



Boosting a Bridge Artificial Intelligence

Veronique Ventos, Yves Costel, Olivier Teytaud, Solène Thépaut Ventos

► To cite this version:

Veronique Ventos, Yves Costel, Olivier Teytaud, Solène Thépaut Ventos. Boosting a Bridge Artificial Intelligence. 2017. hal-01665867

HAL Id: hal-01665867

<https://hal.science/hal-01665867>

Preprint submitted on 17 Dec 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Boosting a Bridge Artificial Intelligence

Véronique Ventos^{*,1}, Yves Costel², Olivier Teytaud^{†,1}, and Solène Thépaut Ventos³

¹LRI, INRIA, CNRS UMR 8623, University Paris-Saclay, France

²WBRIDGE5, Lyon, France

³LMO, CNRS UMR 8628, University Paris-Saclay, France

Preprint submitted and accepted to International Conference on Tools with Artificial Intelligence (ICTAI 2017). The final version will be included at the ICTAI proceedings

Abstract

Bridge is an incomplete information game which is complex both for humans and for computer bridge programs. The purpose of this paper is to present our work related to the adaptation to bridge of a recent methodology used for boosting game Artificial Intelligence (AI) by seeking a random seed, or a probability distribution on random seeds, better than the others on a particular game.

The bridge AI Wbridge5 developed by Yves Costel has been boosted with the best seed found on the outcome of these experiments and has won the World Computer-Bridge Championship in September 2016.

1 Introduction

Games have always been an excellent field of experimentation for the nascent techniques in computer science and in different areas of Artificial Intelligence (AI) including machine learning (ML). Despite their complexity, games problems are much easier to understand and to model than real life problems. Indeed, most games have a limited number of simple rules and have been subject to in-depth human analysis over time. Systems initially designed for games are then used in the context of real applications. Therefore, next-generation Watsons, the IBM system that has beaten two champions of Jeopardy!¹ in 2011, are used as consultant machines in fields as varied as medicine, cybersecurity, and business analytics.

^{*}Corresponding author: `ventos@lri.fr`

[†]Now at: google

¹Jeopardy! is a game consisting of finding a question from clues constituting the Response expressed in natural language.

In the last decades, the design of champion-level systems dedicated to a game (game AI) were considered as milestones of computer science and AI. In 1994, the checkers (english draughts) player system Chinook [16] became the first computer program to win a world champion title against humans. Three years later, the IBM supercomputer Deep Blue defeated the world chess champion Garry Kasparov. In both cases, systems benefited from the increased computing power of computers. More recently, despite increasing processing power, a major challenge of Artificial Intelligence was to design a computer program reaching the level of a human Go grand master. Until 2006 the best Go programs did not exceed the level of an average amateur player. The first revolution came with Mogo ([14],[7]) and Craizystone [5] two Go AI with strategies combining several ML methods including an adaptation to the UCT algorithm (*Upper Confidence bounds applied to Trees*) which allowed to explore more often the best node in a tree search. In march 2016, AlphaGo designed by a team of DeepMind, a British company specializing in artificial intelligence and games [17] became a grand master after winning four games in one match against Lee Sedol, one of the best players in the world. In May 2017, AlphaGo defeated by 3 to 0 the world champion Ke Jie.

In Checkers, Chess or Go, players share complete information about the current state of the game. A longstanding challenge of AI is related to games of incomplete information where the game situation world is only partially observable. For instance, Poker is an incomplete information game because opponents hands remain hidden from view during play. In January 2017, the Poker AI Libratus developed by Carnegie Mellon University won a heads-up no-limit Texas hold'em poker event against four of the best professional players. This success is not yet happened with regard to the game of bridge another game of incomplete information which then provides a challenging problem for AI.

Bridge is a trick-taking game, played with 52 standard cards opposing two pairs of players. Cards are dealt randomly to the four players; each of them only sees his hand, that is to say the 13 cards he received. Bridge is therefore an incomplete information game where the players do not have common knowledge of the game being played. Throughout the game, the hidden information is reduced, the incompleteness decreasing either in a certain manner (e.g cards put on the table) or with a high probability (e.g. by information exchanged during the biddings step). In this game of incomplete information, the main goal of each player consists in 'rebuilding' the hidden hands in order to make decisions. Most current bridge programs use Monte-Carlo simulations, i.e. producing repeated random samples and considering their average in making a decision. Such Monte-Carlo methods were initiated as early as 1949 by Nicholas Metropolis [12]. The purpose of our work is to adapt to the game of bridge a recent seed methodology which optimizes the quality of the simulations and which has been defined and validated in other games as shown in [19, 3] and to test whether the random seeds have an impact on the bridge AI performance.

Since 1996, best bridge programs can annually participate in an official World Computer-Bridge Championship (WCBC). This competition is organized by the World Bridge Federation (WBF, President Gianarrigo Rona) and the American Contract Bridge League under direction of Alvin Lévy (www.computerbridge.com/). To check whether the adaptation to bridge could be efficient (the gain is variable according to the games), we chose among the participants in WCBC a bridge AI to be boosted. The selected bridge AI is the one developed by Yves Costel for the bridge program Wbridge5² (see fig 1). WBridge5, boosted as described in the present paper, was ranked first in the round-robin of the world computer bridge championship 2016, and then won the semifinal and final.

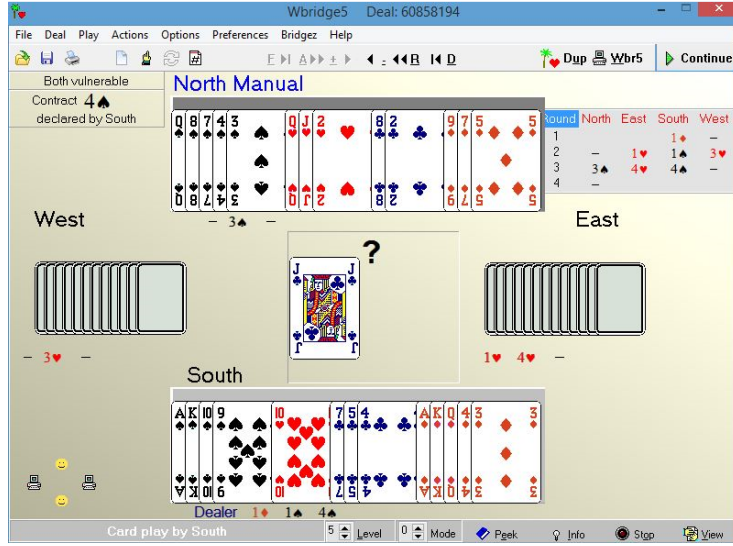


Figure 1: Screen shot of Wbridge5

This paper outlines the methodology and the experimental results related to the adaptation of the seed methodology to Wbridge5 together with the description of the world computer bridge championship 2016.

To begin with, in section II we introduce the game of bridge, we describe a state of art of computer bridge and we summarize the seed methodology. Section III presents an adaptation of the seed method on Wbridge5. Experiments related to the boosting of Wbridge5 are described in section IV. Section V is dedicated to describing the world computer bridge championships won by the boosted version of Wbridge5. Finally, related and future work are presented section VI.

²<http://www.wbridge5.com>

2 Prerequisites

In the present section, we introduce the prerequisites, namely bridge, computer bridge and the seed methodology.

2.1 Bridge

Bridge is made of two parts full of complexities: the bidding phase then the card play. The bidding phase can be seen as a coded language used by players to pass information to their partner about their hand (most of the time, balanced or unbalanced, suits distribution and strength according to the number of points of their cards). The goal for each side is to reach an optimal contract. The contract specifies the minimum number of tricks among the thirteen to be won in the second phase. During the card play the goal is to fulfill (or to defeat for the opposite side) the contract reached during the first step.

From computer game aspects, bridge is a multi-player game:

- **with incomplete information:** players only see a portion of the cards, therefore they have an incomplete knowledge of the state. They also do not know the gains resulting of their actions. However, players know : the possibilities of actions, other player's possibilities of actions and their motives.
- **with perfect memory:** it is not always the case in practice, but we can assume each player can remember which cards have been played previously and by whom.
- **non-cooperative:** bridge is not a cooperative game even though there is cooperation between partners.
- **sequential:** for the two parts of a board, the players' order of actions is decided (clockwise).
- **finite:** the set of strategies of each player is finite.
- **with constant sum:** not exactly with zero sum but with constant sum since both pairs are playing for thirteen tricks.
- **incrementally scoring:** the number of won tricks used to compute the score increases all among the board.

Scoring in Bridge

The possible results for the games covered in previous seed approaches [3, 11, 4] like Domineering, Atari-Go, or Phantom-Go (Go with hidden information) are 0 for a loss, 0.5 for a draw and 1 for a win. In bridge, we need more precise results. Indeed, it is important to know the exact points difference between the winner and the loser. Here are some explanations about the scoring of a board. Bridge is a trick-taking game where the score is related to the contract reached during the bidding phase and the number of won tricks in the card play phase.



Figure 2: Bidding box used in duplicate bridge competitions

The different contracts are represented in figure 2. Let n be a number between 1 and 7 and $S \in (\spadesuit, \heartsuit, \diamondsuit, \clubsuit, \text{NT})$. The contract nS determines the minimum number of tricks the pair commits to win ($n + 6$) and which suit is the trump, NT to expressing the fact that there is no trump. For instance $4\heartsuit$ is fulfilled if the number of tricks won is at least 10 ($4 + 6$) with a \heartsuit trump. The rules during the card play step are quite simple. For every trick, each player plays one card in turn on the table. When the four players have played a card, the trick is over and is won by the player who played the highest card in the suit originally played, or by the highest trump. He will also be on lead at the following trick. The board is over when all the 52 cards have been played. The number of tricks won by each pair is checked and the score of the board is calculated from this number and the number of won tricks required by the contract. For instance, let us consider that a pair has reached the contract of $4\heartsuit$. If the pair wins ten tricks as required by the contract, it obtains a score of +420, if the pair gets only 9 tricks the contract is defeated and the side scores -50,

We focus on duplicate bridge scoring (see https://en.wikipedia.org/wiki/Bridge/_scoring for more details) which is based on relative performance reducing the randomness factor. A team match is a match between two teams, each team being constituted by two pairs. Same boards are played at two different tables where one pair from each team is seated in opposite directions. Players are represented by North, East, South and West (abbreviated to N, E, S, W) the pairs being NS and EW. If the first pair of team A is sitting in North-South and playing against the first pair of team B, then the second pair of team A is sitting in East-West at the other table and playing against the second pair of team B. The final scoring for a board consists in computing the difference between the scores at the two tables. For instance if Team A got +420 points and Team B -50, the score is +470 for the team A. This score is then converted in International Match Point scale (IMPs) in order to compress potential big

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 3: Table for converting points into IMP

differences. Note that the conversion is not linear (see Table 3). A team can obtain between from -24 to 24 IMPs for each board. The score of a match is the sum of the scores in IMPs of each board.

A bridge game consist in n boards with n known in advance (most of the time, n is between 20 and 100). The winning team is the one with the best cumulative score in IMPs over the n boards.

2.2 State of art of Computer Bridge

Some algorithms were designed in the 60s and the 70s to solve bridge problems but their performance was very modest due to computing power limitations [2, 21]. Bridge Baron, which was created in 1983, was the first computer program allowing to process complete deals, at a very modest playing level though. Facing a card play problem, a human player elaborates a plan involving different steps displaying several bridge techniques: finesse, squeeze, etc ([18]) applied with reasonable success AI planning methods (Hierarchical task network) in order to generate and evaluate these techniques. This approach allowed Bridge Baron to win the WCBC in 1997, although its level was much weaker than the average level of a non-professional bridge player.

More recent programs use for the card play a technique widely known as double-dummy analysis (DDA): a solver maximizes the number of tricks for each side in a simplified version of bridge. In this version, the final contract is given and the 4 hands are known: the general framework is deterministic and it is a situation of complete information. More precisely, the DDA gives the number of tricks won by each side for Spade, Heart, Diamond, Club or No Trump contracts when all four players know the emplacement of the 52 cards, and each player plays optimally. A tree-search can then be used in this simplified game, as well as routine alpha-beta methods. Values of the leafs are computed using the DDA. As the number of won tricks is growing, this number offers upper and lower boundaries for the alpha-beta process, with smaller and smaller intervals. DDA is the basis of bridge programs since it is used as an evaluation function even in the bidding part.



Figure 4: Screenshot of BBo's retransmission. In the upper right corner: the bidding sequence, in the lower left corner: the current scores in IMPs of each team

Albeit these improvements the branching factor b remains too high for the card play problems to be solved by brute force. Bridge programs began to use the following two steps method: reducing the state-space using symmetries of bridge game (typically defining small cards as equivalent) then using the double dummy solver on a sample of boards generated using Monte Carlo methods and consistent with the sequence of auctions and the cards played so far. The branching factor is then significantly reduced from b to $0.76b$. Broadly known as partition search, Ginsberg formalized the method in [9] and more thoroughly in [8]). The GIB (Ginsberg's Intelligent Bridge player) program won the WCBC in 1998 and 1999.

The best-known online game is Bridge Base Online (BBO) represented in figure 4. Bridge Base Inc. was founded in 1990 by Fred Gitelman. BBO is free, operates as a supervisor and allows human players to use GIB robots.

[15] establishes an extensive state of art on computer bridge. Unfortunately, few articles are written, the designers of the bridge programs being reluctant to reveal details about their code, but it seems that they have been using similar techniques with only slight variations.

2.3 Seed methodology

Recently, a methodology [19, 3, 4, 11] has been proposed for boosting the computational intelligence of randomized game-playing programs. The principle of the approach is: given a stochastic AI, checking its performance by comparing different deterministic versions of the AI in order to choose the best ones.

A stochastic computer program uses a random seed which is a number initializing a pseudo-random number generator. For one seed there is only one generated pseudo-random sequence. In the field of computer security a random seed can then be used as a secret key. The determination of the AI is done according to the fixation of the random seed. It is then possible to compare different deterministic versions of the AI and to plot the winning rates, sort, and compare the variations to the standard deviations.

Let us introduce some definitions:

- **The game is stochastic if**, even if each player uses a same deterministic policy, the result is not the same. For example, Bridge is stochastic from the shuffling of the cards; Monopoly is stochastic, because there are dice.
- **The AI is stochastic if**, even if the information it has received is exactly the same, with exactly the same history of moves, it will not necessarily play the same moves. Typically, an AI is stochastic if it uses pseudo-random.
- **A stochastic AI has clean random seeds if** it is not stochastic when you set up a random seed.

This methodology is designed for an AI which is both

- stochastic;
- with clean random seeds.

To apply the seed methodology, we must first define a simulator. Then, in the case of 2 players (or two teams) let us use notation $r = \text{simulator}(s_1, s_2)$ where:

- s_1 is the seed for player 1.
- s_2 is the seed for player 2.
- r is the result of the game, 1 if player 1 wins, 0.5 in case of draw, 0 if player 2 wins. You might have intermediate values for games with scores; the principle is to have greater values when player 1 performs better.

Because the AI has clean random seed, it becomes deterministic when the seed is fixed. To compare the different seeds, we need to build a matrix with the results of games between the different deterministic AI.

For the simplest seed methodologies, we then build a matrix R with $R(i, j) = \text{simulator}(i, j)$, for $i \in \{1, 2, \dots, N\}$ and $j \in \{1, 2, \dots, N\}$, which will be used for choosing the best seeds. In fact, seed methodologies include several approaches:

- BestSeed approach:
 - just pick up i such that the sum $R(i, 1) + R(i, 2) + \dots + R(i, N)$ is maximum for choosing an excellent seed i for player 1.
 - and pick up j such that the sum $R(1, j) + R(2, j) + \dots + R(N, j)$ is minimum for choosing an excellent seed j for player 2.
- Robust BestSeed approach: instead of the best seed, pick up the k best seeds, where k is a robustness parameter, and your policy is the uniform policy over these k seeds.
- Nash approach: compute the Nash equilibrium of the matrix R (with row player maximizing); this provides p (probability distribution on seeds for the first player) and q (probability distribution on seeds for the second player). Then:
 - choose seed i with probability $p(i)$ for the player 1;
 - choose seed j with probability $q(j)$ for the player 2.

It has been empirically shown in ([19, 3, 11]) that the seed methodology optimizes the quality of the simulations in several games like Atari-Go, Domineering, Breakthrough. The success rate of the boosted algorithm against the non-boosted baseline is usually positive even in the case of an incomplete information game like Phantom-Go [4]. The goal is to check whether it is the case for bridge AI.

3 Adaptation of the seed methodology to computer bridge

In bridge AI, seeds are used to generate samples of hidden hands according to the context. These samples are used to evaluate decisions to make without knowing all of the cards in play.

3.1 Monte-Carlo in computer bridge

Monte-Carlo simulations are mainly used to make decisions in two situations: late decisions in the bidding phase and card play.

- **bidding:** a set of rules is manually built from human expertise according to the chosen bidding system known by the four players. First auctions are often chosen according to this set of rules and the knowledge of classical situations. Some boards can be handled without using any simulation in the bidding part if the previous architecture is sufficient. However, a set of rules does not cover all the auctions (there exist approximately 10^{47} different sequence of auctions). Human players have the same problems, they use rules of the bidding system when it is possible but sometimes

they must make decisions not handled by the rules. In this case, good players compare the different options (in general there are 2 or 3 possibilities) by imagining some possible hidden hands according to the previous auctions. They choose the best auction related to these possibilities. For bridge programs, simulations are mainly used to choose the final decision (the contract that will be played) and as humans, they allow the system to choose between several (2 or 3) predetermined choices. In this case, there is generation of hands (partners and opponents) consistent with the current bidding sequence since it represents constraints on hands. For each generated board (a generation is equivalent to a possible real board), the result of the different options is evaluated using the DDA. The system then chooses the best auction according to this evaluation.

- **Card play:** During this phase, there are only three active players: the declarer and the two defenders. Indeed, after the lead (i.e. the first played card), the declarer's teammate (called the dummy) puts his hand face up on the table and then bows out of the action entirely. Each player (defenders or declarer) sees his own hand and the hand of the dummy. The declarer must imagine, according to the sequence of auctions and the lead, the two hidden hands and take successive decisions in order to maximize the number of tricks. Nowadays, computer programs simulate samples of hands for the opponents and select the winning action in most cases according to DDA.

3.2 Chosen AI: Wbridge5

The purpose is to check whether the optimization seed method can be effective on bridge AI. The first step consists in choosing a bridge AI which satisfies the different conditions for which the method can be applied. The selected AI is the one developed by Yves Costel for the bridge program Wbridge5. The program satisfies the seed methodology conditions since it is stochastic and it uses pseudo-random number generators in order to obtain a sample of hidden hands and allows to set the generators seed which makes it deterministic.

The second step consists in making some choices related to the adaptation of the seed methodology to the game of bridge which is different on several aspects. First of all, the methodology has been previously studied in the context of two player games and for games with only one step whereas bridge is a four player game with two different steps (bidding and card play).

We must handle the fact that there are four AI (two against two). Therefore, four seeds need to be chosen. In order to make the interpretation of the results easier, we choose a seed for each pair instead of for each player.

Another choice is related to the fact that the two phases of bridge are so different that they can be seen as two different games. Furthermore, bidding and card play do not use the same logic. The simulations are not used in the same way when you have to take a decision during the bidding and during the card play. A bad score can be due to a sub-optimal contract or bad card play

In order to make the experiment more accurate, we need to test the seed used during the bidding according to a perfect play and the seed used during the card play by assuming that the contract is optimal.

We choose to restrict experiments to the optimization of the biddings seed for the following reasons: the final contract is crucial - even if your card play is optimal, the pair obtains a bad score if the contract is wrong. Bidding phase is known to be one of the most difficult problems for computer bridge. Simulations are more useful during the bidding where the incompleteness concerns three hands, that is to say 39 cards, while there are at most 25 unknown cards during the card play after the lead. Note that the choice of the lead is decided by another part of the program and that the last tricks are almost always played with all the cards known by all the players. Finally, information from the bidding and the cards played previously are available during card play - therefore, the number of unknown cards is restricted enough to make possible the use of more costly methods without DDA. In practice, for computational reasons, systems use double-dummy approach for choosing then playing cards at the beginning of the phase (between 0 to 4 first played cards depending on the system) and use single-dummy approaches, which are more effective but more costly, for the rest of the board. Single-dummy methods are usually not documented since developers keep their own version private. Note that approaches used by bridge AI in the card play stage have satisfying results but also various shortcomings: the simulation takes into account the opponents' bids (or the absence of bidding), but it does not (yet, and usually) take into account their earlier actions during the card play - whereas a human player takes into account his opponents' auctions, bridge computer programs are often unable to have this kind of reasoning. Finally, card play being very time-consuming, we do not use the card play module of Wbridge5. The evaluation function is defined according to DDA as explained in the following section.

4 Experiments

The first step of experiments consists in building a matrix with the results of matches between the various deterministic versions of Wbridge5 which can be generated from the stochastic AI by choosing seeds since Wbridge5 has clean seeds. The matrix allows us to compare the different seeds.

In the following, let us denote $WB(i)$ the deterministic version of Wbridge5 with seed i .

4.1 Bridge data

The duplicate scoring (see section II.A) is the scoring used for official competitions both for humans and robots. The size of matches in computer bridge competitions is usually 64 boards.

In order to compare 40 seeds, we build a 40x40 matrix R ; each entry $R(i, j)$ containing the result in IMPs of a 64-board match between $WB(i)$ and $WB(j)$.

As an illustration, $R(2, 3) = 12$ represents the fact that the Wbridge5 with seed 2 against wbridge5 with seed 3 wins the 64-board match with an advance of 12 IMPs.

Description of a match WB(i) against WB(j):

there are two game tables with for each table two instances of WB(i) and two instances of WB(j). At table 1, instances of WB(i) are NS and instances of WB(j) are EO. At table 2, instances of WB(j) are NS and instances of WB(i) are EO. The 64 boards are the same at the two tables. Each AI has then exactly the same cards as the opponent at the other table.

Scoring: let us recall that we want to evaluate the biddings according to an optimal card play. Moreover, the card play stage is computationally expensive. Consequently, we decide to only use the bidding part of wbridge5 and to compute the score at each table by comparing the contract reached by the AI and the number of won tricks computed with the double-dummy solver of Wbridge5. It is then sufficient to compute the difference in points between the score linked to the two tables. Finally, the difference is converted into IMPs according to the conversion table 3.

Remarks:

1. During the experiment, we observe that there is no simulation for 10% of boards, and in 75% of cases the contract is the same for the two different seeds considered. In these two cases, the score of the board is then 0.
2. The seed has an impact in 15% of cases but these cases concern important decisions with high value in IMPs (this kind of boards are called swing boards).
3. Since the AI is deterministic when the seed is fixed, $R(i, j) = -R(j, i)$ and the diagonal contains 0 values since it concerns matches between four same AI.
4. The set of boards of each match is disjoint from the set of boards of every other match. Each seed is then tested on 2496 (39×64) different boards. The matrix is built using 49920 boards ($(40 \times 39 \times 64) / 2$).

Results: here is the vector of the cumulative scores is, from seed 1 to seed 40:
 (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).
 The bigger is the cumulative score, the better is the seed. The worst seed is the seed 22 (with a cumulative score of -263) and the best seed is the seed 39 (+269). The won matches rate is : 56.4% for seed 39 and 41.02% for seed 22.

There are several competing approaches in the seed methodology [19, 3, 4], including BestSeed and Nash. The BestSeed approach usually provides greater success rates against the original AI than the Nash approach. On the other hand, the Nash approach is more robust and provides significant improvements against opponents who might learn against our algorithm. More precisely, the Nash approach allows us to obtain a stochastic boosted AI by giving a probability distribution over the seeds. The goal is to make sure the AI is not predictable for the opponents. Because of the complexity of the game, predictability is not a problem for bridge. Therefore we use BestSeed allowing use to choose the seed 39 with +269 cumulated IMPs.

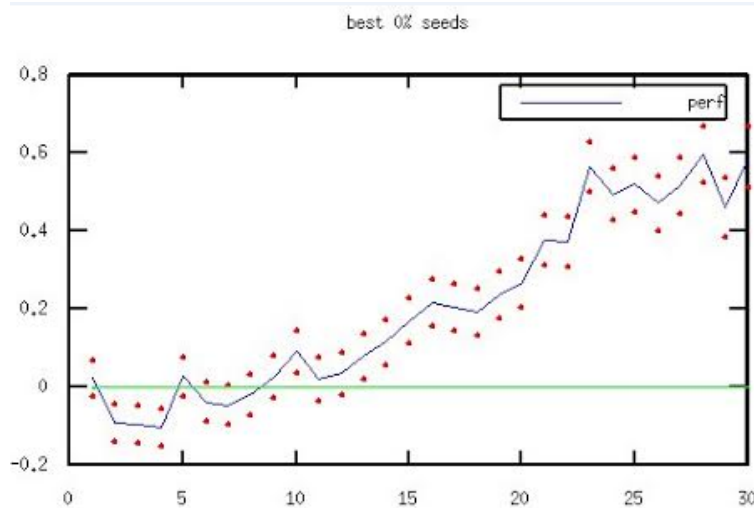


Figure 5: Performance of the best seed (39) w.r.t. the size of a randomly known submatrix

In the two following sections, we present some results related to the performance of the best seed found. The first approach presented in B. is a cross-validation one on the current matrix. The second approach presented in C. consists in empirically testing the performance of the best seed over new 1000-board match between different versions of Wbridge5 and the boosted one.

4.2 Validation w.r.t. the impact of the matrix size

The validation is related to the 40x40 matrix mentioned above. Importantly, we properly apply cross-validation - the performance is tested in parts of the matrix which have not been seen. The general idea of this cross-validation is to study the impact of the best seed by randomly keeping a part of the matrix to form the test set.

For each submatrix size (from 1 to 30), we compute the performance, in IMPs, of the best seed selected on a randomly known submatrix against the remaining seeds. The process is iterated in order to obtain an average performance for each matrix size. Results of the cross-validation of the best seed are presented in Fig. 5 where the x-axis represents the size of a the submatrix, while the y-axis represents the average performance, in IMPs, of the best seed selected on this submatrix against the remaining seeds.

The results of the robustified version are presented in Fig. 6 where the x-axis represents the size of a the submatrix, while the y-axis represents the average performance, in IMPs, of the the best X% seed selected on this submatrix against the remaining seeds.

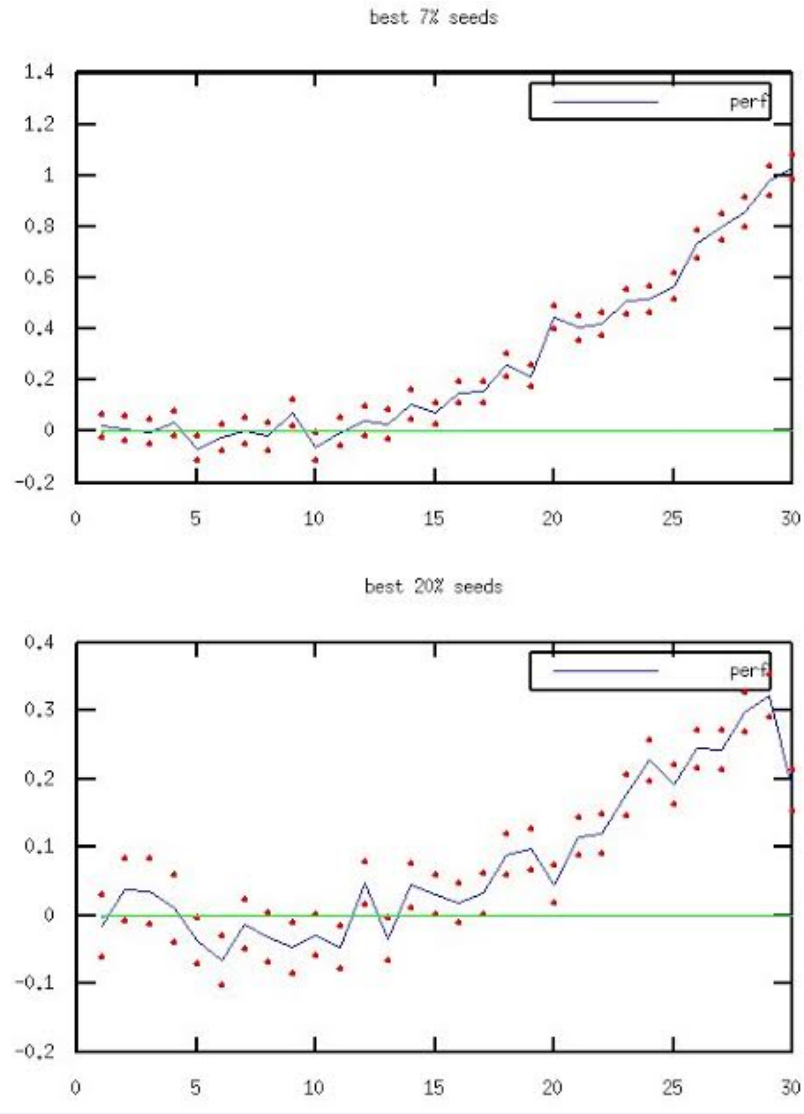


Figure 6: Performance of the X% best seed approach w.r.t. the size of a randomly known submatrix

4.3 Validation according to empirical results out of the matrix

We test the best seed found in the matrix on new 1000-board matches. As previous, scores are computing using the double-dummy solver. Here are some results related to the cumulative score and to the rate of won boards.

We compare the deterministic program with the best seed (seed 39) and the worst (seed 22). The program with the best seed wins with a difference of 163 ± 2.62 IMPs, over 1000 games, i.e. 0.16 IMP per board. The rate of won boards for seed 39 is : 59%.

According to the fact that the cumulative scores of the best seed and the second one (seed 13 with 266 IMPs) were very close, we decide to compare the two seeds in a new 1000-board match. The program with the best seed wins with a difference of 138 ± 2.86 IMPs, over 1000 games, i.e. 0.13 IMP per board. The rate of won boards for seed 39 is: 54%.

Finally, we test the performance of the best seed with a third match against the program with the current seed (99) which had been chosen arbitrarily in the past of WBridge5. The program with the best seed wins with a difference of 97 ± 2.75 IMPs on 1000 boards. This means an improvement of $\simeq 0.1$ IMP per board.

Without going into further details it is important to point out that a gain of 0.1 IMP per board is a great improvement for bridge players with similar game level. The won boards rate for seed 39 against the seed 99 is: 55.5%. These experimental results ensuring a significant performance improvement compared to the original WBridge5, Yves Costel decides to use the boosted version for the Wbridge5 participation to the 20th World Computer-Bridge Championship.

5 World Computer-Bridge Championship



Figure 7: World Computer-Bridge Championship in Wroclaw, september 2016

The 20th World Computer-Bridge Championship was held on September 10-15, alongside the World Bridge Games in Wroclaw, Poland. The organization is carried out by Alvin Levy who provided boards, played a tourney director role and established the ranking (for more details see <https://bridgerobotchampionship.wordpress.com/>). The technical part is entirely

handled by Gérard Joyez who has developed the central program Table Manager of these championships which allows the sharing of information between robots while avoiding cheating. This year the 8 participants and their awards are : Bridge Baron (USA, 1997), Meadowlark Bridge (USA, 2000), Micro Bridge (Japan), Q-Plus Bridge (Germany), RoboBridge (Netherlands), Shark Bridge (Denmark, 2011 and 2014), Wbridge5 (France, 2005, 2007 and 2008) et Xinrui (China). We note that the absence of the current world champion Jack (Netherlands), the come-back of Meadowlark Bridge and the new entry of Xinrui which is a serious opponent since Xinrui is designed by a company dedicated to computer bridge founded by Yuzhang Liu with a team of 20 persons.

The competition begins with a qualification stage called Round-Robin where every AI play against each other in 32-board matches. At the end of this stage, the four best programs are qualified for the semi-finals, the first one faces the fourth, while the second faces the third. The two winners of these semi-finals face each other for a final match of 64 boards.

When there are several matches between teams as it is the case during the Round-Robin, for each match the difference in IMPs is converted into Victory Points or VPs with a minimum of 0 and a maximum of 20 per match. The use of such scales reduces the impact of possible massive IMPs blow-outs that may occur in some matches. Here is the overall team ranking related to the cumulation of VPs during the Round-Robin (7 matches): (Wbridge5 91.87, Micro Bridge 90.07, Bridge Baron 89.21, Shark Bridge 80.12 , Q-Plus Bridge 78.76 , Xinrui 78.50 , RoboBridge 50.58 , Meadowlark Bridge 0.89).

Wbridge5 finishes first of the Round-Robin with a very little lead over the second and third Xinrui, the newcomer, with a great start, was about to qualify before its last match against Shark Bridge where it was heavily defeated. Generally speaking, the competition between the first six programs was very tight. In this way, Q-plus bridge only lost the fourth place during the last board of the last match. The semi-finals and finals are also very tight. Since it is a K.O. phase, scores are in IMPs. Micro-Bridge beaten Bridge Baron with a fewer difference: 144-138 IMPs. Wbridge5 wins in semifinal 140.6-131 IMPs. This difference of 9.6 IMPs is about the same as the carry-over coming from the victory of Wbridge5 against Shark during the Round-Robin. The final is as tight as the semifinals. Before the last two boards, Wbridge5 loses by 17 IMPs against Micro Bridge. However, Wbridge5 wins 23 IMPs in the last two boards for a final score of 162 to 156. The difference of 6 IMPs for 64 boards corresponds to a gain of 0.09375 IMP per board. The estimated advantage of the best seed in comparison to the old one is $\simeq 0.1$ IMPs per board. All over the competition, the AI made the difference in the bidding part by finding a better contract than its opponent. Our empirical experiments show that the seed has an impact on difficult decisions like these ones. Finally, Wbridge5 won its fourth world champion title, eight years after its last victory.

6 Conclusions

6.1 Related work

A large part of the current research effort in card games uses deep learning (e.g. [13] for Poker), with clear successes; this has reached Bridge (see the recent [10, 23] and the older [22]). In [23], a deep reinforcement learning model has been designed in order to achieve automatic bid learning task in a subproblem of bidding called non competitive bidding sequences. This kind of bidding sequence occurs in only 27 % of boards (statistics obtained from the data on the WBF site). Besides, the subproblem is related to a two-player process rather than to a four-player one. Automatic bidding has been studied in [1, 6] where a PIDM (Partial Information Decision Making) algorithm has been designed in order to predict reasonable auctions. In [6], a self-organizing map neural network has been used to effectively bid no trump hands. Our approach is not in competition with these methods; it could be used on top of any method which uses stochasticity, in particular to avoid blunders related to unlucky seeds.

6.2 Work in progress and Further Work

Current research is directed towards extending this analysis and a flurry of experiments on Wbridge5 in progress aim to extend the framework. First of all, in order to decrease the offline computational cost, we begin to extend our work by using an advanced bandit methodology [11] rather than the brute force version used in the present work. The time gain will allow us to make more precise experiments in order to find best specific seeds for different situations in the bidding depending on the level of the contract (e.g slams which are contracts involving to win at least 12 of the 13 tricks). Finally, a possible extension of the work is related to the optimization of seeds used in the card play step.

As stated in [20], bridge is a great challenge for AI and much work related to the definition of a bridge AI remains to be done. Currently, the average level of best current bridge AI is well below world class level. The level of bridge programs is close to that of Go programs before 2006. The designing of a hybrid architecture including recent numeric and symbolic machine learning modules is currently underway. Work presented in this paper fits into this architecture. The seed methodology has been applied to several games, but to the best of our knowledge the present paper presents simultaneously the first application to Bridge and the first application to a world-level program. WBridge5 equipped with the seed methodology won the 20th World Computer-Bridge Championship, whereas it had not won this yearly competition since 2008.

ACKNOWLEDGMENT The authors would like to thank Bridge players J.B. Fantun, J.C. Quantin, J.P. Desmoulins and M. Schoenauer, M. Sebag co-heads of INRIA Saclay Project-Team TAU for all the constructive discussions we had together.

References

- [1] Asaf Amit and Shaul Markovitch. Learning to bid in bridge. *Machine Learning*, 63(3):287–327, 2006.
- [2] Elwyn R Berlekamp. Program for double-dummy bridge problems a new strategy for mechanical game playing. *Journal of the ACM (JACM)*, 10(3):357–364, 1963.
- [3] T. Cazenave, J. Liu, and O. Teytaud. The rectangular seeds of domineering. In *2015 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 530–531, Aug 2015.
- [4] Tristan Cazenave, Jialin Liu, Fabien Teytaud, and Olivier Teytaud. Learning opening books in partially observable games: using random seeds in phantom go. In *Computational Intelligence and Games (CIG), 2016 IEEE Conference on*, pages 1–7. IEEE, 2016.
- [5] Rémi Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In *International Conference on Computers and Games*, pages 72–83. Springer, 2006.
- [6] Lori L DeLooze and James Downey. Bridge bidding with imperfect information. In *Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on*, pages 368–373. IEEE, 2007.
- [7] Sylvain Gelly, Levente Kocsis, Marc Schoenauer, Michele Sebag, David Silver, Csaba Szepesvári, and Olivier Teytaud. The grand challenge of computer go: Monte carlo tree search and extensions. *Communications of the ACM*, 55(3):106–113, 2012.
- [8] Matthew L. Ginsberg. Gib: Imperfect information in a computationally challenging game. *Journal of Artificial Intelligence Research*, 14:303–358, 2001.
- [9] M.L. Ginsberg. Partition search. *Proceedings Of The National Conference On Artificial Intelligence*, 1996.
- [10] Chun-Yen Ho and Hsuan-Tien Lin. Contract bridge bidding by learning. In *Proceedings of the Workshop on Computer Poker and Imperfect Information at the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.
- [11] Jialin Liu, Olivier Teytaud, and Tristan Cazenave. Fast seed-learning algorithms for games. In *International Conference on Computers and Games*, pages 58–70. Springer, 2016.
- [12] Nicholas Metropolis and Stanislaw Ulam. The monte carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.

- [13] Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert-level artificial intelligence in no-limit poker. *arXiv preprint arXiv:1701.01724*, 2017.
- [14] Sylvain Gelly, Yizao Wang, Rémi Munos and Olivier Teytaud. Modification of uct with patterns in monte-carlo go. *Technical Report RR-6062*, 32:30–56, 2006.
- [15] M Paul. Bethe. *The state of automated bridge play*, 2010.
- [16] Jonathan Schaeffer, Neil Burch, Yngvi Björnsson, Akihiro Kishimoto, Martin Müller, Robert Lake, Paul Lu, and Steve Sutphen. Checkers is solved. *science*, 317(5844):1518–1522, 2007.
- [17] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.
- [18] Stephen JJ Smith, Dana S Nau, and Thomas A Throop. Total-order multi-agent task-network planning for contract bridge. In *AAAI/IAAI, Vol. 1*, pages 108–113. Citeseer, 1996.
- [19] David L St-Pierre and Olivier Teytaud. The Nash and the bandit approaches for adversarial portfolios. In *Computational Intelligence and Games (CIG), 2014 IEEE Conference on*, pages 1–7. IEEE, 2014.
- [20] Véronique Ventos and Olivier Teytaud. Le bridge, nouveau défi de l’intelligence artificielle ? *Revue d’Intelligence Artificielle*, 31(3):249–279, 2017.
- [21] Anthony I Wasserman. Realization of a skillful bridge bidding program. In *Proceedings of the November 17-19, 1970, fall joint computer conference*, pages 433–444. ACM, 1970.
- [22] B Yegnanarayana, Deepak Khemani, and Manish Sarkar. Neural networks for contract bridge bidding. *Sadhana*, 21(3):395–413, 1996.
- [23] Chih-Kuan Yeh and Hsuan-Tien Lin. Automatic bridge bidding using deep reinforcement learning. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI)*, pages 1362–1369, 9 2016.