
Le bridge, nouveau défi de l'intelligence artificielle ?

Véronique Ventos, Olivier Teytaud

LRI, INRIA, Univ Paris-Sud, CNRS, Université Paris-Saclay, France
ventos@lri.fr,olivier.teytaud@inria.fr

RÉSUMÉ. Le jeu de bridge est un jeu extrêmement complexe et ce aussi bien pour les humains que pour les programmes de bridge. Contrairement au jeu de Go où l'IA AlphaGo a récemment atteint le niveau de grand maître, les IA de bridge sont encore loin des meilleurs joueurs humains. La différence essentielle entre les deux jeux porte sur l'aspect information incomplète omniprésent dans le bridge. Un programme expert de ce jeu pourrait donc réaliser des tâches différentes et complémentaires de celles traitées par AlphaGo. La première partie de cet article est consacrée à la présentation des différents aspects du bridge et de quelques pistes de réflexion sur les différents challenges qui lui sont inhérents ainsi que des résultats obtenus par la communauté bridge informatisé. Dans la seconde partie, nous présentons nos travaux concernant l'adaptation au bridge d'une méthode récente permettant d'optimiser des IA de jeux en recherchant une graine aléatoire meilleure que les autres sur le jeu concerné. L'IA Wbridge5 développée par Yves Costel a été boostée avec la meilleure graine trouvée à l'issue de ces expériences et a remporté les championnats du monde de robots de bridge en septembre 2016.

ABSTRACT. The game of bridge is very complex both for humans and for computer bridge programs. While the AI Go AlphaGo recently reached the level of grandmaster go, it is not the case for bridge where robots are still far from the best human players. The main difference between the two games is that bridge is a partially observable game. An expert program on bridge could therefore perform tasks that are different from and complementary to those handled by AlphaGo. The first part of this article is devoted to the presentation of various aspects of the bridge and some thoughts on the numerous challenges inherent to them together with a state of art of the computer bridge community. In the second part, we present our work related to the adaptation to bridge of a recent method using for boosting game AIs by seeking a random seed better than the others on a particular game. The AI bridge Wbridge5 developed by Yves Costel has been boosted with the best seed found on the outcome of these experiments and won the world championship bridge robots in September 2016.

MOTS-CLÉS : bridge informatisé, apprentissage automatique, monte-carlo, amélioration d'IA.

KEYWORDS : computer bridge, machine learning, monte-carlo, boosting AI.

DOI:10.3166/RIA.31.249-279 © 2017 Lavoisier

1. Introduction

« Le bridge est un jeu de cartes, de type jeu de levées, consistant à comptabiliser le nombre de levées réalisées. Il se pratique avec un jeu de 52 cartes, par quatre joueurs, opposant deux équipes composées de deux partenaires » (wikipédia).

Si l'on s'en tient à cette définition, le bridge paraît bien inoffensif, pourtant nous voulons proposer ce jeu comme nouveau challenge de l'IA. En effet, le bridge est un jeu extrêmement complexe et ce aussi bien pour les humains que pour les robots de bridge qui ne dépassent pas le niveau de bons amateurs.

La conception d'un robot joueur de go de niveau grand maître a été pendant des années un défi majeur pour les chercheurs en intelligence artificielle. Le nombre très élevé de combinaisons (de l'ordre de 10^{600}) rend impossible l'utilisation de méthodes traditionnelles de type *force brute* efficaces sur d'autres jeux comme les échecs ou les dames anglaises. Ceux-ci ont bénéficié des progrès liés à la puissance des ordinateurs qui ont permis la conception de *superordinateurs* pouvant évaluer un nombre gigantesque de positions par seconde. Le plus connu est le superordinateur *Deep Blue* développé par IBM qui a battu en 1997 le champion du monde d'échecs de l'époque Garry Kasparov. Cependant, c'est le programme de dames anglaises *Chinook* qui est le premier à avoir obtenu un titre de champion du monde contre un humain en 1994. Dans le cas des dames anglaises, l'approche basée sur une expertise humaine¹ a conduit à la résolution complète du jeu avec la définition d'une stratégie de jeu infaillible et une preuve publiée dans (Schaeffer *et al.*, 2007) que si les deux adversaires jouent parfaitement le meilleur résultat possible sera une partie nulle. Les dames anglaises est donc un jeu résolu ce qui n'est pas le cas des échecs même si le superordinateur d'échecs Hydra (200 millions de positions évaluées par seconde) a obtenu un classement jamais atteint par les humains. Le classement est basé sur le calcul de points Elo utilisés pour comparer les performances des joueurs. Au premier janvier 2017, onze joueurs seulement ont dépassé 2 800 points dans leur carrière sans jamais atteindre 2900. Hydra a obtenu près de 3000 points Elo et les programmes d'échecs plus récents comme Stockfish, Houdini ou Komodo ont dépassé 3200 Elo.

En ce qui concerne les programmes de Go, le facteur de branchement de l'arbre de recherche et la difficulté croissante pour évaluer une position rendent inefficaces les programmes de type force brute. Pour améliorer l'évaluation des positions, les programmes ont commencé à utiliser à partir de 1993 des méthodes de Monte-Carlo dont le principe a été inventé par Nicholas Metropolis et publié dans (Metropolis, Ulam, 1949). En dépit de cette amélioration et de la puissance de calcul des ordinateurs de l'époque, jusqu'en 2006 les meilleurs programmes ne dépassent pas le niveau d'un joueur amateur moyen. La première révolution est arrivée avec l'introduction de stratégies couplant l'utilisation de méthodes de Monte-Carlo et une adaptation au Go de l'algorithme UCT (*Upper Confidence bounds applied to Trees*) qui permet d'explorer plus souvent le meilleur coup dans un arbre de recherche. Les programmes Mogo

1. Les connaissances ont toutes été entrées manuellement dans le programme.

(projet TAO de l'INRIA Futurs et de l'école Polytechnique (Munos, Teytaud, 2006) et (Gelly *et al.*, 2012)) et CrazyStone (projet SEQUEL de l'Inria Futurs (Coulom, 2006)) ont utilisé cette nouvelle approche dès 2006, ce qui leur a permis de remporter des victoires contre des joueurs professionnels mais sur des plateaux plus petits que le goban classique (19X19) ou avec des pierres de handicap. En janvier 2016, une équipe de DeepMind, entreprise britannique spécialisée dans l'intelligence artificielle et les jeux, publie un article (Silver *et al.*, 2016) sur la présentation d'AlphaGo le premier programme de Go ayant battu un joueur professionnel sur un goban classique sans handicap. AlphaGo est un programme utilisant et intégrant différentes techniques d'apprentissage relativement récentes. Deux mois plus tard, l'IA AlphaGo devient grand maître après sa victoire par quatre parties à une dans un match l'opposant à Lee Sedol l'un des tout meilleurs joueurs de go au monde.

AlphaGo repose sur trois techniques : les réseaux de neurones profonds, les techniques de recherche de type monte-carlo et l'apprentissage renforcé ou RL pour Reinforcement Learning (Sutton, Barto, 1998). Deux réseaux profonds cohabitent, la tâche du premier est de prédire le prochain coup en utilisant les données de dizaines de milliers de parties d'excellents joueurs humains. Cette première étape d'imitation de l'humain permet à l'IA d'atteindre un niveau correct de jeu. Une fois ce niveau atteint, on fait jouer plusieurs millions de parties opposant l'IA à elle-même. Les nouvelles données générées durant cette étape serviront d'entrée au second réseau profond dont le but sera de renvoyer une évaluation d'un mouvement relativement à l'état courant du plateau. Des techniques d'apprentissage renforcé sont ensuite utilisées pour combiner les deux réseaux. La place de l'apprentissage automatique dans le processus est très importante ce qui est radicalement différent de l'approche utilisée par exemple par *Chinook* où aucune connaissance n'a été apprise par une méthode d'intelligence artificielle. A la suite de la victoire d'AlphaGo, beaucoup se sont demandé quel pourrait être le prochain défi de l'IA, le but étant d'étendre les champs d'application possibles en proposant la résolution de problèmes jusque-là non traités par un système d'IA. Un rapport récent (Heinrich, Silver, 2016) présente une application de Deep RL sur deux variantes du poker. Le poker est un jeu en information incomplète avec un aspect psychologique important. De plus, contrairement au jeu de go, le poker fait intervenir plus de deux joueurs. Dans la liste des prétendants figure aussi StarCraft, un jeu vidéo en temps réel où les joueurs n'ont pas une connaissance parfaite de leur environnement.

Nous pensons que définir un robot de bridge pouvant rivaliser avec les champions humains pourrait constituer une nouvelle étape pour l'intelligence artificielle. Les principales difficultés dans le cadre de ce jeu sont liées aux interactions entre les différentes étapes du jeu et au fait que les raisonnements sont réalisés en information incomplète. La diversité des modes de raisonnement nécessaires pour atteindre, voire dépasser, le niveau des meilleurs joueurs devrait conduire à l'élaboration d'une architecture hybride utilisant des méthodes liées aussi bien à la communauté computer game qu'à celle de l'apprentissage automatique numérique et symbolique. La définition même de cette architecture est un défi. En effet, il faut déterminer pour chaque tâche quelles sont les méthodes existantes les plus appropriées, étudier la faisabilité de leur adaptation au bridge, définir de nouvelles méthodes et enfin déterminer comment

intégrer les différentes parties du système. Cette architecture devra comprendre une adaptation des méthodes utilisées par AlphaGo.

Les avantages pour utiliser le bridge comme application sont les suivants :

- Les règles sont simples et connues de tous.
- Le jeu est séquentiel et facilement modélisable.
- Le score est incrémental, le bridge ne fait donc pas partie des jeux où l'issue est incertaine jusqu'à la fin de la partie.
- Il existe un important volume de données relatives à des parties jouées par des humains ou des robots dans un format standard. Un expert du bridge sachant interpréter les données peut effectuer un travail de classement suivant les tâches attendues.
- La chance intervient mais elle est gommée par la marque duplicate*² qui permet de faire jouer les mêmes données à des joueurs différents.
- Il n'y a pas de problèmes liés au temps réel.

La communauté bridge informatisée (ou computer bridge) est active depuis une quarantaine d'années mais le niveau des robots est loin des meilleurs joueurs humains comme c'était le cas pour le Go il y a quelques années. Les principales difficultés inhérentes au bridge sont relatives aux éléments suivants :

- C'est un jeu à information incomplète.
- Le bridge se joue à quatre (deux équipes de deux).
- Il y a une nécessité d'explicitier les raisonnements que l'on fait.
- Des facteurs psychologiques interviennent.
- L'expertise humaine ne peut suffire à résoudre le jeu.
- L'adaptation à un partenaire est longue et essentielle, elle nécessite une phase d'apprentissage y compris chez les champions.

Une partie de ces difficultés semble difficilement traitable en n'utilisant que les méthodes utilisées par AlphaGo, une IA experte du bridge pourrait donc réaliser des tâches différentes et complémentaires de celles traitées par AlphaGo.

Un des objectifs de cet article est de familiariser le lecteur aux différents aspects du bridge et de faire ressortir des problématiques qui pourront être résolues avec plusieurs techniques existantes ou à développer. C'est l'objet de la première partie de cet article. Plus précisément, la section 2 présente les bases du bridge et de ses différentes phases de jeu ainsi que les raisonnements qu'un joueur humain doit mettre en place pour résoudre les différents problèmes auxquels il est confronté. Les caractéristiques du bridge par rapport aux notions de la théorie des jeux sont décrites et commentées dans la section 3 dédiée au bridge informatisé avec un état de l'art et un bilan des résultats obtenus par cette communauté. La section 4 présente un ensemble de challenges à relever pour améliorer les programmes de bridge.

2. Un glossaire est donné en annexe A, la première occurrence des mots y figurant est suivie d'un astérisque.

La deuxième partie de l'article est dédiée à la présentation de nos travaux en cours et futurs sur le sujet. Le bridge étant à information incomplète les approches les plus fréquentes utilisent des simulations de Monte-Carlo. Ces simulations utilisent des générateurs de nombres pseudo-aléatoires dépendants de la graine utilisée. Dans la section 5, nous présentons nos travaux concernant l'adaptation au bridge d'une méthode récente permettant d'optimiser des IA de jeux en recherchant une graine aléatoire meilleure que les autres sur le jeu concerné (St-Pierre, Teytaud, 2014; Cazenave *et al.*, 2015). L'IA de bridge Wbridge5 d'Yves Costel boostée avec la meilleure graine trouvée à l'issue de ces expériences a participé aux derniers championnats du monde des robots de bridge et les a remportés. La dernière section décrit brièvement nos travaux en cours sur la définition d'un cadre d'apprentissage supervisé visant à améliorer la prise de décisions des robots en fonction du contexte et donne un aperçu des perspectives de recherche qui sont nombreuses.

2. Le bridge

Nous présentons les bases du bridge dans cette section. Pour plus de détails, il existe des tutoriels interactifs en ligne³ permettant de se familiariser avec le jeu.

2.1. Description du bridge

Le bridge est un jeu de 52 cartes qui oppose deux paires de joueurs. Les cartes sont distribuées aléatoirement aux quatre joueurs qui ne voient que leur *main** c'est à dire les treize cartes de leur jeu. Les règles sont connues des joueurs et sont relativement simples. Malgré ceci et en raison de l'aspect information incomplète le bridge est un jeu très compliqué et ce aussi bien pour les humains que pour les programmes de bridge. Le bridge comprend deux phases bien distinctes : les enchères et le jeu de la carte. Les raisonnements effectués lors de la seconde phase utilisent les informations hypothétiques dévoilées lors de la phase d'enchères.

Assimiler les notions de base prend en général plusieurs années aux humains et les méthodes d'apprentissage par exemple de l'Université du Bridge⁴ préconisent de démarrer par une version bridée du jeu appelée mini-bridge où seul le jeu de la carte est réalisé.

2.1.1. La phase d'enchères

En bref : à l'issue des enchères, une des paires aboutit à un contrat final qui détermine l'éventuelle couleur d'atout* (il peut ne pas y en avoir, on parle alors de contrat à sans-atout*) et le nombre de levées* (plis) minimal que la paire s'engage à réaliser. Cette phase nécessite un investissement conséquent de la part d'un débutant pour apprendre

3. <http://www.decouvertedubridge.com>, https://en.wikipedia.org/wiki/Contract_bridge.

4. Département de la Fédération Française de Bridge (FFB) dirigé par Jean-Pierre Desmoulines et dédié à la jeunesse, la pédagogie et la recherche.

la signification des enchères par rapport à un système donné et surtout pour savoir l'utiliser à bon escient. Les enchères peuvent être vues comme un langage codé qu'on utilise pour faire passer des informations au partenaire sur sa main (principalement la distribution* et le nombre de points).

Plus précisément, le but de cette phase pour chaque camp est d'aboutir au meilleur contrat pour lui. Nous ne détaillons pas ici le calcul du score mais il faut savoir qu'il joue un rôle important dans le choix des enchères qu'un joueur peut faire. A partir d'un certain nombre de plis le camp obtient une prime si le contrat est gagné. Il existe des primes de *manche* liées aux contrats qui exigent un nombre minimum de levées dépendant de la couleur d'atout (9 plis à Sans Atout, 10 plis pour un contrat majeur* et 11 plis pour un contrat mineur*) et des primes *de chelems* sur les contrats où l'on s'engage à faire 12 plis ou tous les plis (*petit ou grand chelem*). Une décision fréquente pour un joueur concerne donc la hauteur du contrat : faut-il ou non demander un contrat en dessous du palier de la manche ou du chelem qu'on aura plus de chance de gagner puisqu'il exige un nombre de plis inférieurs mais qui rapportera moins ? Des études statistiques ont permis de déterminer des zones de points pour lesquelles il y a une probabilité assez grande de gagner les différents types de contrat. Pour prendre ces décisions, le joueur compte les points de sa main (cf annexe B) et essaie de transmettre cette information en utilisant des enchères. Les systèmes d'enchères qui varient d'une paire à l'autre tiennent compte de ces études et sont constituées de règles. Le score ne concerne que le nombre de plis réalisés et non le nombre de points dans les plis comme c'est le cas dans d'autres jeux de cartes. Les points associés aux cartes ne servent qu'à l'évaluation de la main et sont basés sur le fait qu'on a plus de chances de remporter un pli avec une carte forte qu'avec une petite carte.

Tâches à réaliser par un joueur humain

Un joueur doit à la fois produire une enchère parmi d'autres en fonction de sa main et de la séquence d'enchères courante et comprendre les enchères de son partenaire et de ses adversaires pour mettre à jour ses croyances. Un joueur humain va se trouver confronté à trois problèmes principaux :

1. La mémoire : le joueur doit se souvenir du système (il existe des systèmes plus ou moins complexes mais aucun n'est simple).
2. Le raisonnement : il doit raisonner afin de déduire des informations sur la main du partenaire mais aussi sur celles des adversaires puisque ceux-ci enchérissent aussi.
3. Le jugement : le joueur doit évaluer sa main en dehors des points et ce en fonction de toutes les informations dont il dispose. Ce jugement se modifie au cours de la séquence.

Pour un programme de bridge la partie mémoire ne posera pas de problème. Les parties raisonnement et jugement de main sont en revanche des challenges de taille pour les IA de bridge.

2.1.2. Le jeu de la carte

En bref : alors que le jeu de bridge est généralement considéré comme le plus complexe des jeux de cartes, ses règles sont relativement simples. Chaque joueur pose une carte à tour de rôle pour chaque pli. Pour cela un joueur doit fournir une carte de la couleur demandée quand il lui en reste et fournir n'importe quelle carte quand il n'a pas plus de carte dans cette couleur. Lorsque les quatre joueurs ont posé une carte, le pli se termine et est remporté par le joueur ayant posé la plus grosse carte dans la couleur demandée ou le plus gros atout. C'est lui qui posera la première carte du pli suivant. La donne se termine lorsque les cinquante-deux cartes ont été jouées. On compte le nombre de levées remportées par chaque camp (cette fois peu importe quel joueur de la paire a remporté le pli). Le score de la donne est calculé à partir de ce nombre et du nombre de levées requises par le contrat.

Après la phase des annonces*, on distingue le camp du déclarant (le joueur qui a fait la dernière enchère) et l'autre camp qu'on appelle « le camp de la défense » ou encore « le flanc ». Le défenseur à la gauche du déclarant va *entamer** c'est à dire choisir et poser une carte de son jeu sur la table. Après l'entame le jeu du partenaire du déclarant est étalé sur la table, il ne reste donc pour chaque joueur que 25 cartes inconnues à répartir entre les deux mains cachées. Chaque carte posée par la suite réduit encore l'incomplétude. En ce qui concerne le score, le but du déclarant est de réaliser au moins autant de plis que nécessite le contrat alors que le but de la défense va être de faire chuter le contrat en faisant le nombre de plis nécessaires pour cela. On pourrait croire que l'objectif pour les deux camps est de faire un maximum de levées mais ce n'est pas le cas et c'est une des caractéristiques qui rend ce jeu si complexe. Par exemple, un bon déclarant se contentera de faire 10 plis en toute sécurité si la manoeuvre pour en faire 11 risque de mettre en péril le contrat. De même le flanc se contentera d'une levée de chute s'il est probable de donner le contrat en tentant de faire chuter le déclarant de 2 ou plus. Il s'agit pour un expert de calculer des espérances de gain s'appuyant sur des probabilités (par exemple de répartition d'une couleur dans les mains cachées), mais aussi sur des facteurs psychologiques tels que la connaissance du niveau et du style de l'adversaire, la manière dont il a joué les premières cartes etc.

Paradoxe du bridge : il n'y a aucune contrainte d'ordre ou autre sur le choix de la carte. Un joueur n'est jamais obligé de mettre une carte plus forte ou de couper. C'est en partie cette simplicité apparente qui rend le jeu de la carte si complexe.

Tâches à réaliser par un joueur humain

Un joueur humain va se trouver confronté aux problèmes suivants :

1. Le choix de l'entame : ce choix est primordial car bien souvent il conditionne la réussite ou non du contrat.
2. La mémoire : il doit se souvenir des enchères, des conventions* du jeu de la carte et des cartes jouées.
3. Le raisonnement : le joueur doit raisonner afin de déduire des informations sur les deux mains cachées.

4. La révision d'hypothèses : il doit revoir ses hypothèses par exemple lorsque celles-ci deviennent trop peu probables ou impossibles en fonction des nouvelles cartes qu'il a vues.

5. Le calcul de probabilité des lignes de jeu envisageables.

Pour résumer on peut dire que pour les deux étapes, une grande partie de la tactique de jeu consiste à reconstruire l'information manquante à partir du contexte.

Avant de rentrer dans les détails liés aux caractéristiques du bridge par rapport aux notions de théorie des jeux, nous insistons sur le fait que le bridge est avant tout un jeu de raisonnement dans lequel l'apprentissage est central et permanent. Chaque tournoi est un enseignement y compris pour des champions et il n'existe aucun championnat où un joueur n'a fait aucune faute.

3. Bridge informatisé

Nous présentons un bilan des résultats obtenus par la communauté *computer bridge* qui s'intéresse depuis une quarantaine d'années à ce sujet en commençant par décrire le bridge d'un point de vue théorie des jeux. Nous faisons un historique des programmes de bridge et donnons les principes généraux des programmes actuels. Enfin, nous décrivons quelques pistes de réflexion sur les différents challenges inhérents aux IA de bridge.

3.1. Bridge et théorie des jeux

Les jeux sont classés en catégories qui se distinguent relativement à des caractéristiques ayant une influence sur la manière de traiter le jeu. Nous donnons les caractéristiques du bridge relativement à ces notions classiques, ces propriétés sont valables aussi bien pour la phase d'enchères que celle du jeu de la carte.

Le bridge est un jeu :

– **A information incomplète** : les joueurs ne voient qu'une partie des cartes, ils ont donc une connaissance incomplète de l'état. Ils ne connaissent pas non plus les gains résultant de leurs actions. Cependant chaque joueur connaît ses possibilités d'action, les possibilités d'action des autres joueurs et leurs motivations.

– **A mémoire parfaite** : ce n'est pas le cas dans un cadre réel mais on va supposer que chaque joueur peut se rappeler à tout moment la suite de coups qui ont été joués précédemment.

– **Non coopératif** : les joueurs ne peuvent se concerter pour définir des stratégies.

– **Séquentiel** : pour les 2 phases, l'ordre des actions des joueurs est déterminé (sens des aiguilles d'une montre).

– **Fini** : l'ensemble des stratégies de chaque joueur est fini.

– **A somme constante** : pas exactement à somme zéro mais à somme constante puisque les deux camps se battent pour treize plis.

– **Répété** dans le cadre d'un ensemble de donnes. La répétition permet de modifier le comportement d'une paire face à une autre. Un joueur peut modifier ses enchères et son jeu de la carte en fonction du niveau estimé des adversaires mais aussi en fonction de son partenaire. Si le joueur constate au fil des donnes que son partenaire est optimiste, il va être plus prudent pour éviter de jouer des contrats trop élevés. On connaît toujours le nombre de parties qu'on va jouer mais suivant les cadres de jeu on peut jouer contre les mêmes paires ou non. Un joueur peut aussi modifier son comportement s'il évalue qu'il a beaucoup d'avance ou de retard dans le match.

3.2. *Historique et description des programmes de bridge*

Il existe deux grandes catégories de programme de bridge. Les premiers appelés *superviseurs* fournissent un ensemble d'outils permettant aux joueurs humains de jouer entre eux via internet. Les superviseurs peuvent être utilisés dans un cadre expérimental de recherche. La seconde catégorie regroupe des *logiciels de jeu* permettant de progresser en jouant seul. Dans certains cas, le joueur dispose d'un nombre limité de donnes choisies par un expert pour leur intérêt pédagogique. Ces donnes pourront être rejouées interactivement et éventuellement accompagnées de commentaires du professeur. D'autres logiciels permettent de jouer un nombre illimité de donnes avec ou contre des robots de bridge. Il n'y a pas de commentaires permettant de faire progresser l'utilisateur les donnes étant choisies aléatoirement. Le Graal pour les logiciels de jeu serait de permettre de jouer un nombre illimité de donnes avec des commentaires associés.

Dans la suite, nous nous intéressons uniquement aux robots de bridge (voir aussi (Paul, 2010) pour un état de l'art sur le bridge informatisé).

Depuis 1996, les meilleurs programmes du monde se rencontrent tous les ans dans le cadre des championnats du monde de bridge des robots (WBBC). Cette compétition est organisée par la Fédération Mondiale de Bridge (WBF, Président Gianarrigo Rona) et la Fédération américaine (ACBL), sous la direction d'Alvin Levy (<https://bridge-robots.com/>).

Avant l'existence de ces championnats et dès les années 1960 sont apparus des algorithmes permettant de résoudre des problèmes de bridge extrêmement limités par la faible puissance de calcul (Berlekamp, 1963 ; Wasserman, 1970). BridgeBaron, paru en 1983, fut le premier logiciel permettant de jouer des donnes entières mais à un niveau encore très faible.

Face à un problème de jeu de la carte, le joueur humain construit un plan de jeu éventuellement divisé en étapes mettant en œuvre des techniques de bridge répertoriées : impasse*, affranchissement d'une couleur*, etc. (Smith *et al.*, 1996) ont utilisé avec un certain succès des techniques de planification (Hierarchical task network)

ayant pour but de déterminer les techniques à mettre en place. Leur approche a permis à BridgeBaron de remporter les WCBC en 1997, tout en jouant à un niveau sensiblement plus faible qu'un joueur moyen de bridge amateur.

Dans le cadre des programmes actuels, la technique la plus connue pour le jeu de la carte est appelée *double dummy* ou *jeu de la carte à 4 jeux*. Elle consiste à définir un solveur maximisant le nombre de plis pour chaque camp sur une version simplifiée du bridge. Un solveur de ce type est la base de tout programme de bridge puisqu'il est utilisé comme fonction d'évaluation.

Dans cette version simplifiée le contrat et les 4 mains sont connus et on ignore volontairement les enchères. Nous sommes alors dans un cadre information complète ce qui rend la version *double dummy* déterministe. Un arbre de recherche peut donc être construit sur ce jeu simplifié et il devient possible d'utiliser les techniques d'élagage classiques telles que l'algorithme Alpha-beta (α - β). Les valeurs des feuilles sont obtenues en utilisant le résultat du solveur à 4 jeux. Comme les plis remportés le sont définitivement, le nombre de plis courant fournit des bornes inférieure et supérieure pour l'élagage α - β avec des intervalles de plus en plus petits. Malgré cela, le facteur de branchement est encore trop grand pour permettre une résolution des problèmes de carte par la force brute. Les programmes de bridge ont commencé à utiliser une méthode comprenant deux phases : réduction de l'espace d'états en utilisant les symétries du jeu de bridge (typiquement en mettant des 'petites' cartes⁵ de la même couleur dans la même classe d'équivalence) puis résolution à quatre jeux d'un échantillon de données générées par une méthode de Monte-Carlo en cohérence avec les enchères produites et les cartes jouées. Le facteur de branchement est substantiellement réduit, passant de b à $b \times 0,76$. Cette méthode connue sous le nom de *partition search* a été formalisée dans (Ginsberg, 1996) et décrite de manière plus approfondie dans (Ginsberg, 2001). Le logiciel GIB (*Ginsberg's Intelligent Bridge player*) créé par Ginsberg a remporté les WCBC en 1998 et 1999.

Sur internet, le plus connu des jeux en ligne s'appelle *Bridge Base Online* (BBO représenté figure 1). La société Bridge Base Inc le développant a été fondée en 1990 par Frederick « Fred » Gitelman. BBO est gratuit et a la particularité de fonctionner comme un superviseur mais également de permettre aux joueurs d'utiliser des robots de type GIB.

3.3. Monte-Carlo dans le bridge

Le principe de la recherche Monte-Carlo dans le cadre des jeux est de faire des statistiques sur les actions possibles à partir de parties jouées aléatoirement. Dans les jeux à information incomplète l'approche Monte Carlo semble la meilleure puisqu'elle permet de générer aléatoirement la complétion de l'information cachée. L'arbre de recherche est l'outil incontournable de tout jeu à information complète, couplé à des

5. La notion de petite carte est floue, selon les cas elle peut concerner les cartes du 2 au 10, ou du 2 au 5.



Figure 1. Retransmission BBO. En haut à droite : la séquence d'enchères, en bas à gauche : les scores en IMP de chacune des équipes

fonctions d'évaluation et à des stratégies locales astucieuses. Toutefois, pour des jeux comme le go ayant un important facteur de branchement et où la fonction d'évaluation n'est pas aisée à construire, des simulations Monte-Carlo permettent de générer des échantillons réduits afin de prendre une décision d'action. Ces simulations sont également utiles au bridge dans deux situations : les décisions finales aux enchères et le jeu de la carte.

Enchères : les premières enchères sont souvent déterminées par un système de règles couplé à un recensement de situations classiques. En revanche, une telle architecture ne permet pas de couvrir toutes les séquences d'enchères (de l'ordre de 10^{47} séquences d'enchères possibles au bridge) et, quand on approche de la décision du contrat final, une simulation est utile afin de générer des mains (celle du partenaire et celles des adversaires) cohérentes avec les enchères précédentes ; l'enchère gagnante dans le plus grand nombre de cas est ainsi sélectionnée.

Jeu de la carte : après l'entame le déclarant voit sa main et celle de son partenaire ; il doit imaginer, en fonction des enchères précédentes et de l'entame les jeux de ses adversaires et prendre des décisions successives afin de mener à bien son contrat. Les logiciels de bridge simulent des échantillons de mains pour les jeux adverses et sélectionnent l'action gagnante dans le plus grand nombre de cas. Cette approche a des résultats généralement satisfaisants mais présente de nombreuses lacunes : si la simulation prend en compte les enchères (ou absence d'enchères) des adversaires, elle ne prend pas (encore) en compte les actions initiales des adversaires au jeu de la carte. Là où un joueur humain raisonnera que si son adversaire joue de telle façon il a tel ou tel jeu, les logiciels de bridge sont actuellement incapables de mener ce genre d'analyse.

3.4. Fonctionnement des IA de bridge actuelle

La plupart des IA de bridge disposent des modules suivants :

- M1 : Générateurs de donnes sous contraintes liées aux caractéristiques des mains. De nombreux programmes de type M1 ont été réalisés dans les années 1990.
- M2 : Algorithme efficace de résolution à 4 jeux qui permet d’avoir un calcul automatique du *Par d’une donne** à partir des 4 mains.
- M3 : Module de production des enchères.
- M4 : Module inverse permettant de traduire les enchères en contraintes.
- M5 : Evaluation de l’espérance de gain en IMP.
- M6 : Choix de l’entame.

Le module M2 est un solveur à quatre jeux qui est utilisé comme fonction d’évaluation aussi bien pour le jeu de la cartes que pendant les enchères. Généralement, les concepteurs de programmes de bridge définissent leur propre solveur, mais on peut utiliser celui développé par Bo Haglund connu comme le plus rapide et le plus utilisé (<http://privat.bahnhof.se/wb758135/bridge/index.html>). En combinant M1 et M2 on obtient un module de test qui permet de donner une approximation sur le pourcentage de réussite d’une ligne de jeu en supposant que tout le monde joue à cartes ouvertes et sur le calcul de l’espérance de gain approximative d’un contrat donné.

Pour le module M3, les programmes disposent d’un ensemble de règles qui permettent de choisir une enchère à partir du contexte. Par exemple, on choisira l’enchère de 1SA pour les mains sans singleton* de 15 à 17 points d’honneur *. La simulation intervient le plus tard possible idéalement pour déterminer le contrat final. Elle peut aussi intervenir lorsqu’on rencontre des situations non prévues (séquence d’enchères non répertoriée).

Le module M4 est utilisé pour ne pas générer de main incompatible avec les enchères précédentes. Le module M5 utilise les sorties des modules précédents pour évaluer les différentes options. Enfin, le module M6 concerne l’entame qui est un compartiment du jeu à part où les programmes utilisent des stratégies différentes. En voici quelques-unes :

- choisir la carte naturelle pour chaque couleur et faire ensuite des simulations sur ces 4 choix.
- choisir une règle parmi un ensemble de règles en tenant compte du contexte. Si le choix n’est pas unique sélectionner la meilleure entame en faisant des simulations.
- préférer la couleur la plus longue.

Certains choix de concepteurs sont liés à d’autres critères que l’optimalité immédiate. En effet, ils peuvent écarter une entame un peu meilleure statistiquement mais très étrange au profit d’une autre plus classique qui sera plus appréciée par le joueur humain partenaire ou adversaire du robot. Les concepteurs ne dévoilent pas facilement leur code et il n’y a pas beaucoup de documentations sur le fonctionnement des programmes. La communauté constate depuis quelques années que les progrès des robots

sont très marginaux. Ceux-ci progressent toujours, mais en couvrant des situations particulières qu'ils géraient mal auparavant et qui sont codées en dur dans le programme.

4. Challenges pour les programmes de bridge

La principale tâche consiste à compléter l'information manquante à partir du contexte. Le contexte varie suivant les étapes. Pour la partie enchères, il s'agit de la séquence d'enchères courante. Pour la partie jeu de la carte le contexte comprend la séquence d'enchères mais aussi les cartes déjà jouées. Sur la partie jeu de la carte, on peut assez facilement évaluer chaque position et le résultat final. Pour la partie enchères, l'évaluation d'une séquence d'enchères en cours n'a pas de sens. Par exemple, il existe des enchères artificielles qui ne traduisent pas une intention de jouer le contrat relatif à l'enchère. On peut par contre évaluer le résultat c'est à dire évaluer le contrat final auquel on aboutit.

Challenges enchères

1. Apprendre automatiquement une nouvelle manière de compter les points permettant de prendre des décisions avec un système de règles associé.

2. Apprendre à réagir en face d'un partenaire donné et mettre à jour automatiquement la base de règles en fonction du partenaire. Ce challenge est intéressant pour les robots jouant en face d'humains car ceux-ci se lassent vite de l'incompréhension de leur partenaire robot.

3. Revoir la fonction d'évaluation liée à la méthode à 4 jeux qui n'est pas satisfaisante. En effet, celle-ci va compter comme gagnant un contrat s'il existe au moins une ligne de jeu gagnante avec la meilleure défense possible. Une donne sur laquelle le contrat ne gagne qu'en utilisant une ligne de jeu très peu probable que l'on peut qualifier de mal jouée sera comptée comme positive alors qu'il est certain que tous les experts ne réussiront pas leur contrat puisqu'ils choisiront une ligne perdante mais plus prometteuse d'après les probabilités. On peut envisager d'associer des poids aux lignes de jeu en fonction de ces critères probabilistes pour pondérer le résultat de la fonction d'évaluation mais les calculs sont difficiles et coûteux.

4. Revoir le jugement de la main en fonction des enchères : il faudrait envisager une abstraction des mains de plus en plus fine au fur et mesure que la séquence se précise. Dans le cadre d'une approche symbolique utilisant un langage compréhensible par l'humain, on pourra utiliser des termes du langage de plus en plus précis. Par exemple, au début on peut considérer uniquement les valeurs et couleurs des 13 cartes puis on effectue des saturations relatives par exemple à la distribution, au nombre de cartes dans la couleur principale du partenaire etc. Les opérations sont plus coûteuses sur un langage plus riche mais elles se feraient lorsque l'incertitude est plus faible.

5. Apprendre automatiquement un système d'enchères : définir un système d'enchères optimal est un problème ouvert sur lequel se penchent des théoriciens du bridge depuis de nombreuses années. Des travaux utilisant des réseaux neuronaux à partir des règles du jeu de bridge n'ont rien donné de concluant.

6. Mettre des degrés d'agressivité différents en fonction des situations (partenaire, adversaire, retard dans un match). On peut affiner en mettant des degrés différents pour un même joueur sur un même match (ex : agressif en intervention, mais pas en réponse à l'intervention etc).

Challenges jeu de la carte

1. Evaluation d'une ligne de jeu* : on peut dire d'un bridgeur qu'il a très bien joué une donne* alors qu'il a fait deux plis de moins que tous les autres si le raisonnement global qu'il dit avoir suivi semble meilleur que celui des autres joueurs. L'évaluation d'une ligne de jeu* est basée sur des calculs de probabilités assez simples. Par exemple, on dit qu'une ligne de jeu est à 50 % si le succès de la stratégie mise en place par le joueur dépend du placement d'une carte dans un des deux jeux non visibles. Malgré l'apparente simplicité des calculs, il est beaucoup plus complexe d'évaluer une ligne de jeu car il faut utiliser toutes les informations dont on dispose telles que les informations liées aux enchères couplées au compte des points et au compte de distribution. Par exemple si la carte en question est dans une couleur annoncée comme courte chez un adversaire elle a moins de chances de s'y trouver.

2. Utiliser les informations liées à l'entame : au cours du jeu, les joueurs peuvent fournir des cartes trompeuses. C'est la partie « bluff » du bridge, la différence avec le poker est qu'on a un partenaire qui pourra être trompé aussi et mal raisonner ensuite. La carte d'entame est souvent considérée comme sûre.

3. Utiliser les informations liées à la manière de jouer des adversaires : pour cela il faut connaître le niveau des joueurs, on peut envisager aussi d'apprendre à évaluer le niveau ou le style d'un joueur.

4. Améliorer les techniques de flanc des robots : il y a beaucoup à faire sur cette partie qui nécessite une collaboration entre les défenseurs. La complexité provient également des conventions utilisées pour les cartes jouées.

5. Entame : comme nous l'avons vu précédemment l'entame est souvent cruciale. Il est envisageable de se concentrer uniquement sur cette partie du jeu en donnant en entrée d'un réseau profond des données (la séquence d'enchères, la main de l'entameur, l'entame). Le but du réseau sera de prédire la carte entamée parmi les treize.

6. Simulation : les programmes utilisent une approche à 2 jeux donc plus fine que le double dummy quand il reste moins de cartes. Certains programmes décident de le faire toujours au niveau du même pli (par exemple le quatrième), d'autres distinguent les cas suivant la hauteur du contrat (ex : quatrième pour une manche, troisième pour un chelem). On peut automatiser le procédé en repérant automatiquement les endroits où une analyse plus fine est nécessaire.

Challenge commun aux deux phases : reconnaissance de patterns

Il existe 5.36×10^{28} donnes différentes, et donc peu de chances de jouer une même donne. Cependant, on retrouve des situations similaires et la notion de patterns est très importante chez un bon joueur. L'expérience permet de repérer ces patterns pour les différentes phases du jeu. Des ouvrages de bridge proposent de résoudre des pro-

blèmes sur des donnes qui nécessitent la mise en place d'une certaine technique de jeu. Le lecteur pourra ensuite gérer la donne mais aussi tout un ensemble de donnes du même type. Le but est de reconnaître le pattern pour appliquer la technique associée. Pour un déclarant, il existe un ensemble de techniques connues pour maximiser le nombre de plis qu'il peut faire comme : la double coupe, le squeeze, l'affranchissement d'un couleur. Ici encore il faut pouvoir reconnaître les donnes pour lesquelles ces techniques s'appliquent soit pour les appliquer lorsque l'on est déclarant soit pour empêcher le déclarant de les mettre en place quand on est en flanc. La reconnaissance est plus difficile pour le flanc qui ne voit qu'un des deux mains du camp en attaque. Un bon défenseur se sert des enchères pour imaginer la main cachée et deviner quelles techniques le déclarant va essayer de mettre en place. Il se sert également des informations liées aux cartes déjà jouées.

Un challenge important pour le bridge est de pouvoir retrouver ces patterns pour pouvoir mettre en place des stratégies prévues par le système ou apprises.

Dans la suite de l'article, nous présentons nos travaux ayant permis de booster une IA existante.

5. Boosting d'une IA de bridge

Les approches proposées dans (St-Pierre, Teytaud, 2014; Cazenave *et al.*, 2015; Liu *et al.*, 2016) ont montré que le choix de la graine aléatoire pouvait avoir un impact non négligeable sur les performances d'une l'IA utilisant des simulations de Monte-Carlo. Pour cela, les auteurs ont défini et testé des algorithmes permettant de choisir de bonnes graines ou de générer de bonnes distributions de probabilité sur les graines. Dans certains cas l'optimisation est quasi nulle, dans d'autres elle est très significative. Le but de nos travaux est d'adapter la méthode de boosting à une IA de bridge et d'étudier l'impact du choix de la graine aléatoire sur la performance de l'IA.

5.1. Approche de Boosting IA de jeux

Un programme stochastique utilise une graine aléatoire qui détermine la séquence de nombres générés à chaque appel de la fonction random dans la suite du programme (on parle de séquence pseudo-aléatoire). Cette graine peut être générée aléatoirement par exemple en utilisant l'horloge ou fixée de manière arbitraire par le concepteur.

Le principe général de l'approche présentée dans (St-Pierre, Teytaud, 2014) est de construire des IA déterministes à partir d'une IA stochastique et de comparer les variations des résultats en faisant jouer les IA déterministes les unes contre les autres.

L'idée est qu'un programme stochastique est en fait une variable aléatoire distribuée sur un programme déterministe. Soit $IA(w)$ le programme stochastique, $IA(1)$ correspond au programme déterministe de la même IA utilisant la graine 1.

Deux algorithmes de simulation pour des jeux à deux joueurs ont été définis : *BestSeed* qui permet de trouver la meilleure graine et *Nash* qui renvoie une distribution

de probabilité sur les graines. Une extension de la méthode est donnée dans (Liu *et al.*, 2016) et testée sur trois nouveaux jeux Atari-Go, Domineering un jeu combinatoire à base de dominos de différentes tailles, et Breakthrough une variante des échecs.

Le bridge n'est pas dans ce cadre d'information complète mais l'approche a été récemment testée avec succès sur un jeu partiellement observable. En effet, dans (Cazenave *et al.*, 2016) et (Pepels *et al.*, 2015) les auteurs ont réalisé une expérimentation sur une IA de Phantom Go⁶. Les gains sont les suivants : sur un plateau de taille 5X5 le taux de victoires est passé de 50 % à 70 % contre la même IA non boostée et de 0 % à 40 % contre une IA plus forte sur Phantom Go (5X5, 7X7 et 9X9).

5.2. Adaptation au bridge

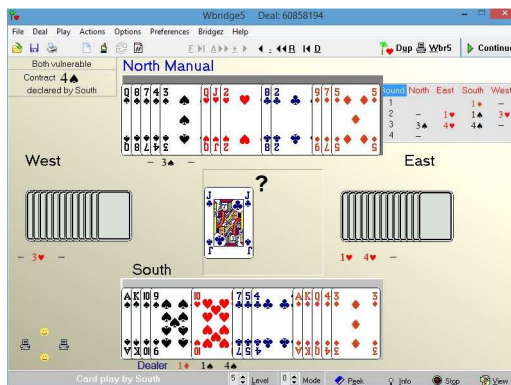


Figure 2. Copie d'écran de Wbridge5

Le but est de savoir si la méthode d'optimisation de la graine peut être efficace sur les IA de bridge et si oui sur quelles parties du raisonnement. L'IA que nous avons sélectionnée est celle développée par Yves Costel pour Wbridge5 (figure 2). Wbridge5 répond aux conditions pour que la méthode puisse être appliquée. Le programme est stochastique et utilise un générateur de nombres pseudo-aléatoires pour obtenir un échantillon des mains cachées. Le programme permet de fixer la graine du générateur ce qui rend le programme déterministe. Nous pouvons donc construire une matrice en faisant jouer les IA déterministes les unes contre les autres.

La première difficulté est de gérer le fait qu'on l'on ait quatre IA et non pas deux comme dans les jeux traités jusqu'ici. Nous devons donc fixer quatre graines. Pour rendre plus simple l'interprétation des résultats, nous avons choisi de fixer une graine pour chaque camp et non pas une pour chaque joueur. Par ailleurs, il n'y a pas de raison a priori pour que la graine soit la meilleure pour les deux étapes de jeu pour

6. Phantom Go est un jeu à deux joueurs à information incomplète qui consiste à jouer au go sans voir les mouvements de l'adversaire.

lesquelles les simulations ne sont pas utilisées de la même manière. Nous avons choisi de commencer par une optimisation de la graine utilisée dans la phase des enchères, le contrat final étant crucial car même si le jeu de la carte est parfait, on obtiendra un mauvais score si le contrat joué n'est pas le bon. Il y a plus de possibilités dans la partie enchères car 3 mains soit 39 cartes sont inconnues tandis qu'il y a seulement 25 cartes inconnues au maximum pendant le jeu de la carte après l'entame. En outre, le jeu de la carte bénéficie des renseignements déduits des enchères et des cartes déjà jouées. Il y a donc beaucoup moins d'inconnues. Les derniers plis sont pratiquement joués à cartes ouvertes. Le temps de calcul consacré au jeu de la carte est également beaucoup plus long que celui consacré aux enchères.

5.3. Les scores au bridge

Les résultats possibles des jeux traités dans (Cazenave *et al.*, 2015 ; Liu *et al.*, 2016) comme Domineering, Atari-Go, ou Phantom-Go sont 0 pour une défaite, 0.5 en cas d'égalité et 1 pour une victoire. Le bridge nécessite des résultats plus précis. En effet, il est important de connaître la différence exacte de points entre le vainqueur et le perdant. Le score d'une donne est relatif au contrat atteint à l'issue des enchères et au nombre de plis gagnés pendant le jeu de la carte.



Figure 3. Boîte à enchères utilisée pendant les compétitions de bridge

Scoring d'une donne

Soit n un nombre entier compris entre 1 et 7 et $S \in (\spadesuit, \heartsuit, \diamondsuit, \clubsuit, SA)$. Le contrat nS^7 est un contrat où le nombre minimum de plis à réaliser par le déclarant est égal à $(n + 6)$ avec S représentant la couleur d'atout s'il y en a une ou le fait qu'il n'y ait pas d'atout si $S = SA$ (Sans Atout). Par exemple, $4\heartsuit$ sera gagné par le camp en attaque si le déclarant a remporté au moins 10 plis en jouant à l'atout \heartsuit . Si le déclarant réalise 10 plis le camp en attaque obtient +420 (ou +620 suivant la marque), si le déclarant a remporté moins de 10 plis l'autre camp marque +50 (ou +100) par levée manquante.

Nous nous focalisons sur la marque duplicata qui est utilisée pour réduire de manière significative le facteur chance. Le principe est le suivant : dans un match s'af-

7. Les différents contrats possibles sont représentés sur la figure 3.

frontent deux équipes A et B composées de deux paires chacune. Les mêmes donnes sont jouées à deux tables différentes, les paires de la même équipe étant installées dans des directions opposées. Les joueurs sont représentés par les points cardinaux Nord, Est, Sud et Ouest (abrégés en N, E, S, O) les paires pouvant être soit NS soit EO. Pour chaque donne du match, la paire NS (resp. EO) de l'équipe A rencontrera la paire EO (resp. NS) de l'équipe A et aura exactement les mêmes mains que la paire NS (resp. EO) de l'équipe B à l'autre table.

Le score final pour une donne consiste à calculer la différence des scores aux deux tables. Par exemple, si l'équipe A a marqué + 420 et l'équipe B - 50, la différence est de 470 en faveur de A. Cette différence est ensuite convertie en *International Match Point scale* (IMP). La conversion non linéaire des points en IMP est représentée figure 4. Une équipe peut obtenir entre -24 + 24 IMP par donne. Le score d'un match est la somme des scores en IMP de chaque donne.

Diff. in Pts.	IMPs	Diff. in Pts.	IMPs	Diff. in Pts.	IMPs	Diff. in Pts.	IMPs
20 - 40	1	270 - 310	7	750 - 890	13	2000 - 2240	19
50 - 80	2	320 - 360	8	900 - 1090	14	2250 - 2490	20
90 - 120	3	370 - 420	9	1100 - 1290	15	2500 - 2990	21
130 - 160	4	430 - 490	10	1300 - 1490	16	3000 - 3490	22
170 - 210	5	500 - 590	11	1500 - 1740	17	3500 - 3990	23
220 - 260	6	600 - 740	12	1750 - 1990	18	4000 and up	24

Figure 4. Table de conversion des points en IMP

Pour plus de détails sur la marque :

https://fr.wikipedia.org/wiki/Marque_du_bridge

5.4. Expérimentations : choix de la meilleure graine

Les expériences ont été réalisées par Yves Costel en utilisant le moteur de bridge de Wbridge5 sur un Dell Ultra book INtel 2.4 Ghz. Nous nous plaçons dans le cadre d'un match par 4 qui oppose deux équipes de 4 joueurs. Le match par quatre est la compétition reine du bridge et c'est avec ce format que les championnats des programmes sont joués. Nous utilisons la marque IMP décrite dans la section précédente.

Avant de construire une matrice complète nous avons étudié l'éventuel impact de la graine sur un match de 64 donnes (c'est le nombre utilisé pour les matches des World Computer-Bridge Championship). Le résultat de chaque donne est calculé à partir du Par de la donne c'est à dire à partir du résultat théorique du jeu à cartes ouvertes et non pas en faisant jouer les 8 robots. L'expérience concerne donc un choix de graine pour les enchères, la graine choisie ne participant pas au processus du jeu de la carte. Nous avons constaté sur ces donnes que la graine n'avait une influence que dans environ 10 % des cas mais que le gain pouvait être important car ces cas concernaient des

décisions tendues (du type « faut-il demander la manche ou le chelem ? ») qui peuvent rapporter ou coûter une dizaine d'IMP sur une seule donne.

Ce résultat étant concluant, nous avons entrepris de construire une matrice de taille 40×40 où chaque cellule $R[i,j]$ contiendra le résultat en IMP d'un match de 64 donnes entre deux instances du programme déterminisées avec la graine i et deux instances déterminisées avec la graine j .

REMARQUES.

1. Durant les expérimentations, nous avons observé qu'il n'y avait aucune simulation pour 10 % des donnes, et que dans 75 % des cas le contrat atteint était le même pour les deux graines.
2. La graine n'a un impact que dans 15 % des cas mais ces cas concernent des prises de décisions importantes avec beaucoup d'IMP en jeu (ces donnes sont appelées donnes à swing).
3. Moins de la moitié des termes est calculée puisque $R[j,i] = -R[i,j]$ et la diagonale est constituée de zéro car elle représente les matches opposant les 8 mêmes IA.

Le vecteur suivant représente les scores cumulés (somme de toutes les valeurs par colonne) par graine testée :

(156 -28 55 167 -170 -188 -16 246 -30 94 -111 -113 266 -73 13 110 54 63 86 21 -53 -263 30 130 -62 -162 -57 -20 -173 -5 -81 57 92 -151 -25 -44 118 -234 269 32).

Plus le résultat est grand, meilleure est la graine. Sur ces expériences, la plus mauvaise graine est la graine 22 (- 263) et la meilleure la graine 39 (+269).

Nash renvoie une distribution de probabilité sur les graines ce qui permet de faire en sorte que l'IA ne soit pas prévisible pour l'adversaire. Ce n'est pas le problème pour le bridge en raison de sa complexité. Nous avons donc utilisé et testé l'approche BestSeed qui est directement adaptable à des jeux de plus de 2 joueurs et nous avons sélectionné la meilleure graine relativement à la somme de toutes les valeurs par ligne de la matrice. La meilleure graine (39 sur notre expérience) est celle dont la somme sur la ligne est la plus grande.

5.5. Validation croisée

Le protocole suivi pour la validation est celui préconisé dans : <https://www.lri.fr/teytaud/loo.pdf>. L'idée générale de cette validation croisée est d'étudier l'impact de la meilleure graine en gardant une dizaine de points sur les quarante pour former l'ensemble test (pour $K=1, 2, 3, \dots, 30$) dans l'algorithme Evaluation.

Algorithme Evaluation

```

Pseudo-code :
nbArms = param
M= matrice des r\resultats
pour K=1, 2, 3, ..., 30
{
  pour i=1, 2, ..., n      (n = minimum 500)
  {
    M=M(randompermutation(40), randompermutation(40))
    submatrix=M( 1:K, 1:K )
    testmatrix=M( 1:K, (K+1):40 )
    score(1) = somme de la ligne 1 de M
    score(2) = somme de la ligne 2 de M
    ...
    score(K) = somme de la ligne K de M
    soit I=(i1, ..., inbArms)
    la liste des indices avec le meilleur score
    resultats(i) = moyenne des lignes dans testmatrix
  }
  performance(K)=moyenne des resultats(i)
}

```

Les courbes ci-dessous représentent le résultat de l'implémentation en Python de l'algorithme Evaluation. Dans un premier temps, nous avons étudié la performance de la meilleure graine sur une nouvelle matrice où toutes les valeurs positives de la matrice initiale ont été remplacées par la valeur 1 représentant une victoire et toutes les valeurs négatives de la matrice par la valeur -1 représentant une défaite.

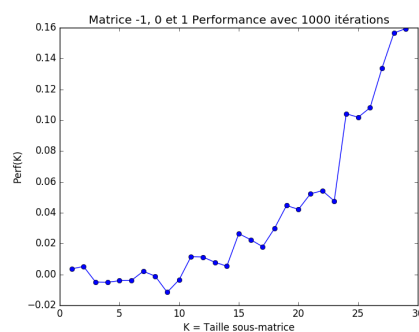


Figure 5. Courbe de performance cadre -1,0 et 1 avec 1000 itérations

La figure 5 représente l'évolution de la performance de la meilleure graine dans le cadre -1, 0 et 1. La performance augmente de manière régulière et significative.

La progression de la performance est moins stable dans le cadre IMP (figure 6) testé avec 3000 itérations. ce qui peut s'expliquer par le fait que la graine a une influence qui provoque de gros écarts mais sur peu de cas comme on l'a vu précédemment.

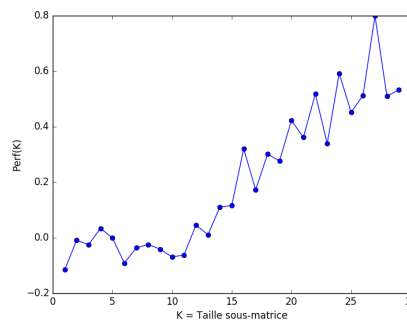


Figure 6. Courbe de performance cadre IMP avec 3000 itérations

Nous avons vérifié que l'écart type diminuait dans le cadre IMP, ce qui est le cas (cf figure 7).

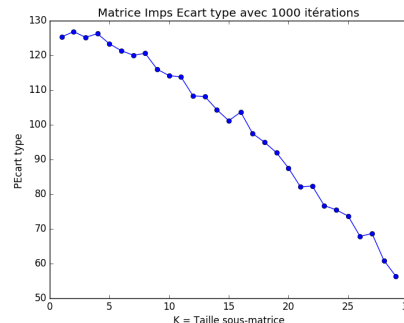


Figure 7. Ecart type cadre IMP avec 1000 itérations

5.6. Validation interne

Un match a été lancé dans les mêmes conditions mais cette fois sur 1000 donnes entre le programme avec la meilleure graine trouvée (39) et le programme avec la plus mauvaise (22). Le programme avec la meilleure graine a gagné avec un écart de $163 \pm 2,62$ IMP, sur 1000 donnes, i.e. 0,16 imp par donne. Un second match de 1000 donnes entre le programme avec la meilleure graine (39) et le programme avec la graine actuelle (99) qui avait été choisie arbitrairement a donné les résultats suivants. Le programme avec la meilleur graine a remporté le match avec un écart de $97 \pm$

2,75 IMP sur 1000 donnes, ce qui correspond à un gain de $\simeq 0,1$ imp par donne. Ces résultats confirment la validation croisée. Dans la matrice chaque graine est testée sur 2496 donnes (64*39), la meilleure graine gagnant 269 IMP et la plus mauvaise perdant 263 IMP soit légèrement plus de 0,1 imp par donne.

A l'issue de ces expérimentations, nous avons décidé de remplacer la graine courante par la meilleure graine pour la participation de Wbridge5 aux championnats du monde de computer bridge.

5.7. Championnats du monde septembre 2016

La vingtième édition des WCBC⁸ a eu lieu à Wroclaw (Pologne) du 5 au 10 septembre 2016 sous la direction d'Alvin Levy qui fournit les donnes, arbitre en cas de litige, et établit le classement. La partie technique est entièrement gérée par Gérard Joyez qui a développé *Table Manager* le programme central de ces championnats puisqu'il permet de gérer l'échange d'informations entre les robots et d'éviter tout système de triche. Voici les huit participants avec leur pays d'origine et leur palmarès : Bridge Baron (USA, 1997), Meadowlark Bridge (USA, 2000), Micro Bridge (Japon), Q-Plus Bridge (Allemagne), RoboBridge (Pays-Bas), Shark Bridge (Danemark, 2011 et 2014), Wbridge5 (France, 2005, 2007 et 2008) et Xinrui (Chine). On notera l'absence du champion du monde en titre Jack (Pays-Bas) et le retour de Meadowlark Bridge après des années d'absence ainsi que l'arrivée d'un nouveau compétiteur Yuzhang Liu dont le robot Xinrui est issu de d'une entreprise chinoise de 20 personnes dédiée au computer bridge et dont il est le fondateur.

Le format de la compétition consiste en une phase qualificative appelée Round-Robin au cours de laquelle tous les robots s'affrontent dans des matchs de 32 donnes. A l'issue de cette phase, les quatre premiers programmes sont qualifiés pour les demi-finales le premier rencontrant le quatrième et le second le troisième dans des matchs de 64 donnes. Les vainqueurs de ces demi-finales jouent ensuite la finale sur un match de 64 donnes. Résultats du Round Robin :

1. Wbridge5 91,87
2. Micro Bridge 90,07
3. Bridge Baron 89,21
4. Shark Bridge 80,12
5. Q-Plus Bridge 78,76
6. Xinrui 78,50
7. RoboBridge 50,58
8. Meadowlark Bridge 0,89

Wbridge5 a terminé en tête du Round-Robin avec un écart très faible sur le deuxième et le troisième. Meadowlark après 12 ans d'absence a connu beaucoup de problèmes

8. www.computerbridge.com pour plus d'informations.

tant techniques que bridgesques. En revanche, Xinrui le nouvel arrivant a bien figuré. Il était même en passe de se qualifier avant de subir une lourde défaite face à Shark Bridge dans son dernier match. En règle générale la compétition a été serrée entre les six premiers. Ainsi Q-plus bridge n'a perdu la quatrième place qualificative que lors de la dernière donne de son dernier match.

Les demi-finales et la finale ont été également très disputées. Wbridge5 a gagné sa demi-finale contre Shark 140,6-131. L'écart de 9,6 IMP correspond en partie au carry-over gagné grâce la victoire de Wbridge5 contre Shark pendant le Round-Robin. L'autre demi-finale a vu Micro-Bridge battre Bridge Baron par un écart encore plus faible : 144-138 alors qu'à la mi-temps du match, Bridge-Baron était en tête de 66 IMP (96 à 30).

La finale a été également très serrée. A deux donnes de la fin, Wbridge5 avait 17 IMP de retard sur Micro Bridge. Cependant, les deux dernières donnes lui ont permis de gagner 23 IMP pour un score final de 162 à 156. Sur ces deux donnes l'IA a fait la différence en trouvant un meilleur contrat que son adversaire. Comme on l'a vu dans le cadre de nos expériences il semble que la graine a un impact sur des décisions tendues comme celles-ci. La différence de 6 points pour 64 donnes correspond à un gain de 0,09375 par donne alors que le gain estimé de la meilleure graine par rapport à l'ancienne est de 0,1 IMP par donne.

Wbridge5 a ainsi gagné son quatrième titre de champion du monde huit années après sa dernière victoire.

6. Perspectives et travaux en cours

La génération d'enchères a été étudiée précédemment dans (Amit, Markovitch, 2006; DeLooze, Downey, 2007) avec l'élaboration d'un algorithme PIDM (Partial Information Decision Making) utilisé pour prédire des enchères qualifiées de raisonnables. Dans (DeLooze, Downey, 2007), un réseau de neurones auto-adaptatif a été utilisé pour enchérir des mains restreintes aux contrats sans atout. Une grande partie des recherches actuelles sur les jeux utilise les réseaux neuronaux profonds (par exemple, (Moravčík *et al.*, 2017) pour le poker), avec des succès nets comme celui de Libratus, intelligence artificielle développée par l'Université Carnegie Mellon, qui a battu quatre des meilleurs joueurs de poker Texas Hold'em sans limite sur 20 jours de compétition. Dans le cadre du bridge les réseaux profonds ont été utilisés dans (Yegnanarayana *et al.*, 1996) et plus récemment dans (Ho, Lin, 2015; Yeh, Lin, 2016). Dans (Yeh, Lin, 2016), un modèle d'apprentissage par renforcement profond a été conçu pour faire de la génération automatique d'enchères dans un cadre restreint aux séquences non compétitives. Cette restriction impose que les adversaires ne participent pas à la phase d'enchères. Notre approche n'est pas en concurrence avec ces méthodes, ce type de séquences ne se produisant que dans 27 % des cartes (statistiques obtenues à partir de données sur le site de la WBF). En outre, le sous-problème traité est lié à un processus à deux joueurs plutôt qu'à quatre joueurs.

Perspectives

L'expérience a été réalisée par Yves Costel et s'est déroulée sur un Dell Ultra book Inspiron Intel 2,4 Gz. On peut envisager de l'étendre en utilisant des machines plus puissantes. Dans l'approche que nous avons mise en place la graine est la même pour toutes les donnes et tous les appels de simulation. Pendant la phase d'enchères, les situations sont répertoriées et classées suivant le nombre de plis que l'on pense faire (zone de partielle, manche ou chelem). Les IA de bridge sont actuellement capables de savoir dans quelle situation elles se situent au moment des prises de décision. On pourrait améliorer l'efficacité de l'approche en recherchant une meilleure graine pour chaque situation spécifique. On pourrait aussi distinguer les situations en fonction du nombre de possibilités. Par exemple, lorsqu'il y a eu beaucoup d'enchères les mondes possibles sont moins nombreux puisque chaque enchère crée une contrainte supplémentaire. Il semble intéressant de construire des matrices avec quatre graines différentes et bien entendu d'étendre la recherche de la meilleure graine sur le jeu de la carte qui utilise aussi des simulations.

Avant de considérer d'autres méthodes pour le bridge il est important de déterminer les sous-problèmes du bridge pour lesquels une approche de type AlphaGo pourrait être efficace. Par exemple, il est intéressant de voir si les techniques récentes de gestion et d'analyse de gros volumes de données (*big data*) utilisées par AlphaGo peuvent être appliquées au bridge. Pour cela il faut pouvoir disposer d'un grand nombre de parties de qualité jouées par des humains. C'est le cas pour le bridge car il existe de nombreuses archives de matches joués dans le cadre de championnats nationaux ou internationaux de haut niveau. A la différence du Go par exemple où une partie ne peut être jouée plusieurs fois, ces données permettent de comparer ce qu'ont fait des joueurs différents sur exactement les mêmes donnes. Les formats de ces archives sont normalisés et facilement utilisables, la difficulté principale est d'y avoir accès et surtout de posséder l'expertise suffisante pour pouvoir les trier (par exemple suivant le système d'enchères* utilisé ou suivant la catégorie lié au niveau du contrat dans laquelle on peut les mettre), et pour pouvoir définir les bonnes caractéristiques pour modéliser les donnes. La première étape d'AlphaGo permettant au système d'atteindre un bon niveau en utilisant des parties jouées par des champions nous semble prometteuse pour le bridge car l'apprentissage humain utilise beaucoup l'imitation. D'autre part, il est très important pour la progression d'un joueur qu'il puisse s'entraîner avec et contre des joueurs ayant un niveau supérieur au sien. Cependant même en admettant que l'on puisse développer une IA de bridge exactement sur le modèle d'AlphaGo, elle ne sera pas complètement satisfaisante car au bridge, il faut non seulement effectuer les bonnes actions mais il faut également pouvoir expliquer pourquoi on les a faites. L'évaluation de la stratégie d'un joueur est indépendante du résultat obtenu. On peut dire qu'un joueur a très bien joué une donne tout en sachant que le score qu'il a obtenu sur cette donne est mauvais. AlphaGo n'utilise pas la force brute classique, mais il ne peut pas expliquer le raisonnement qu'il a suivi pour choisir telle ou telle action, l'approche adoptée reste quantitative et non qualitative. Le fait de coupler des approches numériques et symboliques pourrait permettre d'avoir une approche quali-

tative du bridge. On pourrait par exemple entraîner le robot à prendre des décisions en utilisant des approches réseaux profonds et RL et utiliser l'apprentissage supervisé symbolique pour mettre à jour automatiquement les bases de connaissances des robots. Ceci permettrait la transmission du mode de raisonnement et la caractérisation de l'adaptation du robot à un partenaire différent.

Travaux en cours

La communauté computer bridge n'a pas bénéficié jusqu'ici de grandes ressources alors qu'il faudrait disposer d'une grosse puissance de calcul pour garantir que l'évaluation Monte-Carlo donne des résultats suffisamment précis. On pourrait utiliser comme pour Mogo le parallélisme à grande échelle (avec notamment GRID'5000). Nous avons commencé à étendre nos expérimentations en utilisant une méthodologie de type bandits qui nous permettra de comparer plus de graines en ne construisant qu'une partie de la matrice.

Nous travaillons également sur la définition d'un MDP (*Markov Decision Process*) et d'une adaptation d'un MCTS (*Monte Carlo Tree Search*) pour le bridge. Celui-ci servira d'entrée pour une approche de type apprentissage par renforcement. Plus précisément, nous pensons que le bridge est une bonne application pour utiliser un cadre mixte d'apprentissage par renforcement dans un cadre symbolique comme celui défini dans (Nitti *et al.*, 2015).

Les prises de décisions au début d'une donne ne peuvent être traitées par une approche Monte Carlo car les possibilités sont trop nombreuses. Dans ce cadre, on peut envisager de faire un pré-traitement qui permettrait de simuler les décisions d'un expert ou de mettre à jour la base de règles d'un robot. Des travaux relatifs à ce pré-traitement sont en cours de réalisation en collaboration avec Swann Legras. Ces travaux s'inscrivent dans un cadre d'apprentissage supervisé et sont décrits brièvement ci-dessous.

Dans un cadre d'apprentissage supervisé, la base d'exemples utilisée en entrée est partitionnée en deux ensembles⁹ : E+ (les exemples positifs) et E- (les exemples négatifs). A partir de cette base, le but est soit dans un cadre numérique de simuler les décisions d'un expert en apprenant un estimateur qui renverra + ou - en imitant le comportement lié à la base, soit d'apprendre un ensemble de règles explicites permettant d'imiter l'expert tout en expliquant pourquoi on a choisi + ou -. Dans ce cadre symbolique, il sera possible de mettre à jour la base de règles d'un robot pour l'améliorer si on considère que c'est une bonne règle (l'expert est un champion incontesté), ou encore pour personnaliser la base relativement à un partenaire qui n'aurait pas les mêmes règles que l'IA ce qui peut être une avancée pour les robots jouant en face de différents partenaires humains. On peut aussi envisager d'apprendre plusieurs estimateurs (ensemble d'apprentissage avec des experts plus ou moins agressifs) que l'on

9. Il est possible d'avoir une partition de taille supérieure à deux dans le cas d'un apprentissage multi-concepts ce qui est intéressant si le choix concerne plusieurs actions possibles.

pourra utiliser conjointement ou en fonction des situations (« est-on en avance ou en retard dans le match ? », « les adversaires sont-ils agressifs ou non ? »).

Nous avons commencé à faire une étude sur l'ouverture* de mains limites par rapport à leur nombre de points. Ces mains nécessitent un bon jugement pour prendre une décision sur leur ouverture ou non. Pour obtenir une base d'exemples assez pure, nous avons fait étiqueter des donnes de ce type générées aléatoirement par des experts via l'interface représentée dans la figure 8 où l'expert clique sur Oui s'il ouvre la main et sur Non sinon. Si l'expert ouvre la main l'exemple est placé dans E+ sinon il est placé dans E-. Pour éviter cet étiquetage manuel, on pourrait récupérer des donnes jouées par un joueur expert, faire une sélection (ex : distribution, zone de points et début de séquence pour voir quelle enchère l'expert a produit) et lancer un apprentissage de ce type dessus. Dans ce cadre, nous pouvons aussi envisager de faire un « super robot de bridge » en apprenant sur les donnes jouées par un champion pour les enchères, et par un autre expert connu pour son talent sur un compartiment du jeu de la carte.



Figure 8. Interface étiquetage expert

Le traitement de cette base d'exemples est en cours à la fois pour l'approche numérique et pour l'approche symbolique.

7. Conclusion

Nous avons adapté une méthode de boosting d'IA de jeu existante aux IA de bridge et étudié son efficacité dans un cadre expérimental puis dans le cadre de championnats de robots de bridge. La méthode a donné de bons résultats, l'IA boostée avec la meilleure graine trouvée lors de nos expérimentations ayant remporté les derniers championnats du monde des robots de bridge. Nous avons également recensé un ensemble de défis permettant d'améliorer une IA de bridge en utilisant des techniques d'apprentissage récentes, les IA actuelles étant basées sur des travaux réalisés à la fin des années 1990. Parmi les techniques récentes envisagées, une approche de type réseaux profonds paraît prometteuse sur certains compartiments du bridge. Cependant, il faut être attentif au fait que le bridge a cette particularité d'être joué à quatre sous la forme de deux paires. Au go la première étape a consisté à faire s'entraîner le robot tout seul face à des experts puis face à des versions de lui-même. Au bridge, il faudrait envisager du co-entraînement plutôt qu'un simple entraînement.

L'aspect évolution dans le temps du bridge (l'incertitude liée à la connaissance de l'état réel du monde diminue au cours de la partie) laisse à penser que l'utilisation de réseaux neuronaux récurrents pourrait être efficace. Nous pensons également qu'une approche utilisant les réseaux neuronaux profonds pourrait améliorer le comportement des robots sur la partie entame qui est cruciale au jeu de la carte.

Il reste beaucoup de chemin à parcourir pour aboutir à la création d'une IA pouvant battre des champions de bridge. Il faudrait par exemple pouvoir mettre l'adversaire à la faute (ce qui n'est pas possible avec le double dummy), gérer la coopération en flanc (le point faible des robots au jeu de la carte), adapter sa stratégie au style de l'adversaire, produire des enchères d'obstruction ayant pour but de gêner l'adversaire plutôt que de donner des informations au partenaire.

Cet article est un premier pas vers la définition d'une architecture hybride composée de modules issus de travaux récents en apprentissage automatique numérique et symbolique. Pour la partie symbolique, des travaux récents sur l'apprentissage renforcé utilisant un formalisme logique permettraient à la fois de jouer à un niveau supérieur à celui des IA actuelles et d'expliquer de manière intelligible à un humain le raisonnement effectué par le programme. Dans tous les cas, concevoir un programme de bridge expert serait un pas vers l'intelligence générale artificielle ou IA forte en opposition à l'IA appliquée.

Remerciements

Nous tenons à remercier Yves Costel qui a réalisé les expérimentations et avec qui nous avons beaucoup de plaisir à collaborer. Un grand merci aussi aux bridgeurs Jean-Baptiste Fantun, Jean-Christophe Quantin et Jean-Pierre Desmoulins pour toutes les discussions constructives que nous avons eues sur ce sujet. Enfin, cet article est dédié à Daniel Kayser pionnier français de l'intelligence artificielle qui fut mon ami, mon directeur de thèse et dont l'esprit libre restera à jamais gravé dans nos mémoires.

Bibliographie

- Amit A., Markovitch S. (2006). Learning to bid in bridge. *Machine Learning*, vol. 63, n° 3, p. 287–327.
- Berlekamp E. R. (1963). Program for double-dummy bridge problems: a new strategy for mechanical game playing. *Journal of the ACM (JACM)*, vol. 10, n° 3, p. 357–364.
- Cazenave T., Liu J., Teytaud F., Teytaud O. (2016). Learning opening books in partially observable games: using random seeds in phantom go. *arXiv preprint arXiv:1607.02431*.
- Cazenave T., Liu J., Teytaud O. (2015, Aug). The rectangular seeds of domineering. In *2015 IEEE conference on computational intelligence and games (cig)*, p. 530-531.
- Coulom R. (2006). Efficient selectivity and backup operators in monte-carlo tree search. In *International conference on computers and games*, p. 72–83.
- DeLooze L. L., Downey J. (2007). Bridge bidding with imperfect information. In *Computational intelligence and games, 2007. cig 2007. IEEE symposium on*, p. 368–373.

- Gelly S., Kocsis L., Schoenauer M., Sebag M., Silver D., Szepesvári C. *et al.* (2012). The grand challenge of computer go: Monte carlo tree search and extensions. *Communications of the ACM*, vol. 55, n° 3, p. 106–113.
- Ginsberg M. (1996). Partition search. *Proceedings Of The National Conference On Artificial Intelligence*.
- Ginsberg M. (2001). Gib: Imperfect information in a computationally challenging game. *Journal of Artificial Intelligence Research*, vol. 14, p. 303–358.
- Heinrich J., Silver D. (2016). Deep reinforcement learning from self-play in imperfect-information games. *arXiv preprint arXiv:1603.01121*.
- Ho C.-Y., Lin H.-T. (2015). Contract bridge bidding by learning. In *Proceedings of the workshop on computer poker and imperfect information at the twenty-ninth aai conference on artificial intelligence*.
- Liu J., Teytaud O., Cazenave T. (2016). Fast seed-learning algorithms for games. In *International conference on computers and games*, p. 58–70.
- Metropolis N., Ulam S. (1949). The monte carlo method. *Journal of the American statistical association*, vol. 44, n° 247, p. 335–341.
- Moravčík M., Schmid M., Burch N., Lisý V., Morrill D., Bard N. *et al.* (2017). Deepstack: Expert-level artificial intelligence in no-limit poker. *arXiv preprint arXiv:1701.01724*.
- Munos S. G. W., Teytaud O. (2006). Modification of uct with patterns in monte-carlo go. *Technical Report RR-6062*, vol. 32, p. 30–56.
- Nitti D., Belle V., De Raedt L. (2015). Planning in discrete and continuous markov decision processes by probabilistic programming. In *Joint european conference on machine learning and knowledge discovery in databases*, p. 327–342.
- Paul M. (2010). Bethe. *The state of automated bridge play*.
- Pepels T., Cazenave T., Winands M. H. (2015). Sequential halving for partially observable games. In *Workshop on computer games*, p. 16–29.
- Schaeffer J., Burch N., Björnsson Y., Kishimoto A., Müller M., Lake R. *et al.* (2007). Checkers is solved. *science*, vol. 317, n° 5844, p. 1518–1522.
- Silver D., Huang A., Maddison C. J., Guez A., Sifre L., Van Den Driessche G. *et al.* (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, vol. 529, n° 7587, p. 484–489.
- Smith S. J., Nau D. S., Throop T. A. (1996). Total-order multi-agent task-network planning for contract bridge. In *Aaai/iaai*, vol. 1, p. 108–113.
- St-Pierre D. L., Teytaud O. (2014). The nash and the bandit approaches for adversarial portfolios. In *Computational intelligence and games (cig), 2014 ieee conference on*, p. 1–7.
- Sutton R. S., Barto A. G. (1998). *Reinforcement learning: An introduction* (vol. 1) n° 1. MIT press Cambridge.
- Wasserman A. I. (1970). Realization of a skillful bridge bidding program. , p. 433–444.
- Yegnanarayana B., Khemani D., Sarkar M. (1996). Neural networks for contract bridge bidding. *Sadhana*, vol. 21, n° 3, p. 395–413.

Yeh C.-K., Lin H.-T. (2016). Automatic bridge bidding using deep reinforcement learning. *arXiv preprint arXiv:1607.03290*.

Annexe A. GLOSSAIRE

Sources : wikipedia.org et <http://www.lebridge.info/vocabulairedubridge.html>

A

Affranchir une couleur : jouer une couleur suffisamment souvent pour y promouvoir ses cartes restantes en cartes maîtresses.

Annonce : synonyme d'enchère.

Appel : convention au jeu de la carte qui permet d'indiquer au partenaire si on est intéressé par une couleur donnée ou non.

Atout : couleur préférentielle, choisie au cours des enchères, et qui permet, notamment, de couper les cartes maîtresses de l'adversaire.

B

Barrage : se dit de certaines annonces qui ont pour but d'empêcher les adversaires d'annoncer le bon contrat.

Bicolore : se dit d'une main qui possède deux couleurs longues.

C

Chicane : couleur dans laquelle un joueur ne détient aucune carte (« être chicane cœur »).

Conventions : ensemble d'enchères artificielles avec une signification bien définie ou ensemble de règles au jeu de la carte permettant par exemple de faire des appels.

Couper : jouer une carte d'atout, lorsqu'on ne possède pas la couleur demandée par l'adversaire ou le partenaire. La coupe permet (s'il n'y a pas de surcoupe) de remporter le pli.

D

Distribution : nombre de cartes qu'un joueur possède dans chaque couleur. L'ordre n'intervient pas, une distribution 5-4-3-1 est une distribution dans laquelle le joueur possède 5 cartes dans une couleur (peu importe laquelle), 4 et 3 dans deux autres couleurs et une carte seulement dans la dernière. Une distribution de ce genre est dite irrégulière. Pour simplifier on dira qu'une main régulière est une main sans singleton ni chicane.

Donne : distribution des 52 cartes aux quatre joueurs. Par extension, ensemble des phases de jeu par rapport à cette distribution.

Défausser (se) : jeter une carte d'une couleur différente de la couleur demandée, quand on ne possède pas de carte dans cette couleur.

Doubleton : couleur dans laquelle un joueur détient exactement deux cartes (« être doubleton cœur »).

E

Enchère d'essai : enchère visant à exprimer une incertitude concernant le palier du contrat idéal. Cette enchère pose une question au partenaire comme par exemple : « faut-il jouer la manche ou se contenter d'une partielle ? ».

Enchère artificielle : enchère qui par agrément, transmet au partenaire une information (différente de celle généralement admise) autre que l'intention de jouer dans la dénomination faite ou dans la dernière dénomination exprimée.

Entame : première carte jouée par le joueur qui se trouve à la gauche du déclarant.

H

Honneurs : l'As, le Roi, la Dame, le Valet et le Dix sont des honneurs, les autres cartes (du 2 au 9) sont appelées des petites cartes.

I

Impasse : technique de jeu consistant à manier une couleur de sorte à pouvoir faire un ou plusieurs plis avec des cartes de rang inférieur à celles de l'adversaire.

L

Levée : unité qui permet de déterminer le résultat du contrat. Une levée est formée par les quatre cartes que chacun des joueurs joue à son tour. La levée est remportée par le camp du joueur ayant mis la carte la plus forte.

Ligne de jeu : stratégie suivie par le déclarant pour mener à bien son contrat.

Longue : se dit d'une couleur qui possède au minimum cinq cartes.

M

Majeure : une des deux couleurs Pique ou Cœur.

Marque duplicate : compétition opposant deux équipes où chaque donne est dupliquée pour être jouée par un camp à une table et par l'autre camp ensuite.

Mineure : l'une des deux couleurs Carreau ou Trèfle.

O

Ouverture : un joueur ouvre s'il est le premier à la table à faire une annonce autre que Passe.

Ouvreur : joueur qui a ouvert .

P

Par d'une donne : résultat théorique auquel une donne devrait aboutir si tous les joueurs jouent parfaitement et ont une connaissance parfaite de la donne.

S

Sans-Atout : caractère d'un contrat dans lequel il n'y a pas d'atout.

Singleton : couleur dans laquelle un joueur ne détient qu'une seule carte.

Annexe B. Evaluation des mains

1) **Le compte des points d'honneur** le plus utilisé pour l'évaluation des mains est le suivant : 4 points pour un As, 3 pour un Roi, 2 pour chaque Dame et 1 point par Valet détenu. Il existe d'autres méthodes avec des valeurs différentes qui prennent par exemple en compte le nombre de 10 ou d'autres critères.

2) **Le compte des points de distribution** consiste à tenir compte de la distribution d'une main pour ajouter des points au compte des points d'honneur de la main. En général, on ajoute un point pour une belle couleur cinquième, un point si on a un singleton dans une couleur annexe (à moduler si on sait que c'est une couleur longue chez le partenaire), et deux points pour une chicane*.

