markovitch publie [14]. Ils présentent un modèle d'apprentissage d'automate à états.

3.4 Nash-Q (Hu & Wellman 1998)

En 1998, Junling Hu et Michael Wellman définissent un cadre théorique et un algorithme d'apprentissage par renforcement appelé Nash-Q [37]. Nash-Q est supposé s'appliquer aux jeux à somme quelconque. L'exemple de test est le jeu de la grille 3x3 avec deux agents, deux barrières et une case objectif. Mais en 2000, Michael Bowling identifie des problèmes de convergence de l'algorithme Nash-Q [8].

3.5 IL et JAL (Claus & Boutilier 1998)

En 1998, Caroline Claus at Craig Boutilier effectuent une expérience avec des agents apprenants indépendents (Independent Learner = IL) ou utilisant les actions jointes (Joint Action Learner = JAL) [18]. L'article est limité aux jeux matriciels "coopératifs". Dans [18], le terme coopératif est pris au sens où les agents se coordonnent par leurs actions et pas au sens de Nash [49, 52] où les agents communiquent. Un IL met à jour des Q values pour chaque action. Un JAL met à jour des Q values pour chaque action jointe. Ils étudient le "penalty" game de la table 4.

	a0	a1	a2
b0	10	0	k
b1	0	2	0
b2	k	0	10

Table 4: Le "penalty" game . k est négatif.

Les auteurs disent que pour des valeurs suffisamment négatives de k, l'équilibre atteint est (a1, b1). Ces résultats me surprennent beaucoup pour les JAL. Un JAL va trouver que la Q value de (a0, b0) est 10. Après, pourquoi choisir l'action qui maximise les Q values des actions propres? Pourquoi ne pas choisir l'action dont une des actions jointes est maximale? L'exploitation est très mauvaise selon moi. Par ailleurs, selon moi, les ILs pourraient converger vers la bonne action jointe à condition d'utiliser une meilleure stratégie d'exploration. La stratégie d'exploration utilisée dans [18] est celle de Boltzmann, qui, selon moi, est inadéquate. Avec une exploration de type UCB, je pense que les équilibres convenables peuvent être atteints par des ILs. Ensuite, les auteurs étudient le "climbing" game de la table 5.

Au climbing game, les ILs convergent dans un premier temps vers (a2, b2) avec la valeur 5. Puis ayant convergés, le joueur ligne converge vers b1 et la valeur devient 6. Puis, ayant à nouveau convergés, le joueur colonne converge vers a1 et la valeur devient 7. Après, les joueurs restent bloqués sur (a1, b1)

	a0	a1	a2
b0	11	-30	0
b1	-30	7	6
b2	0	0	5

Table 5: Le "climbing" game .

équilibre sous-optimal alors que l'optimum est (a0, b0) qui est aussi un équlibre de Nash. On voit qu'avec une mauvaise stratégie d'exploration, les ILs convergent vers une valeur sous-optimale. Je ne comprends pas pourquoi les JALs ne convergent pas vers (a0, b0). A nouveau, pourquoi un JAL ne choisit-il pas l'action dont une des actions jointes est maximale? Dans la fin de l'article, les auteurs tentent de biaiser l'exploration de manière compliquée est ratée. Cet article maladroit est très souvent cité (?) et les deux types d'agents apprenants, IL et JAL, très connus.

3.6 Roshambo 2000

En 2000, une compétition de Pierre-Papier-Ciseaux, Roshambo, a été organisée par Darse Billings [6]. Les compétiteurs se sont concentrés sur les modèles de l'adversaire et les modèles de modèles, etc. et pas sur les algorithmes d'apprentissage multiagent (!). Iocaine Powder de Dan Egnor et utilisant des niveaux méta a gagné [26].

3.7 IGA (Singh, Kearns & Mansour 2000)

En 2000, Satinger Singh, Michael Kearns et Yishay Mansour étudient la convergence, ou non convergence, de l'ascension de gradient infinitésimale (IGA = Infinitesimal Gradient Ascent) dans les jeux 2x2 à somme quelconque [62]. Ils précisent d'abord le résultat connu selon lequel IGA peut ne pas converger. Ensuite, ils appellent α et β les poids des actions des stratégies mixtes des deux joueurs, avec $0 <= \alpha <= 1$ et $0 <= \beta <= 1$. Ils présentent l'ascension de gradient dans les jeux répétés. Puis, ils montrent l'équation différentielle (dépendant de α et β) correspondant à l'ascension de gradient infinitésimale et interprètent la résolution de cette équation en fonction de la matrice 2x2, appelée U, gérant cette équation. La résolution dépend de plusieurs cas: U non inversible, U inversible avec des valeurs propres imaginaires, U inversible avec des valeurs propres réelles. Si U n'est pas inversible, les solutions sont pures et situées sur un coin du carré $[0, 1] \times [0, 1]$. Si U est inversible avec des valeurs propres imaginaires, les courbes de niveau sont des ellipses. Le résultat dépend de la position du centre de l'ellipse par rapport au carré $[0, 1] \times [0, 1]$. Si le centre est à l'extérieur du carré, la solution est dans un coin du carré (?). Si le centre est à l'intérieur du carré mais que les valeurs initiales sont sur une ellipse contenue complètement dans le carré, la solution est un cycle qui boucle le long de l'ellipse (équilibre de Nash mais pas de convergence, plutôt un bouclage). Si le centre

est à l'intérieur du carré mais que les valeurs initiales sont sur une ellipse non contenue complètement dans le carré, la solution est située à l'intersection de l'ellipse et de la frontière. Si le centre est sur une frontière du carré, la solution est sur la frontière du carré. Si U est inversible avec des valeurs propres réelles, les auteurs montrent que la solution est sur la frontière du carré. Finalement, les auteurs ont classifié graphiquement les cas de convergence et ont proposé un algorithme de d'asecnsion de gradient. L'étude est théorique mais intéressante car elle propose une interprétation graphique dans le carré $[0, 1] \times [0, 1]$. Ne s'applique qu'a des jeux à deux joueurs.

3.8 Rmax (Brafman & Tennenholtz 2001)

En 2001, Ronen Brafman et Moshe Tennenholtz proposent Rmax [13]. C'est une extension de E3 aux jeux de Markov avec une généralisation des deux stratégies explicites de E3 (exploration et exploitation) en une seule dans laquelle l'exploration devient implicite. Cela utilise l'idée simple et classique en apprentissage par renforcement de mettre des retours fictifs très grands (R max) sur les noeuds non explorés. L'algorithme Rmax est désormais très connu dans la communauté d'apprentissage par renforcement, mais je trouve que l'idée originale revient à [40].

3.9 Friend-or-Foe-Q (Littman 2001)

Toujours en 2001, Michael Littman a proposé Friend-or-Foe Qlearning (FFQ) [42]. Cet algorithme s'applique aux jeux de Markov à somme quelconque. Il suppose que les joueurs soient identifiés comme ami ou ennemi, et dans ces conditions, l'algorithme converge toujours. Il utilise les équilibres de Nash compétitifs et les équilibres de Nash coopératifs. L'exemple de test est le jeu de la grille 3x3 avec deux agents, deux barrières et une case objectif.

3.10 Bully et Godfather (Littman & Stone 2001)

Avec Peter Stone, au lieu de se focaliser sur les équilibres de Nash, il part d'une situation d'enchères [66] pour définir des comportements de "leader". La situation d'enchères est la suivante. Deux agents s'engagent dans un jeu d'enchères sur deux articles, A et B. L'enchère démarre à 1$ et peut monter jusqu'à 3$ pour un article. Chaque joueur évalue les articles à 4$. Si un joueur enchérit seul sur un article, il obtient l'article pour 1$, gagnant 3$ net. Si les deux joueurs enchérissent sur le même article, son prix monte à 3$ et il est attribué au hasard à l'un des jeux joueurs. Dans ce cas, un joueur gagne 1$ net et l'autre rien. En moyenne dans ce cas, un joueur gagne 0.5$. On suppose que ou bien le joueur 1 enchérit sur l'article A seul (action 1) ou bien sur les 2 (action 2), et que le joueur 2 enchérit ou bien sur l'article B seul (action 1) ou bien sur les 2 (action 2). On représente cette situation par le jeu matriciel de gauche de la table 6.

	a1	a2	a1	a2
a1	3 , 3	0.5 , 3.5	3 , 3	1.5 , 3.5
a2	3.5 , 0.5	1 , 1	3.5 , 1.5	1 , 1

Table 6: A gauche, le jeu des enchères. Si les deux joueurs enchérissent sur leur seul article, ils obtiennent chacun 3$ net. S'ils enchérissent tous les deux sur les deux articles, ils obtiennent $0.5\$ + 0.5\$ = 1\$$. Si un joueur enchérit sur son article et l'autre sur les deux, celui qui a enchérit sur son article seul, gagne 0.5$, et l'autre gagne $3\$ + 0.5\$ = 3.5\$$, d'où les valeurs du jeu matriciel. Le jeu ressemble au dilemme du prisonnier. A droite, les valeurs d'un jeu légèrement modifié. Le jeu ressemble alors au jeu "chicken".

On remarque alors que le jeu des enchères ressemble au dilemme du prisonnier. Si les valeurs de ce jeu matriciel sont modifiées comme indiqué par la matrice de droite de la table 6, alors on reconnaît le jeu "chicken". Les auteurs remarquent ensuite que les algorithmes "best-response" sont des algorithmes "suiveurs" au sens où ils s'adaptent à la stratégie de l'adversaire en trouvant une meilleure réponse à cette stratégie, mais ils n'essaient pas de modifier la stratégie de l'autre agent ou d'en tenir compte. Q learning est un exemple de suiveur best-response. Les auteurs introduisent ensuite les comportements d'algorithmes de type "leader". Un joueur peut être "leader" en choisissant ses actions de manière à influencer un suiveur potentiel. Le plus simple des leaders est "Bully" (bestial? bovin?). Bully choisit l'action maximale pour lui en supposant que l'adversaire est un suiveur best-response. Autrement dit, Bully's action = argmax sur i m1(i,ji*) avec ji* = argmax sur j m2(i,j). Un autre leader connu est "Godfather". Godfather choisit une paire d'actions cible telle que, pour chaque joueur, le retour de cette action cible est supérieure à sa valeur de sécurité. Tant que le joueur adversaire joue l'action complémentaire de Godfather, Godfather continue de jouer l'action cible. Si l'adversaire arrête de faire cette action, alors Godfather joue l'action obligeant l'adversaire à n'obtenir que sa valeur de sécurité (ou moins bien s'il joue mal). La valeur de sécurité est celle obtenue par un joueur Minimax maximisant ses retourspropres sachant que l'adversaire essaie de minimiser ces retours. Autrement dit, Minimax's action = argmax sur i m1(i, ji*) avec ji* = argmin sur j m1(i, j). Et valeur de sécurité = max sur i m1(i, ji*) avec ji* = argmin sur j m1(i, j). L'article présente les résultats de Qlearning 1, Qlearning 0, Bully et Godfather jouant les uns contre les autres. Qlearning 1 est un Qlearning dont les états tiennent compte de l'action précédente. Bully et Godfather obligent Qlearning 1 vers des paires ciblées et battent Qlearning 0 (qui ne peut apprendre en fonction de l'action précédente). L'article conclut sur l'intérêt des leaders par rapport aux suiveurs. Il prévoit que le domaine de l'ARMA progressera beaucoup en étudiant non seulement les suiveurs mais aussi les leaders, d'autant plus que les leaders utilisés dans l'article sont très simples. Le point faible est l'absence de résultat entre joueurs d'une même catégorie (suiveurs entre eux ou leaders entre

eux).

3.11 PHC et WoLF (Bowling & Veloso 2001)

Toujours en 2001, Michael Bowling et Manuela Veloso définissent les propriétés (réductrices) de "rationalité" et "convergence" des agents apprenants. Le principe de rationalité dit que contre des adversaires stationnaires, l'agent apprenant doit converger vers la "best-response" de ces adversaires. Le principe de convergence dit simplement que contre un agent stationnaire, l'agent apprenant doit converger. Dans ce cadre, ils définissent le principe WoFL (= Win or Learn Fast) et un algorithme associé [11]. Ils utilisent notamment l'idée de pas d'apprentissage distincts selon que l'on gagne (pas d'apprentissage faible) ou que l'on perd et qu'il faut apprendre vite (pas d'apprentissage grand). Pour savoir si on gagne ou perd, l'algorithme maintient deux politiques: la politique habituelle et la politique moyenne, et il compare l'espérance des Q values si on utilise la politique habituelle ou si on utilise la politique moyenne. Il montre aussi l'algorithme PHC (Policy Hill Climbing) qui améliore une politique sans la rendre déterministe. Cet article est très intéressant.

3.12 Belief et History, PHC-Exploiter (Chang & Kaelbling 2001)

Toujours en 2001, Yu-Huan Chang et Leslie Pack Kaelbling propose une classification des agents jouant aux jeux répétés suivant deux critères: (1) taille de l'historique des actions passées prises en compte par l'agent et (2) croyance sur la taille de l'historique utilisée par l'adversaires [16]. Cela donne une dimension "Historique": (H0, H1 ou H∞) et une dimension "Belief" (B0, B1 ou B∞). D'où 9 classes. MinimaxQ et NashQ sont classés H0B0. Qlearning0, WoLF-PHC sont classés B0H∞. Qlearning1 est classé B1H∞. Bully est classé B∞H0. GodFather est classé B∞H1. Et un hypothétique "multiplicative weight" est classé B∞H∞. Les auteurs montrent que la classe B∞H∞ est la classe pertinente à étudier. Ils proposent un algorithme, PHC-Exploiter, extension de PHC, intégrant la classe B0H∞. Dans cette classe, au vu des jeux de test effectués, PHC-Exploiter semble surpasser les algorithmes existants. Reste à étendre cet algorithme pour qu'il intègre la classe intéresante B∞H∞. L'article mentionne plusieurs jeux dont Hawk-Dove à gauche de la table 7.

3.13 L'année 2001

Toujours en 2001, Bikramit Banerjee, Sandip Sen et Jing Peng étudient minimax-Q [41] dans le cadre des jeux à somme nulle. Ils étudient aussi minimax-SARSA et montrent qu'il est meilleur que minimax-Q dans les jeux à somme quelconque [5]. Amy Greenwald, Amir Jafari, Gunes Ercal et David Gondek étudie les algorithmes d'apprentissage multi-agent sans regret [32]. Ils montrent que ces algorithmes obtiennent des résultats similaires à "fictitious play": ils ne convergent pas toujours mais quand ils convergent ils convergent vers une équilibre de

	a1	a2		Théatre	Football
a1	0 , 0	3 , 1	Théatre	2 , 1	0 , 0
a2	1 , 3	2 , 2	Football	0 , 0	1 , 2

Table 7: A gauche, Hawk-Dove game. A droite, le jeu de la bataille des sexes. La femme préfère aller au théatre qu'au football et l'homme l'inverse. Cependant, l'homme et la femme préfèrent avant tout être ensemble. Il y a deux équilibres purs $(0, 0)$ et $(1, 1)$ dans lesquels un joueur est avantagé, et un équilibre mixte $(1/2, 1/2)$ équitable entre les deux joueurs mais inférieurs aux équilibres purs.

Nash. Inversement, un équilibre de Nash est sans regret. Un joueur "fictitious" est un joueur qui connaît ses actions, ses retours, les retours des autres et les actions des autres à chaque étape. Un joueur "informé" connaît ses actions, ses retours et les retours des autres, et un joueur "naïf" connaît ses actions et ses retours uniquement. Les algorithmes sans regret peuvent boucler (par exemple sur le Shapley game). L'article mentionne plusieurs jeux dont la bataille des sexes à droite de la table 7.

3.14 Multiagent Factored MDP (Guestrin & al)

En lien avec les "factored" MDP [7], Carlos Guestrin a publié des articles sur l'apprentissage multi-agent coopératif. Un problème complexe est à résoudre. Il est décomposé, "factorisé", en sous-problèmes plus simples. Dans cette problématique, un problème doit être défini entre autre par un retour global qui est fonction des retours locaux définis par les sous-problèmes. Les agents apprenants se coordonnent pour résoudre le problème, la coordination globale est assurée par l'existence du retour global [34, 35, 33]. Les travaux de Guestrin ont des titres attractifs : "Multiagent planning with factored MDPs" ou "Coordinated Reinforcement learning", mais ils sortent du cadre de l'apprentissage multi-agent non coopératif, contexte de ce document. Néanmoins, parce que traitant des MDP factorisés, ils sont intéressants.

3.15 IGA-WoLF (Bowling & Veloso 2002)

En 2002, Michael Bowling et Manuela Veloso reprennent WoLF [11] et l'associe avec IGA [62] pour donner IGA-WoLF [12]. L'idée nouvelle est d'avoir des taux d'apprentissage variables dans la formule de mise à jour de IGA. Et, afin de déterminer le taux d'apprentissage dynamiquement, ils gardent l'ancienne idée de WoLF consistant à reconnaître si on "gagne" ou si on "perd". Dans la seconde partie du papier (qui est expérimentale), ils reprennent malheureusement PHC, et pas IGA-WoLF, et donnent les résultats connus dans [11]. En 2003, Michael Bowling soutient sa thèse sur l'apprentissage multi-agent en présence d'agents avec limitations [10].

3.16 Correlated Q (Greenwald & Hall 2003) –

En 2003, Amy Greenwald et Keith Hall publient "Correlated Q Learning" [31]. L'idée est de définir des mécanismes de sélection d'équilibre basés sur des fonctions d'objectifs globales. Ils donnent quatre fonctions objectif à maximiser: la somme, le minimum ou le maximum des retours des agents et une autre (que je ne comprends pas). Ils appellent les algorithmes associés "utilitaires", "égalitaires", "républicains" ou "libertaires" (!). Le point faible de cette approche est le retour en arrière vers la centralisation que l'on veut éviter. Question: quel est le lien avec les équilibres corrélés de Aumann [2] ?

3.17 AWESOME (Connitzer & Sandholm 2003) –

En 2003, Vincent Connitzer et Tuomas Sandholm publient AWESOME, un algorithme "général" qui converge en auto-jeu et apprend une meilleure réponse contre des agents stationnaires [19]. (AWESOME = Adapt When Everybody is Stationary, Otherwise Move to Equilibrium). Finalement, cet article n'est pas terrible sur le fond, puisque il ne considère que des autres agents stationnaires: quel différence avec le mono-agent ? Le fait de converger vers un équilibre de Nash en auto-jeu ? Bof. La suite de l'état de l'art montre que ce n'est pas un résultat satisfaisant, puisque il faut converger vers un équilibre de Nash non Pareto dominé. Faire une best-response en auto-jeu n'est pas satisfaisant non plus puisque le but est de coopérer avec son partenaire pour trouver une best-response meilleure que si on ne coopère pas. En plus, le code de AWESOME calcule en dur l'équilibre de Nash avec la fonction $computeNashEquilibrium()$ qui est une boîte noire.

3.18 GIGA (Zinkevich 2003)

A ICML 2003, Martin Zinkevich, présente [73] sur la programmation "convexe" en ligne (?) et l'applique à la théorie des jeux et à Infintesimal Gradient Ascent (IGA) [62]. Cet article semble bien mais il est théorique et ne contient aucun résultat expérimental.

3.19 L'algorithme S et le MASD (Stimpson & Goodrich 2003) ++

A ICML 2003, Jeffrey Stimpson et Michael Goodrich présente le "satisficing algorithm" ou "S-algorithm" [64]. D'abord, les auteurs insistent, justement selon moi, sur le fait qu'un algorithme doit converger vers une solution Pareto optimale, et non pas vers un équilibre de Nash comme la plupart de l'état de l'art l'a supposé [41, 37, 62, 12, 19]. L'équilibre de Nash doit servir de solution éventuelle de repli lorsque les autres agents sont inimicaux. L'idée de l'algorithme reprend celle du bargaining problem de Nash [49] dans laquelle un agent a un niveau d'aspiration, ou niveau de demande, et choisit une action pour que le niveau d'aspiration soit atteint. Au début le niveau d'aspiration est élevé

pour tous les agents. Quand un retour arrive, un agent met à jour son aspiration avec une formule de type RL. Si l'aspiration est supérieure au retour, l'agent choisit une action aléatoirement, sinon il garde l'action précédente. Par ailleurs, les auteurs définissent le "Multi Agent Social Dilemma" (MASD), c'est-à-dire le jeu du "dilemme social" pour N agents, ayant $M + 1$ actions chacun. Une action d'un agent i correspond à mettre une quantité de ressources ui entière pour la société globale et une quantité $M - ui$ pour lui. L'agent reçoit un retour dépendant de ui et de $u - i$. Par exemple, pour le jeu du dilemme du prisonnier habituel, on a $N = 2$ et $M = 1$. Sur le jeu du dilemme social, l'algorithme converge vers des solutions Pareto optimales en auto-jeu et vers des solutions de replis qui sont des équilibres de Nash si les autres agents sont égoistes. Les plus de cet article: (1) l'idée de dire que en auto-jeu la Pareto optimalité est pertinente et pas l'équilibre de Nash, (2) la définition d'un jeu à N joueurs et M actions qui dépasse le cadre habituel des 2 joueurs et des 2 actions, (3) les très bons résultats des expériences en auto-jeu ET contre des adversaires égoïstes; les moins: (1) les tests ne sont faits que sur le dilemme social et pas sur des jeux de nature compétitive, (2) les autres agents ne sont pas non-stationnaires, (3) l'exploration est complètement aléatoire. Avec un titre différent [65] est une copie conforme de [64].

3.20 Les dérivations de S (Crandall & Goodrich 2004) +

A AAAI 2004, Jacob Crandall et Michael Goodrich présentent une extension du S algorithm [64]. L'extension présentée se nomme "SAwT" (= S Algorithm With Trembles) [21]. L'idée de SAwT est d'insérer des "trembles" (tremblements?) de taille β avec une probabilité η dans la formule de mise à jour des aspirations. Cette technique est censée permettre de gérer la non-stationarité des autres agents, ce que ne fait pas le S algorithm. On prend β de taille légèrement supérieure au plus grand retour du jeu. Le papier ne dit pas quelle est la stratégie de décroissance de β ou η au cours du temps. Les résultats sont ceux obtenus sur le jeu de la bataille des sexes (table 7). En 2004, les auteurs ont aussi publié des dérivations de l'algorithme S: le SPAM agent (Social and Payoff Maximizing Agent) [20] et, même chose, le SaPMA [22].

3.21 MetaStrategy (Shoham & Powers 2003, 2004) ++

En 2003, Yoav Shoham et Rob Powers critiquent l'approche utilisée jusque là, trop basée sur l'équilibre de Nash. Ils proposent des "agendas" [60].

En 2004, ils proposent de nouveaux critères et un nouvel algorithme pour l'apprentissage multi-agent [54]. BR_ϵ est une stratégie best response généreuse: pourvu que l'on soit ϵ en dessous de notre best-response, on cherche la best-response pour l'autre agent. BullyMixed est une extension de Bully aux stratégies mixtes. Les nouveaux critères à respecter par les apprenants sont:

(Propriété 1) obtenir une BR_ϵ contre les "gentils",

(Propriété 2) obtenir un équilibre de Nash non Pareto-dominé en auto-jeu,

(Propriété 3) obtenir le niveau de sécurité contre les "méchants".

L'algorithme "MetaStrategy" utilise plusieurs sous-stratégies: BR_ϵ, Minimax, "Godfather", "BullyMixed" et des classes d'adversaires. MetaStrategy démarre par une phase de coodination/exploration pour déterminer la classe de l'adversaire. Après, il choisit entre 3 stratégies. Si l'adversaire est stationnaire, la stratégie est best response. Si la stratégie BullyMixed a réussi pendant la phase d'exploration, il la garde. Sinon il adopte une stratégie par défaut égale à la best response aux H derniers coups de l'historique. Si la valeur moyenne tombe en dessous de la valeur de sécurité, MetaStrategy joue la stratégie minimax. L'algorithme MetaStrategy est testé avec succès sur GAMUT.

3.22 GAMUT (Nudelman & al 2004) ++

En 2004, Michael Weinberg et Jeffrey Rosenschein étudient l'apprentissage multi-agents "Best-Response" dans des environnements non stationnaires [72]. Eugene Nudelman et ses collègues lancent une étude systématique ("run the gamut") sur les jeux utilisés comme tests en apprentissage multi-agent, et ils montrent que souvent les tests publiés ne sont pas suffisants pour démontrer la validité d'un algorithme. Ils proposent GAMUT, un générateur de jeux matriciels à mettre en entrée d'algorithmes pour les tester [53].

3.23 ReDVaLeR (Banerjee & Peng 2004) +

Bikramit Banerjee et Jing Peng publient l'algorithme "ReDVaLeR" (= Replicator Dynamics with a Variable Learning Rate) [4]. L'article repose sur la "Replicator Dynamics" de [29]. Le point positif est le fait de converger vers une best response contre des agents stationnaires (normal), d'obtenir un retour sans regret contre des agents utilisant des séquences fixes arbitraires de politiques non stationnaires, et enfin en auto-jeu converger vers l'équilibre de Nash. Point faible: les tests sont absents. Point fort: non stationnarité des autres apprenants.

3.24 Safe strategies (McCracken & Bowling 2004) +

Les auteurs constatent que les algorithmes utilisant des modèles de l'adversaires sont risqués: soit le modèle de l'adversaire est juste, et tout va bien, soit il est incorrect et tout va mal. Ils proposent de rajouter un module de garanti de valeur de sécurité mis en oeuvre lorsque le modèle de l'adversaire est faux [44]. Ils définissent les stratégies ϵ-certaines.

3.25 M-Qubed (Crandall & Goodrich 2005) ++

A ICML 2005, Jacob Crandall et Michael Goodrich présentent "M-Qubed", un algorithme d'apprentissage pour "combattre", "compromettre" et "coopérer" [23]. Argumenté par le "Folk theorem", ils définissent deux propriétés: la propriété de sécurité disant qu'un agent ne doit pas utiliser des stratégies donnant un résultat inférieur à la valeur minimax (valeur de sécurité), et la propriété

de coopération/compromis qui spécifie comment un agent peut “influencer” un autre agent et dans quelles conditions il peut “être influencé”. Cet article est très bien. Il est curieusement dans la catégorie “best response” puisque rien n'indique explicitement d'utilisation de stratégie “leader” (type Bully ou Godfather). “M-Qubed” stands for “MMM-Q” which in turn stands for “Max or MiniMax Q”. L'article est testé sur les jeux de la table 8, de la table 9, de la table 10, sur Chicken et sur Prisonner's Dilemma.

	r	p	s
r	0, 0	-1, 1	1 , -1
p	1, -1	0 , 0	-1 , 1
s	-1, 1	1 , -1	0 , 0

	a	b	c
a	0 , 0	0 , 1	1 , 0
b	1 , 0	0 , 0	0 , 1
c	0 , 1	1 , 0	0 , 0

Table 8: A gauche, Rock Paper Scissor. C'est un jeu à somme nulle pour lequel il existe un unique équilibre de Nash: $(1/3, 1/3, 1/3)$. A droite, Shapley's Game. C'est un jeu de coordination dans lesquels les joueurs doivent faire tourner équitablement celui qui gagne 1.

	a1	a2
a1	1 , -1	-1, 1
a2	-1, 1	1 , -1

	a1	a2
a1	2 , 2	0 , 0
a2	0 , 0	4 , 4

Table 9: A gauche, Matching Pennies, c'est un jeu à somme nulle avec pour équilibre de Nash $(1/2, 1/2)$. A droite, Coordination Game. C'est un jeu dans lequel les deux joueurs doivent jouer (1) la même action, et (2) celle qui est dominante.

	a1	a2
a1	2 , 2	3 , -5
a2	-5, 3	4 , 4

	a1	a2
a1	0 , 3	3 , 2
a2	1 , 0	2 , 1

Table 10: A gauche, Stag Hunt. C'est un jeu de coordination dans lequels il y a deux équilibres purs $(0, 0)$ et $(1, 1)$. (C'est bizarre, j'ai vu d'autres définitions de stag hunt?). A droite, Tricky Game. C'est un bouclage entre les 4 paires d'actions.

L'idée de compromis coopération vient du “folk theorem”. Le folk theorem dit que, dans le jeu répété, il existe une infinité d'équilibres de Nash situés dans une portion convexe de l'espace des payoffs: la partie située au dessus de la valeur de sécurité du jeu. NB: dans un jeu répété on peut tenir compte de l'historique de taille w des actions jouées par les autres joueurs. L'algorithme M-Qubed utilise des états avec historique de taille 1. L'algorithme M-Qubed utilise deux politiques: (a) une politique de “coopération compromission” (CC)

qui est greedy sur la meilleure Q value de l'état du jeu répété si elle est supérieure à la valeur de sécurité, et minimax sinon; (b) une politique de sécurité qui est greedy sur la meilleure Q value de l'état du jeu répété si la moyenne de tous les payoffs est supérieure à la valeur de sécurité, et minimax sinon. M-Qubed tient a jour la probabilité $1 - \beta$ d'utiliser (a) et β d'utiliser (b) en comparant le retour avec la moyenne sur tous les états. Par ailleurs, pour gérer l'exploration, M-Qubed randomise avec une probabilité η les noeuds non explorés si l'en reste. Les résultats présentés sont très bons sur le dilemme du prisonnier, chicken, shapley's game, staghunt, tricky game. Cet article mentionne rQl qui doit être Q learning sur jeu répété.

3.26 Manipulator (Powers & Shoham 2005) ++

Toujours en 2005, Rob Powers et Yoav Shoham publient l'algorithme "Manipulator" prévu pour jouer contre des adversaires apprenants et adaptatifs [55]. Manipulator ressemble Metastrategy [54]. La propriété 1 devient la propriété d'optimalité ciblée. La propriété 2 devient la propriété de compatibilité. La propriété 2 devient la propriété de sécurité. BullyMixed est remplacé par une version généreuse de "Godfather stochastique". Une stratégie "MemBR" est définie comme étant la best response contre une "stratégie conditionnelle". "Godfather stochastique" et la "stratégie conditionnelle" ne sont pas clairs dans l'article. A relire. Comme Metastrategy, Manipulator démarre par une phase de coodination/exploration pour déterminer la classe de l'adversaire. Après, il choisit entre 3 stratégies. Les idées sont à nouveau validées par un test utilisant GAMUT dans lequel Manipulator est excellent. L'article mentionne les deux jeux de la table 11.

	Cooperate	Defect
Cooperate	3 , 3	1 , 4
Defect	4 , 1	2, 2

	Left	Right
Up	1 , 0	3 , 2
Down	2 , 1	4 , 0

Table 11: A gauche, Prisonner's Dilemma. Il y a un équilibre de Nash $(Defect, Defect)$ vers lequel les joueurs tendent naturellement. Mais il n'est pas optimal car dominé par $(Cooperate, Cooperate)$. A droite, Stackelberg Game. Il y a un équilibre de Nash $(Down, Left)$ non optimal et dominé par $(Up, Right)$.

Par ailleurs, Michael Littman et Peter Stone montrent un algorithme calculant l'équilibre de Nash d'un jeu 2x2 répété. L'algorithme est polynomial en temps en fonction de la "taille" des nombres situés dans la matrice du jeu [43].

3.27 Hedged Learning (Chang & Kaelbling 2005)

Dans [17], les auteurs combinent l'algorithme Hedge et la minimisation du regret sur des jeux répétés avec des algorithmes experts et utilisant des modèles de l'adversaire.

3.28 Convergence + No-regret (Bowling 2005)

Dans [9], Michael Bowling présente l'importance exagérée de la convergence (convergence vers quoi?) et l'importance de la minimisation du regret. Il étudie alors comment combiner convergence et regret nul.

3.29 L'année 2006

En 2006, Alain Dutech, Raghav Aras et François Charpillet présentent la coordination par les jeux stochastiques [25]. Olivier Gies et Brahim Chaib-draa présentent la coordination multiagent avec une méthode utilisant le Q-learning et les "jeux" adaptatifs [30]. Cet article contient des erreurs mais ses deux points positifs sont, un de ré-utiliser l'idée de mémoire limitée, deux, d'étudier deux jeux réalistes simplifiés plus complexes que les jeux matriciels 2x2. Etudier des jeux matriciels 2x2 offre l'avantage d'isoler un problème théorique. Etudier des jeux réalistes simplifiés tels que le jeu de la coordination sur grille 3x3 ou le jeu de la poursuite sur grille 5x5 est un peu plus difficile (donc potentiellement intéressant) mais la complexité existante sur ces jeux simples cache les vrais problèmes de la coordination multiagent. A la fin, le lecteur ne sait pas très bien quel est l'apport de l'article. L'article est un peu maladroit. Un, il désigne par jeu adaptatif ce qui est en fait une stratégie de joueur. Deux, il dit que les algorithmes actuels sont limités car incapables de gérer la multiplicité des équilibres de Nash et de converger vers l'équilibre Pareto optimal. Je n'en suis pas certain: cf M-Cube ou le S algorithm. Trois, les expériences sont faites en auto-jeu seulement. Quatre, page 399, la figure 3 donnent 5 couples de trajectoires en disant qu'ils correspondent à des équilibres de Nash alors que seuls 3 sur les 5 en sont effectivement. Les deux premiers couples de la figure 3 avec (90, 80)ne sont pas des éqilibres de Nash car la trajectoire 1 n'est pas la meilleure réponse de la trajectoire 2. Cela casse un peu les résultats des expériences mesurant des pourcentages d'atteintes d'équilibres de Nash Paréto-optimaux sur l'ensemble des équilibres de Nash. Cinq, pour démontrer clairement l'apport de l'algorithme (atteindre les équilibres de Nash Pareto-optimaux là où le Qlearning simple échoue), il manque des expériences sur des jeux matriciels.

3.30 Principes CoLF et CK (Munoz de Cote & al 2006)

A AAMAS 2006, Enrique Munoz de Cote, Alessandro Lazaric, et Marcello Restelli présentent une approche d'apprentissage de la coopération dans le jeu du dilemme social [48]. Le jeu du dilemme social est celui de [64]. Les auteurs définissent deux nouveaux principes: CoLF (= Change or Learn Fast) inspiré de [11] et CK (= Change and Keep). Pour moi, bien que les auteurs disent que CoLF est inspiré de WoLF, CoLF est l'opposé de WoLF. En effet, si l'environnement est non stationnaire, CoLF utilise un pas d'apprentissage faible (donc n'apprend pas), et si l'environnement est stationnaire, CoLF utilise un grand pas d'apprentissage, donc apprend vite, d'accord, mais ne peut pas

converger (?). Pour moi, intuitivement, j'ai l'impression qu'il faudrait faire le contraire (!), comme le fait WoLF. Le principe CoLF, en plus de la Q-value du Q learning, utilise une P-value, moyenne coulissante des retours, et une S-value, moyenne coulissante des différences entre les retours et les P-values. La non-stationarité est détectée si la différence entre les retours et les P-values dépassent la S-value. L'idée est bien. A essayer. Le principe CK repose sur l'idée que, lorsque soi-même on change de stratégie, on la garde (Keep) un temps pour laisser le temps aux autres apprenants (d'apprendre?) et on ne met plus a jour la Q-value. Etant donné que l'apprenant reste, me semble-t-il, un seul pas de temps dans l'état Keep, je ne vois pas bien l'intérêt de ce principe; les autres agents n'ont pas le temps de s'adapter en un pas de temps (?). Peut-être est-ce lié au fait d'avoir des états de MDP qui sont les actions jointes du temps précédent? Le fait de garder la même action pendant un temps permet aux autres agents de s'adapter sur du stable. Par ailleurs, la formule du MASD [64] est utilisée dans cet article. A titre d'info et d'exercice, j'ai cherché la correspondance affine ($R = \lambda \times formule + \mu$) entre la formule du MASD et les retours R des cases de la matrice du jeu du dilemme du prisonnier (cf jeu de gauche dans la table 12). Je trouve évidemment $N = 2$ et $M = 1$ et aussi $\lambda = \mu = 2$ et $k = 3/4$. Pour $k = 1/2$, les agents du MASD sont encouragés à être altruistes, et le MASD correspondant à $k = 1/2$ est celui du milieu dans la table 12. Pour $k = 1 - \epsilon$, les agents sont encouragés à être égoïstes et le MASD correspondant à $k = 0.9$ est celui de droite dans la table 12.

	Cooperate	Defect	Cooperate	Defect	Cooperate	Defect
Cooperate	3 , 3	1 , 4	3 , 3	2 , 3	3 , 3	-2 , 7
Defect	4 , 1	2 , 2	3 , 2	2 , 2	7 , -2	2 , 2

Table 12: A gauche, le dilemme du prisonnier usuel . Au centre, le cas avec une coopération évidente et sans dilemme ($k = 1/2$): $(Cooperate, Cooperate)$ est à la fois l'équilibre de Nash et Paréto optimal. A droite, le cas assez extrême ($k = 0.9$) dans lequel les joueurs égoïstes et défectueux sont bien récompensés; $(Cooperate, Cooperate)$ est Paréto optimal mais difficile à atteindre; $(Defect, Defect)$ est un équilibre de Nash très stable.

Concernant les résultats, ils ne sont appliqués qu'au MASD. Une série de courbes montrent des résultats dépendants de γ, le taux de réduction, avec k fixé mais dont la valeur n'est pas donnée. Une série de courbes montrent des résultats dépendants de k avec γ fixé mais dont la valeur n'est pas donnée. Quand $k >= 0.90$ aucun des principes, CoLF ou CK, ne permettent de convergence. Certaines courbes correspondent à $k = 1/3$, alors que $1/2 <= k < 1$ (!). Finalement, je trouve cet article maladroit mais contenant une idée potentiellement intéressante, à peut-être essayer un jour, pour mesurer la non-stationarité avec des P-values et des S-values. NB: ne pas confondre les S-values avec les aspirations du S algorithme [64].

3.31 CorrStrategy (Vu & Powers & Shoham 2006) +

En 2006, Thuc Vu, Rob Powers et Yoav Shoham publient l'algorithme "CorrStrategy" [70]. Ils publient le critère d'optimalité ciblée. Ce critère prend en compte la classe d'adversaires de l'apprenant, et l'algorithme donne une best-response en fonction de la classe d'adversaires. Leur travail prend en compte le cas de N agents (et pas seulement 2). Ces agents sont soient des copies de l'agent (self-agent) soient des agents appartenant à une classe d'agents donnée. Le critère d'optimalité ciblée consiste à être optimal dans ce cas cible. "CorrStrategy" est un peu bizarre car elle commence par une phase initiale de signalement dans laquelle les self-agents jouent une séquence fixée à l'avance. Ainsi, à la fin de cette phase les agents savent qui est un self-agent et qui ne l'est pas, avec une probabilité très grande fonction de la longueur de la séquence. Est-ce tricher ? Oui dans un sens, parce que l'objectif de cette phase est la communication, et pas la maximisation des retours. Il faudrait mesurer le regret amené par cette phase. Non dans un autre sens, car cette communication passe par les actions et pas par un média dédié à la communication. Si tous les agents appartiennent à la classe cible, alors CorrStrategy suppose que l'ensemble des agents s'insère dans un environnement stationnaire, et il cherche la meilleure réponse à cet environnement. Sinon (si il y a au moins un autre self-agent), alors CorrStrategy utilise de la programmation linéaire pour trouver les équilibres Pareto-optimaux, puis les self-agents essaient de se coordonner vers les mêmes actions jointes. L'article de dit pas comment. L'approche est prouvée théoriquement (CorrStrat vérifie les critères) et testée expérimentalement sur GAMUT [53].

3.32 Teach, Partition, Coordinate, Monitor TPCM (Powers & al 2007) +

En 2007, Rob Powers, Yoav Shoham et Thuc Vu re-présentent leur "nouveau" critère général et un cadre algorithmique pour l'apprentissage dans les SMA [56]. L'algorithme 1 se nomme PCM(S) = Partition, Coordinate, and Monitor pour les adversaires S. L'algorithme 2 se nomme TPCM(A) = Teach, Partition, Coordinate, and Monitor for the target classe of opponents A. L'approche repose sur des modules dépendants des cas: jeux à 2 ou n joueurs, adversaires stationnaires ou adaptatifs. L'approche est testée sur GAMUT [53].

3.33 AIJ special issue (May 2007)

A la suite du succès grandissant de MAL, surtout depuis 2003 2004, un numéro spécial de Artificial Intelligence a été publié sur ce thème en Mai 2007. Le numéro est structuré autour de l'article de Yoav Shoham, Rob Powers et Trond Grenager [61]. Cet article dresse un état de l'art et propose 5 agendas: "computationnel" (utiliser l'ordinateur pour déterminer des équilibres de Nash, corrélés, Pareto-optimaux), "descriptif" (l'objectif est de modéliser les êtres humains multi-apprenants), "normatif" (dans le formalisme de la théorie des jeux, comprendre le liens entre les règles d'apprentissage et les équilibres correspondants),

"prescriptif coopératif" (comment apprendre dans le cadre multi-agent avec communication, il s'agit de résolution de problèmes distribués) et enfin "prescriptif non coopératif" (comment apprendre dans le cadre multi-agent sans communication, c'est-à-dire le cadre qui nous intéresse). Cet article est très bien car c'est une synthèse, bien utile quand on découvre un état de l'art. Les articles suivants discutent de ce découpages en 5 catégories de problème. Certains articles du numéro spécial détaillent un agenda, d'autres remettent en cause le découpage, d'autres l'affine, d'autres proposent de nouveaux agendas, notamment les agendas de "modélisation", de "conception" et d'"ingénierie" qui mettent l'accent sur le passage d'un problème réel au formalisme théorique utilisé dans l'état de l'art. Bref les autres articles remettent en cause les cinq agendas, ce qui était le but de Shoham, Powers et Grenager: proposer des agendas pour susciter des réactions et faire évoluer l'état de l'art.

Le deuxième article de Tuomas Sandholm [58] montre, entre autres choses, que l'agenda computationnel (calculer et trouver les équilibres d'un jeu indépendamment des méthodes d'apprentissage) est déjà bien avancé. Il existe beaucoup d'outils mathématiques permettant de trouver les équilibres de Nash. Il existe notamment une méthode mathématique pour trouver les équilibres de nash d'un jeu matriciel à n joueurs et à somme quelconque [45]. Le logiciel Gambit implémente cette méthode.

4 Apprentissage mono-agent

Cette partie cite des références sur l'apprentissage mono-agent, récupérées récemment.

4.1 VAPS (Baird & Moore 1998), Prioritized Sweeping (Moore & Atkeson 1993)

Complètement indépendamment de l'ARMA, en 1998, Leeson Baird et Andrew Moore publient un résultat théorique très général sur l'AR [3]: ils montrent comment voir l'AR comme une descente de gradient stochastique sur une fonction d'erreur dépendante de l'algorithme (Qlearning, SARSA, Reinforce, etc.). Cette classe d'algorithmes, appelée VAPS (= Value Action Policy Search), donne le résultat naturel: convergence locale (comme BACKPROP). Ce résultat est valable dans tous les cas d'observabilité (complète ou partielle) et dans le cas d'apprentissage de fonction de valeurs ou celui de recherche directe de politique (Qlearning, TD apprennent des fonctions de valeur alors que DYNA, Reinforce apprennent directement des politiques).

Par ailleurs, j'ai récupéré des articles d'AR mono-agent: Priorited Sweeping [47] de Moore et Atkeson.

4.2 E3 (Kearns & Singh 1998)

En 1998, Michael Kearns et Satinger Singh proposent l'algorithme E3 (E cube) qui réussit à obtenir un retour "quasi-optimal" dans les PDM en un temps

polynomial [40]. L'idée est d'avoir trois sortes de noeuds: les noeuds jamais traversés (qui n'existent pas), les noeuds "connus" pour lesquels les statistiques sur les retours sont suffisantes, et les noeuds "inconnus" par lesquels on est déjà passé quelques fois mais où les statistiques sont insuffisantes. Quand un nouveau noeud devient connu (ce qui est obligatoire au nom du "pigeonhole principle"), au moyen de l'algorithme d'évaluation de politique, l'algorithme calcule d'une part la fonction de valeurs optimale pour exploiter maximalement les retours observés dans les états connus, et d'autre part la fonction de valeur pour explorer maximalement, c'est-à-dire atteindre le plus rapidement possibles les noeuds jamais atteints. Si la politique d'exploitation calculée est "presque" optimale, il l'exécute, sinon il exécute la politique d'exploration calculée. L'algorithme explicitement exploite ou explore, d'où son nom "Explicit Exploit or Explore". Il est prouvé converger en un temps polynomial.

4.3 L'année 2000

En 2000, Satinder Singh, Tommi Jaakkola, Michael Littman et Csaba Szepesvari donnent des résultats de convergence d'algorithmes d'apprentissage par renforcement (mono-agent) en montrant l'importance des stratégies d'exploration [63]. Ils définissent les stratégies d'exploration "décroissantes" et les stratégies d'exploration "persistantes". Les stratégies décroissantes offrent l'avantage de pouvoir converger mais elles ont du mal à s'adapter à des environnements non stationnaires. Les stratégies persistantes peuvent s'adapter à la non-stationnarité. Les auteurs nomment "GLIE" les stratégies qui sont gloutonnes à la limite (GLIE = Greedy in the Limit of Infinite Exploration) et "RRR" les stratégies utilisant une méthode (très utilisée en pratique) basée sur les rangs des actions (RRR = Restricted Ranked-based Randomized). Ils mentionnent aussi l'exploration de Boltzmann.

5 Apprentissage bayésien

Il est difficile de faire une partition en sous-sections avec une sous-section apprentissage bayésien. Cette sous-section rassemble simplement les références contenant de l'apprentissage bayésien.

5.1 Les années 90

En 1993, Ehud Kalai et Ehud Lehrer publient un article montrant que, dans les jeux répétés, des croyances subjectives des joueurs mises à jour de manière bayésienne font que l'on peut prédire les stratégies des joueurs et donc qu'un joueur peut se diriger vers un équilibre de Nash [38]. Qui plus est, si les joueurs ne disposent que d'une information incomplète sur les matrices de retour de leurs adversaires, cela marche encore. Cet article est théorique et aucune expérience n'est effectuée.

En 1997, Dov Monderer et Moshe TennenHoltz décrivent un agent non bayésien qui joue répétitivement un jeu à information incomplète contre la nature [46]. L'article se démarquant de Bayes explicitement par son titre, je le place tout de même dans la sous-section bayésienne (!).

5.2 Factored MDP (Boutilier & al 1999)

En 1999, Craig Boutilier, T Dean et S Hanks présentent un article de fond sur les PDM "factorisés" [7]. L'idée est de dire que la plupart des actions effectuées par un agent sont indépendantes les unes des autres. Pour cela, parmi d'autres outils, les réseaux bayésiens sont adaptés pour modéliser ce type de PDM. Le PDM est remplacé par un PDM factorisé, plus simple. Cet article est pédagogique et fondateur dans la suite. En 1999, Michael Kearns et Daphne Koller publient l'algorithme DBN-E3 [39] qui adapte l'algorithme E3 [40] aux PDM factorisés [7].

5.3 Après 1999

Par ailleurs, dans l'idée d'utiliser l'approche bayésienne, Dearden, Friedman et André publient un article sur l'exploration bayésienne dans l'AR basé sur les modèles [24]. Modèle est à prendre au sens des retours (r) et des transitions (t) dans le formalisme de PDM l'AR mono-agent (par opposition a "model-free" pour l'AR mono-agent classique avec apprentissage de fonctions de valeurs V ou Q). L'idée ressemble à celle de E3 [40] ou Rmax [13] dans lesquels on apprend le modèle, puis on calcule la politique optimale pour le modèle appris avec la programmation dynamique (value iteration). Dans [24], Dearden, Friedman et André utilisent "prioritized sweeping". L'agent possède un état de croyance et tient à jour les probabilités d'un PDM donné connaissant ces croyances. A chaque retour et transitions observés, l'état de croyance change et les probabilités des PDM aussi.

En 2003, Gerald Tesauro a proposé Hyper-Q [69]. L'idée est de ne pas utiliser les fonctions de récompense et les fonctions Q des autres agents. Au contraire les valeurs des stratégies mixtes sont apprises à la place des actions de base, et les actions des autres agents sont estimées avec les actions observées et l'inférence bayésienne. Georgios Chalkiadakis et Craig Boutilier présentent une approche bayésienne de la coordination [15].

6 Conclusion

Ce document a décrit l'état de l'art de l'apprentissage par renforcement multi-agent. Il a séparé les références anciennes issues de la théorie des jeux, les références sur l'apprentissage multi-agent et sur les jeux stochastiques, les références sur l'apprentissage par renforcement mono-agent et enfin les références sur l'apprentissage bayésien multi-agent.

Il est difficile de sélectionner les références importantes de l'apprentissage multi-agent. Néanmoins, j'aime bien: la méthode de Julia Robinson pour trouver la valeur minimax d'un jeu à deux joueurs à somme nulle [57], l'article fondateur de John Nash sur les jeux non-coopératifs [51], la définition par Aumann des équilibres corrélés [2], la méthode sans regret de Sergiu Hart et Andreu Mas-Colell convergente vers les équilibres corrélés [36], l'article précurseur de Ming Tan sur les proies et les prédateurs [68], la série des algorithmes dérivés du Q-learning, l'étude des jeux 2x2 sous l'angle de la descente de gradient de Singh, Kearns et Mansour [62], Bully et Godfather [66] de Littman et Stone, les principes de rationalité et de convergence de Bowling et Veloso [11], la classification des agents suivant l'historique et la croyance de Chang et Kaelbling [16], la base de jeux de test GAMUT [53], MetaStrategy [54], Manipulator [55] et CorrStrat [70] de Powers et Shoham, les factored MDP [7], M-Qubed de Crandall et Goodrich [23] basé sur le satisficing algorithm, l'article de synthèse sur les 5 agendas de [61], la thèse de Michael Bowling [10].

Je suis débordé par les articles mathématiques, cités par [58], et consistant à calculer les équilibres de Nash par des méthodes autres que l'apprentissage, cf par exemple l'état de l'art de McKelvey et McLennan [45].

En apprentissage mono-agent, j'ai aimé Rmax de Brafman et Tennenholtz [13], E3 de Kearns et Singh [40] pour l'idée d'une exploration exploitative, et VAPS [3] pour la preuve de convergence locale des algorithmes de RL.

Reste à laisser reposer tout cela et voir.

References

[1] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. Gambling in a rigged casino: the adversarial multi-armed bandit problem. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 322–331. IEEE Computer Society Press, Los Alamitos, CA, 1995.

[2] Robert Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1:67–96, 1974.

[3] L. Baird and A. Moore. Gradient descent for general reinforcement learning. In *NIPS*, 1998.

[4] B. Banerjee and J. Peng. Performance bounded reinforcement learning in strategic interactions. In *AAAI*, pages 2–7, 2004.

[5] Bikramjit Banerjee, Sandip Sen, and Jing Peng. Fast concurrent reinforcement learners. In *IJCAI*, pages 825–832, 2001.

[6] Darse Billings. Thoughts on roshambo. *ICGA Journal*, 23(1):3–8, 2000.

[7] C. Boutilier, T. Dean, and S. Hanks. Decision theoretic planning: Structural assumptions ans computational leverage. *Journal of Artificial Intelligence Research*, 1999.

[8] M. Bowling. Convergence problems of general-sum multi-agent reinforcement learning. In *ICML*, pages 89–94, 2000.

[9] M. Bowling. Convergence and no-regret in multiagent learning. In *NIPS*, 2004.

[10] Michael Bowling. *Multiagent Learning in the Presence of Agents with Limitations*. PhD thesis, CMU, 2003.

[11] Michael H. Bowling and Manuela M. Veloso. Rational and convergent learning in stochastic games. In *IJCAI*, pages 1021–1026, 2001.

[12] Michael H. Bowling and Manuela M. Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136(2):215–250, 2002.

[13] R. Brafman and Moshe Tennenholtz. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. In *IJCAI*, 2001.

[14] David Carmel and Shaul Markovich. Learning models of intelligent agents. In Howard Shrobe and Ted Senator, editors, *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference, Vol. 2*, pages 62–67, Menlo Park, California, 1996. AAAI Press.

[15] G. Chalkiadakis and C. Boutilier. Coordination in multiagent reinforcement learning: A bayesian approach. In *AAMAS*, 2003.

[16] Yu-Han Chang and Leslie Pack Kaelbling. Playing is believing: the role of beliefs in multi-agent learning, 2001.

[17] Yu-Han Chang and Leslie Pack Kaelbling. Hedged learning: regret-minimization with learning experts. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 121–128, New York, NY, USA, 2005. ACM Press.

[18] C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multi-agent systems. In *AAAI*, pages 746–752, 1998.

[19] V. Conitzer and T. Sandholm. AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. In *ICML*, pages 83–90, 2003.

[20] Jacob W. Crandall and Michael A. Goodrich. Estabishing reputation using social commitment in repeated games. In *AAMAS Workshop on Learning and Evolution in Agent based Systems*, 2004.

[21] Jacob W. Crandall and Michael A. Goodrich. Learning ϵ-pareto efficient solutions with minimal knowledge requirements using satisficing. In *AAAI Spring Symposium on Artificial Multiagent Learning*, 2004.

[22] Jacob W. Crandall and Michael A. Goodrich. Multiagent learning during on-going human-machine interactions: The role of reputation. In *AAAI Spring Symposium: Interaction between Humans and Autonomous Systems over Extended Operation*, Stanford, California, 2004.

[23] Jacob W. Crandall and Michael A. Goodrich. Learning to compete, compromise, and cooperate in repeated general-sum games. In *ICML*, pages 161–168, New York, NY, USA, 2005. ACM Press.

[24] Richard Dearden, Nir Friedman, and David Andre. Model based bayesian exploration. In *UAI*, pages 150–159, 1999.

[25] A. Dutech, R. Aras, and F. Charpillet. Coordination par les jeux stochastiques. In *CARI*, 2006.

[26] Dan Egnor. Iocaine powder. *ICGA Journal*, 23(1):33–35, 2000.

[27] Dean Foster and Rakesh Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behaviour*, 21:40–55, 1997.

[28] Dean Foster and Rakesh Vohra. Regret in the on-line decision problem. *Games and Economic Behaviour*, 29:7–35, 1999.

[29] D. Fudenberg and K. Levine. *The theory of Learning in Games*. MIT Press, Cambridge, 1998.

[30] O. Gies and B. Chaib-Draa. Apprentissage de la coordination multiagent, une méthode basée sur le Q-learning par jeu adaptatif. *RIA*, 20:385–412, 2006.

[31] Amy Greenwald and Keith Hall. Correlated-Q learning. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 242–249, 2003.

[32] Amy Greenwald, Amir Jafari, Gunes Ercal, and David Gondek. On no-regret learning, fictitious play, and nash equilibrium. In *ICML*, pages 226–233, 2001.

[33] Carlos Guestrin, Daphne Koller, and Ronald Parr. Multiagent planning with factored MDPs. In *NIPS*, 2001.

[34] Carlos Guestrin, Michail Lagoudakis, and Ronald Parr. Coordinated reinforcement learning. In *ICML Proceedings*, 2002.

[35] Carlos Guestrin, Shobha Venkataraman, and Daphne Koller. Context-specific multiagent coordination and planning with factored MDPs. In *AAAI Symposium*, 2002.

[36] Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, September 2000.

[37] J. Hu and M. Wellman. Multi-agent reinforcement learning: theoretical framework and an algorithm. In *ICML*, pages 242–250, 1998.

[38] Ehud Kalai and Ehud Lehrer. Rational learning leads to nash equilibrium. *Econometrica*, 61(5):1019–45, September 1993. available at http://ideas.repec.org/a/ecm/emetrp/v61y1993i5p1019-45.html.

[39] M. Kearns and D. Koller. Efficient reinforcement learning in factored MDPs. In *IJCAI*, 1999.

[40] M. Kearns and S. Singh. Near-optimal reinforcement learning in polynomial time. In *ICML*, 1998.

[41] M. Littman. Markov games as a framework for multi-agent reinforcement learning. In *ICML*, pages 157–163, 1994.

[42] M. Littman. Friend-or-foe Q-learning in general-sum games. In *ICML*, 2001.

[43] Michael L. Littman and Peter Stone. A polynomial-time Nash equilibrium algorithm for repeated games. *Decision Support Systems*, 39:55–66, 2005.

[44] Peter McCracken and Michael Bowling. Safe strategies for agent modelling in games. In *Proceedings of AAAI Symposium*, 2004.

[45] Richard McKelvey and Andrew McLennan. Computation of equilibria in finite games. *Handbook of Computational Economics*, 1996.

[46] D. Monderer and M. Tennenholtz. Dynamic non-bayesian decision making. *Journal of Artificial Intelligence Research*, 7:231–248, 1997.

[47] Andrew W. Moore and Christopher G. Atkeson. Prioritized sweeping: Reinforcement learning with less data and less time. *Machine Learning*, 13:103–130, 1993.

[48] E. Munoz, A. Lazaric, and M. Restelli. Learning to cooperate in multi-agent social dilemmas. In *AAMAS*, 2006.

[49] J. Nash. The bargaining problem. *Econometrica*, 28:155–162, 1950.

[50] J. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36:48–49, 1950.

[51] J. Nash. Non cooperative games. *Annals of Mathematics*, 54:286–295, 1951.

[52] J. Nash. Two-person cooperative games. *Econometrica*, 21:128–140, 1953.

[53] E. Nudelman, J. Wortman, Y. Shoham, and K. Leyton-Brown. Run the gamut: a comprehensive approach to evaluating game-theoretic algorithms. In *AMAAS*, 2004.

[54] R. Powers and Y. Shoham. New criteria and a new algorithm for learning in multiagent systems. In *AMAAS*, 2004.

[55] R. Powers and Y. Shoham. Learning against opponents with bounded memory. In *IJCAI*, 2005.

[56] Rob Powers, Yoav Shoham, and Thuc Vu. A general criterion and an algorithmic framework for learning in multi-agent systems. *Machine Learning*, 67(1-2):45–76, 2007.

[57] J. Robinson. An iterative method of solving a game. *Annals of Mathematics*, 44:296–301, 1951.

[58] Tuomas Sandholm. Perspectives on multiagent learning. *Artificial Intelligence Journal*, 171:382–391, 2007.

[59] L. Shapley. Stochastic games. *Proceedings of National Academy of Science*, 39:1095–1100, 1953.

[60] Y. Shoham and R. Powers. Multi-agent reinforcement learning; a critical survey. Technical report, Stanford University, 2003.

[61] Yoav Shoham, Rob Powers, and Trond Grenager. If multi-agent learning is the answer, what is the question? *Artificial Intelligence*, 171:365–377, 2007.

[62] Satinder Singh, Michael Kearns, and Yishay Mansour. Nash convergence of gradient dynamics in general-sum games. In *UAI*, pages 541–548, 2000.

[63] Satinder P. Singh, Tommi Jaakkola, Michael L. Littman, and Csaba Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning*, 38(3):287–308, 2000.

[64] Jeffrey L. Stimpson and Michael A. Goodrich. Learning to cooperate in a social dilemma: a satisficing approach to bargaining. In *ICML*, 2003.

[65] Jeffrey L. Stimpson and Michael A. Goodrich. Nash equilibrium or nash bargaining? choosing a solution concept for multi-agent learning. In *AAMAS Workshop on Game Theoretic and Decision Theoretic Agents*, Melbourne, Australia, 2003.

[66] Peter Stone and Michael L. Littman. Implicit negotiation in repeated games. In John-Jules Meyer and Milind Tambe, editors, *Pre-proceedings of the Eighth International Workshop on Agent Theories, Architectures, and Languages (ATAL-2001)*, pages 96–105, 2001.

[67] R. Sutton and A. Barto. *Reinforcement Learning: an introduction*. MIT Press, 1998.

[68] M. Tan. Multi-agent reinforcement learning: independent vs cooperative agents. In *ICML*, pages 330–337, 1993.

[69] G. Tesauro. Extending Q-learning to general adaptive multi-agent systems. In *ICML*, 2003.

[70] Thuc Vu, Rob Powers, and Yoav Shoham. Learning against multiple opponents. In *AAMAS*, pages 752–759, New York USA, 2006. ACM Press.

[71] C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.

[72] M. Weinberg and J. Rosenschein. Best-response multiagent learning in non-stationary environments. In *AMAAS*, 2004.

[73] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of ICML'03*, 2003.