Winning Boosting of a Bridge Artificial Intelligence

Véronique Ventos[1], Yves Costel[2], Olivier Teytaud[3], and Solène Thépaut Ventos[3,4]

[1]LRI, CNRS UMR 8623, Univ. Paris South, Paris, France
Email: ventos@lri.fr
[2]Lyon, France
Email: yves.costel@gmail.com
[3]Inria, CNRS UMR 8623, Univ. Paris-Sud, Gif-sur-Yvette, France (now at Google)
Email: olivier.teytaud@inria.fr
[4]LMO, CNRS UMR 8628, Univ. Paris-South, France
Email: solene.thepaut@u-psud.fr

***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 bridge adaptation 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 won the World Computer-Bridge Championship in September 2016.

I. Introduction

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 sees only his hand, that is to say the 13 cards he received. 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. Throughout the game, the incompleteness decreases 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. Most of current bridge programs use Monte-Carlo simulations to achieve this task. 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 [1], [2], [3], [4].
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 on the direction of Alvin Lévy (https://bridgerobotchampionship.wordpress.com/). To check whether the adaptation could be efficient on bridge (the gain is variable according to the games), we have merely chosen 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 Wbridge5[1] (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.

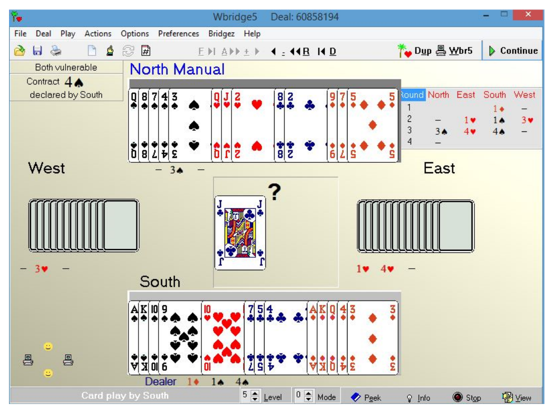

Fig. 1: Screen shot of Wbridge5

This paper outlines the methodology and the experimental results related to the adaptation of the seed methodology on Wbridge5 together with the description of the world computer bridge championship 2016.
To begin with, we introduce in section II the seed methodology, the game of bridge and we give a state of art of computer bridge. Section III presents the adaptation of the seed method on Wbridge5. Experiments related to the boosting of Wbridge5 are described in section IV. Finally, the section V is dedicated to the world computer bridge championships won by the boosted version of Wbridge5.

[1]http://www.wbridge5.com

II. PREREQUISITES

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

A. Seed methodology

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, because the initial deck is randomized; 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 $r = simulator(s_1, s_2)$ 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 AIs.

For the simplest seed methodologies, we then build a matrix R with $R(i,j) = simulator(i,j)$, for $i \in \{1, 2, ..., N\}$ and $j \in \{1, 2, ..., N\}$, 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 $R(i,1) + R(i,2) + ... + R(i,N)$ is maximum for choosing an excellent seed i for player 1.
  - and pick up j such that the sum $R(1,j) + R(2,j) + ... + R(N,j)$ is minimum for choosing an excellent seed j for player 2.
- Robust BestSeed: 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.

B. Scoring in Bridge

The possible results for the games covered in [2], [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 looser. 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. The different contracts are

Fig. 2: Bidding box used in duplicate bridge competitions.

represented in figure 2). Let n be a number between 1 and 7 and S $\in$ (♠, ♡, ♢, ♣, 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♡ will be fulfilled if the number of tricks won is at least 10 (4 + 6) with a ♡ 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 the first player of 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♡. If the pair wins ten tricks as required by the contract, it will obtain a score of +420, if the pair gets only 9 tricks the contract is defeated and the side will score -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 Est-West at the other table and playing

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

Fig. 3: Table for converting points into imp.

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 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.

C. State of art of Computer Bridge

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.

Long before the WCBC some algorithms were designed in the 60s and the 70s to solve bridge problems but their performance was very modest due to processing power limitations [5], [6]. 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 ([7]) 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.

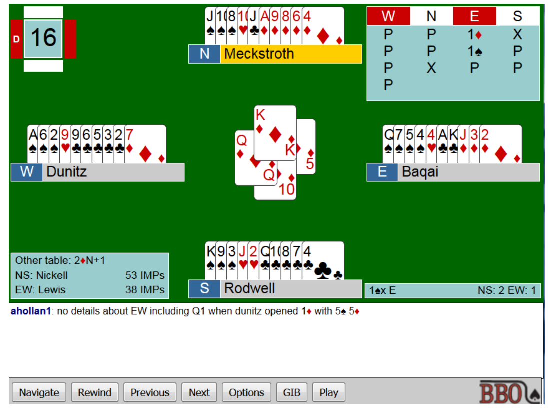

Fig. 4: Screenshot of BBo's retransmission. In the upper right corner: the bidding sequence, in the lower left corner: the current scores in IMPs of each team.

More recent programs use for the card play a technique widely known as double-dummy solver: the 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 double dummy solver 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 double-dummy solver. 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. Such a solver is the basis of bridge programs since it is used as an evaluation function even in the bidding part. Generally, bridge program designers defined their own solver but one can use the double-dummy solver developed by Bo Haglund which is the fastest and most widely used public-domain (http://privat.bahnhof.se/wb758135/bridge/index.html).

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 [8] and more thoroughly in [9]). 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.

Automatic bidding has been studied in [10], [11] where a

PIDM (Partial Information Decision Making) algorithm has been designed in order to predict reasonable auctions. In [11], a self-organizing map neural network has been be used to effectively bid no trump hands.

[12] establishes an extensive state of art on computer bridge. 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.

Currently, the average level of best current Bridge AIs are still far from professional players' level. Their level is close to the level of good amateurs.

III. Adaptation of the seeds methodology to computer bridge

A. Monte-Carlo in computer bridge

In bridge AIs, 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 exists 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 take 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 double-dummy solver. 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 double-dummy analysis (DDA).

B. Chosen AI: Wbridge5

The purpose is to check whether the optimization seed method can be effective on bridge AIs. The first step consisted 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 consisted 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 AIs (two against two). Therefore, four seeds need to be chosen. In order to make the interpretation of the results easier, we have chosen 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 linked to a sub-optimal contract or a bad quality card game. 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 chose 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 will obtain 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 only 25 unknown cards at most during the card play after the lead (see section III.A). 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 used 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 AIs 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 will take 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 a DDA as explained in the following section.

IV. Experiments

The first step of experiments consisted 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.

A. Bridge data

The duplicate scoring (see section II.B) is the scoring used for official competitions both for humans and robots. The size of matches in final matches in computer bridge competitions is 64 boards. We then choose to fill in the matrix with the result (in IMPs) of 64-boards matches.

In order to compare 40 seeds, we built a 40x40 matrix R; each entry $R(i,j)$ containing the result in IMPs of a 64-boards 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-boards match with an advance of 12 IMPs.

Match conditions:

The set of boards of each match is disjoint from the set of boards of every other match.

Table 1 : two instances of WB(i) are NS and two instances of WB(j) are EO. Table 2 : two instances of WB(j) are NS and two instances of WB(i) are EO. The boards are the same at the two tables.

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 decided 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.

Remarks:

1. During the experiment, we observed 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. For a human, it is difficult to evaluate the percentage of boards where a pair has no decision to make and should only apply the convention system.

Seed	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	1	3	16	-15	26	-30	-28	-27	3	-1	21	-6	5	8	-10	24	-23	26	8
2	-1	0	12	0	15	31	-1	11	-15	4	48	-3	-6	-36	-10	-22	8	-5	-19	3
3	-3	-12	0	-15	-16	1	-30	-8	-4	-10	-25	57	24	6	-41	-1	-3	-3	-4	17
4	-16	0	15	0	33	21	27	0	2	30	11	27	-57	-21	-1	-20	33	-13	21	-6
5	15	-15	16	-33	0	-20	32	10	-40	-29	52	-50	41	7	-2	1	17	-12	12	-31
6	-26	-31	-1	-21	20	0	-38	-11	10	-60	7	-19	-23	-24	-25	-16	13	30	-53	1
7	30	1	30	-27	-32	38	0	-30	-4	-17	-28	-26	-6	0	-15	-19	-7	29	-1	40
8	28	-11	8	0	-10	11	30	0	-8	-6	10	-16	-1	0	6	1	1	7	-21	-14
9	27	15	4	-2	40	-10	4	8	0	4	6	-4	1	-4	-10	-29	-28	-27	-10	-9
10	-3	-4	10	-30	29	60	17	6	-4	0	45	2	-1	-36	5	40	3	-9	26	-2
11	1	-48	25	-11	-52	-7	28	-10	-6	-45	0	-7	-21	31	-12	21	-27	3	-3	13
12	-21	3	-57	-27	50	19	26	16	4	-2	7	0	19	-14	63	8	-7	-4	12	-14
13	6	6	-24	57	-41	23	6	1	-1	1	21	-19	0	33	17	1	4	19	3	4
14	-5	36	-6	21	-7	24	0	0	4	36	-31	14	-33	0	-2	-37	-9	-28	-43	-9
15	-8	10	41	1	2	25	15	-6	10	-5	12	-63	-17	2	0	16	-22	-40	-20	-35
16	10	22	1	20	-1	16	19	-1	29	-40	-21	-8	-1	37	-16	0	25	23	0	15
17	-24	-8	3	-33	-17	-13	7	-1	28	-3	27	7	-4	9	22	-25	0	-7	-4	0
18	23	5	3	13	12	-30	-29	-7	27	9	-3	4	-19	28	40	-23	7	0	-23	1
19	-26	19	4	-21	-12	53	1	21	10	-26	3	-12	-3	43	20	0	4	23	0	-5
20	-8	-3	-17	6	31	-1	-40	14	9	2	-13	14	-4	9	35	-15	0	-1	5	0
21	-19	17	-10	-34	-3	5	14	-1	10	19	15	-12	14	4	10	-20	-36	7	-13	-8
22	-15	-37	-21	18	41	4	-4	20	11	-9	-23	17	5	-2	-20	7	9	1	12	9
23	-42	34	-14	3	24	-6	6	-9	-3	-16	-26	22	3	20	-28	11	14	-6	-5	-3
24	3	20	30	-15	24	-1	3	0	-21	22	23	-8	-21	39	-18	-16	-8	62	35	-45
25	66	-2	20	8	15	-11	-9	-28	-34	3	-14	20	-8	-28	-2	0	5	-7	-26	-5
26	-4	-5	-30	5	-9	-28	-13	7	-16	-10	7	-2	-21	-18	-17	-23	-2	-24	-30	19
27	-11	-17	27	-19	4	-12	-1	-32	5	11	12	-15	-30	-1	4	5	13	8	29	28
28	-48	4	-17	-12	7	-15	-24	-29	-7	12	6	46	-23	39	8	-12	0	-33	-3	-7
29	-18	-24	-38	-11	-10	30	1	15	-14	-2	-2	7	-18	-36	2	-20	-12	25	-5	-35
30	9	13	-30	18	-11	37	-7	5	-16	-10	-7	15	-12	-20	-24	18	7	12	-54	25
31	-33	-10	2	-9	-29	-20	-11	-54	24	31	-19	34	11	5	-1	-9	-30	-72	31	11
32	-20	-28	-21	-12	4	-20	-19	-18	7	19	13	-11	7	-3	21	19	3	-17	-15	1
33	29	12	-24	19	-19	-27	30	-23	30	12	-10	25	-10	-18	-4	21	24	-21	-8	4
34	-2	31	-25	-3	4	-16	-30	11	-8	-4	-33	17	-17	-29	-17	-21	39	-2	-6	-7
35	8	24	-22	-17	63	-29	-3	-33	-6	10	24	-8	-21	19	-6	-10	-23	23	15	39
36	-62	28	-5	-20	-30	21	-5	1	-9	-22	36	24	-4	-16	24	26	-24	-4	15	10
37	22	-2	8	-1	22	0	4	-1	37	1	12	7	2	30	-23	-32	-26	22	-5	5
38	22	-29	40	-27	-30	-44	-7	-53	-1	4	-5	-12	-2	19	-6	22	-19	-29	-1	-18
39	-3	-5	-3	28	36	0	32	6	18	-21	-15	30	32	10	4	34	-37	-9	35	2
40	-37	18	8	0	38	53	15	-15	-1	10	-10	-2	-36	-16	-2	19	13	39	9	-23

Seed	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	Cumulation
1	19	15	42	-3	-66	4	11	48	18	-9	33	20	-29	2	-8	62	-22	-22	3	37	156
2	-17	37	-34	-20	2	5	17	-4	24	-13	10	28	-12	-31	-24	-28	2	29	5	-18	-28
3	10	21	14	-30	-20	30	-27	17	38	30	-2	21	24	25	22	5	-8	-40	3	-8	55
4	34	-18	-3	15	-8	-5	19	12	11	-18	9	12	-19	3	17	20	1	27	-28	0	167
5	3	-41	-24	-24	-15	9	-4	-7	10	11	29	-4	19	-4	-63	30	-22	30	-36	-38	-170
6	-5	-4	6	1	11	28	12	15	-30	-37	20	20	27	16	29	-21	0	44	0	-53	-188
7	-14	4	-6	-3	9	13	1	24	-1	7	11	19	-30	30	3	5	-4	7	-32	-15	-16
8	1	-20	9	0	28	-7	32	29	-15	-5	54	18	23	-11	33	-1	1	53	-6	15	246
9	-10	-11	3	21	34	16	-5	7	14	16	-24	-7	-30	8	6	9	-37	1	-18	1	-30
10	-19	9	16	-22	-3	10	-11	-12	2	10	-31	-19	-12	4	-10	22	-1	-4	21	-10	94
11	-15	23	26	-23	14	-7	-12	-6	2	7	19	-13	10	33	-24	-36	-12	5	15	10	-111
12	12	-17	-22	8	-20	2	15	-46	-7	-15	-34	11	-25	-17	8	-24	-7	12	-30	2	-113
13	-14	-5	-3	21	8	21	30	23	18	12	-11	-7	10	17	21	4	-2	2	-32	36	266
14	-4	2	-20	-39	28	18	1	-39	36	20	-5	3	18	29	-19	16	-30	-19	-10	16	-73
15	-10	20	28	18	2	17	-4	-8	-2	24	1	-21	4	17	6	-24	23	6	-4	2	13
16	20	-7	-11	16	0	23	-5	12	20	-18	9	-19	-21	21	10	-26	32	-22	-34	-19	110
17	36	-9	-14	8	-5	2	-13	0	12	-7	30	-3	-24	-39	23	24	26	19	37	-13	54
18	-7	-1	6	-62	7	24	-8	33	-25	-12	72	17	21	2	-23	4	-22	29	9	-39	63
19	13	-12	5	-35	26	30	-29	3	5	54	-31	15	8	6	-15	-15	5	1	-35	-9	86
20	8	-9	3	45	5	-19	-28	7	35	-25	-11	-1	-4	7	-39	-10	-5	18	-2	23	21
21	0	15	-6	3	33	11	12	15	-11	37	-61	-16	2	-31	-9	24	-11	-2	-13	-4	-53
22	-15	0	-27	-26	-15	13	-30	-18	-1	-49	1	-21	-48	-13	-29	-6	-35	28	15	-10	-263
23	6	27	0	-22	10	18	-10	24	6	-33	25	41	-15	9	7	-29	-24	3	17	-9	30
24	-3	26	22	0	2	2	44	-5	-24	-22	11	-37	42	-14	4	-25	3	-5	2	-1	130
25	-33	15	-10	-2	0	5	-21	-17	-3	6	-10	16	-30	3	-2	10	4	23	17	4	-62
26	-11	-13	-18	-2	-5	0	40	5	-4	17	26	12	13	-18	-2	25	-6	-5	-3	1	-162
27	-12	30	10	-44	21	-40	0	-1	2	3	23	-44	23	-15	-13	-13	-13	-1	-9	28	-57
28	-15	18	-24	5	17	-5	1	0	35	-1	39	9	-1	-6	-8	18	1	-14	9	10	-20
29	11	1	-6	24	3	4	-2	-35	0	-17	-21	16	-2	21	29	-27	-20	8	-24	29	-173
30	-37	49	33	22	-6	-17	-3	1	17	0	-31	-8	19	-11	50	-10	16	-26	-27	-4	-5
31	61	-1	-25	-11	10	-26	-23	-39	21	31	0	6	-17	37	42	4	-18	17	5	-7	-81
32	16	21	-41	37	-16	-12	44	-9	-16	8	-6	0	-3	29	9	39	15	-14	13	33	57
33	-2	48	15	-42	30	-13	-23	1	2	-19	17	3	0	-1	-15	4	24	28	-11	4	92
34	31	13	-9	14	-3	18	15	6	-21	11	-37	-29	1	0	23	-7	-3	-11	-44	-1	-151
35	9	29	-7	-4	2	2	13	8	-29	-50	-42	-9	15	-23	0	-11	33	21	-19	-10	-25
36	-24	6	29	25	-10	-25	13	-18	27	10	-4	-39	-4	7	11	0	4	7	-39	-4	-44
37	11	35	24	-3	-4	6	13	-1	20	-16	18	-15	-24	3	-33	-4	0	14	-18	10	118
38	2	-28	-3	5	-23	5	1	14	-8	26	-17	14	-28	11	-21	-7	-14	0	12	1	-234
39	13	-15	-17	-2	-17	3	9	-9	24	27	-5	-13	11	44	19	39	18	-12	0	-22	269
40	4	10	9	1	-4	-1	-28	-10	-29	4	7	-33	-4	1	10	4	-10	-1	22	0	32

Fig. 5: 40x40 matrix R; each entry R(i,j) containing the result in IMPs of a 64-boards match between WB(i) and WB(j).

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 AIs.

Results: 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 (-263) and the best seed is the seed 39 (+269). The complete matrix is represented in Fig. 5.

There are several competing approaches in the seeds methodologies[1], [2], [3], [4], including BestSeed and Nash. The BestSeed approach usually provides greater success rates against the original AI tan 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. Given the intrinsic randomization of bridge, we assume that taking care of learning opponents was not necessary, therefore we use BestSeed allowing use to choose the seed 39 with +269 cumulated IMPs.

In the two following sections, we present some results related to the performance of the best seed found. The first approach presented in section IV.B is a cross-validation one on the current matrix. The second approach presented in section IV.C consisted in empirically testing the performance of the best seed over new 1000-boards matches.

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

For each matrix size we compare the performance of the best seed (resp. the version robustified by using the best X% seeds) against a random seed. This is tested, in this section, on the 40x40 matrix mentioned above. The x-axis in Fig. 6 represents the size of a randomly known submatrix, while the y-axis represents the average performance, in imps, of the best seed (resp. seed distribution, for the best X% version) selected

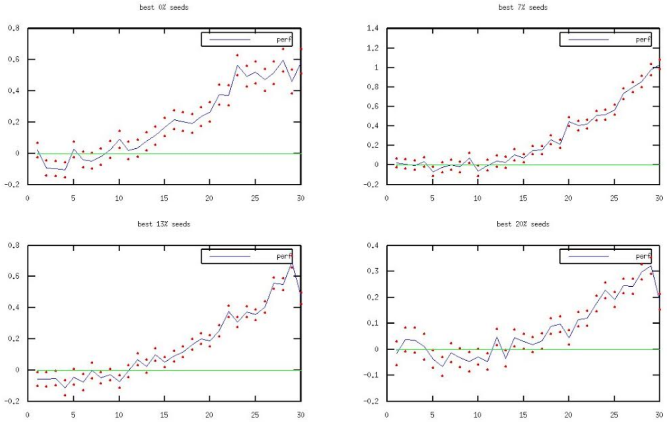

Fig. 6: Performance of the best seed and the X% best seed approach, in Imp/game.

on this submatrix against the remaining seeds. Importantly, we properly apply cross-validation - the performance is tested in parts the matrix which have not been seen.

C. Validation according to empirical results out of the matrix

We tested the best seed found in the matrix on 1000-boards matches. As previous, scores are computing using the double-dummy solver (see section 4.A).

We compared the deterministic program with the best seed (seed 39) and the worst (seed 22). The program with the best seed won with a difference of 163$\pm$ 2.62 IMps, over 1000 games, i.e. 0.16 imp per board.

The second match was between the program with the best Seed (39) and the program with the current seed (99) which had been chosen arbitrarily in the past of WBridge5. The program with the best seed won with a difference of 97 $\pm$ 2.75 Imps on 1000 boards. This means an improvement of $\simeq$0.1 imp per board.

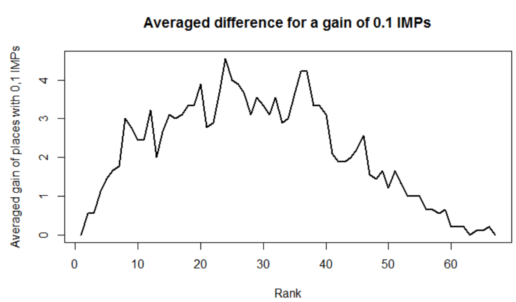

Fig. 7: Influence of adding 0.1 imp on the rank of players, out of 67 ranks.

Without going into further details it is important to point out that a gain of 0.1 imps per board is a great improvement for bridge players. During international competitions, the participants are ranked in a Butler, which gives the averaged number of imps won or lost by each pair per board. In Figure 7 the y-axis represents the number of places won by a pair if 0.1 imps are added to its original score on the whole championship, in function of the x-axis giving the original rank on 67 pairs. Data has been found on the WBF site (http://www.worldbridge.org/) from last World and European bridge championships.

V. World Computer-Bridge Championship

Given experimental results (see above for the 1000 games validation we have made, ensuring a significant performance improvement compared to the original WBridge5), Yves Costel decided to use the boosted version for the Wbridge5 participation to the 20th World Computer-Bridge Championship.

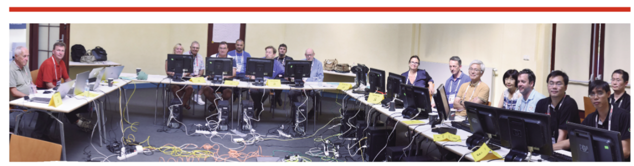

Fig. 8: World Computer-Bridge Championship in Wroclaw, september 2016.

The 20^{th} World Computer-Bridge Championship was held on September 10-15, alongside the World Bridge Games in Wroclaw, Poland. It was organized 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 current the world champion Jack (Netherlands), the come-back of Meadowlark Bridge and the new entry of the robot of Xinrui which is a serious opponent since Xinrui is an entreprise dedicated to computer bridge founded by Yuzhang Liu with a team of 20 persons.

In the competition, there is a qualification stage called Round-Robin where every AI plays against each other in matches of 32 boards. 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 IMP 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 finished first of the Round-Robin with a very little lead over the second and third (Wbridge5 91.87 VPs, 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).

Meadowlark after 12 years of absence had a lot of technical and bridge related issues. On the other hand, Xinrui, the newcomer, made a great start. It 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 were also very tight. Since it is a KO phase, scores are in IMPs. Wbridge5 won 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. In the other semifinal, Micro-Bridge has beaten Bridge Baron with a fewer difference : 144-138 IMPs while after the first half Bridge Baron was winning by 66 Imps (96-30).

The final was as tight as the semifinals. Before the last two boards, Wbridge5 was losing by 17 Imps against Micro Bridge. However, Wbridge5 won 23 Imps in the last two boards for a final score of 162 to 156. All over the competition, the AI made the difference by finding a better contract than its opponent. As we saw during our experiments, it seems that the seed has an impact on difficult decisions like these ones. 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.

Finally, Wbridge5 won its fourth world champion title, eight years after its last victory.

VI. Conclusions

A large part of the current research effort in card games uses deep learning (e.g. [13] for Poker), with clear successes (Libratus, artificial intelligence developed by Carnegie Mellon University, has beaten four of the best heads-up no-limit Texas hold'em poker players in a 20-day competition); this has reached Bridge (see the recent [14], [15] and the older [16]). In [15], a deep reinforcement learning model has been designed in order to achieve automatic bid learning task in a subproblem of bidding called non competitive sequence biddings. This kind of biddings sequence occurs in only 27 % of boards (statistics obtained from the data on the WBF site). Besides, the subproblem is related to a two-players process rather than to a four-player one. 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.

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, previsibility is not a problem for bridge. We therefore used and tested the BestSeed approach which is easily adaptable to games with more than 2 players. 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 20^{th} World Computer-Bridge Championship; whereas it had not won this competition since 2008.

Work in progress and Further Work

A flurry of experiments on wbridge5 currently in progress aim to extend the framework. First at all, since BestSeed is quite expensive, we begin to extend our work by using an advanced bandit methodology rather than the brute force version used in the present work. The gain of time will allow us to make more precise experiments in order to find best specific seeds for different situations in the bidding depending on the height of the contract (e.g attempts to 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.

References

[1] D. L. St-Pierre and O. Teytaud, "The Nash and the bandit approaches for adversarial portfolios," in *Computational Intelligence and Games (CIG), 2014 IEEE Conference on*. IEEE, 2014, pp. 1–7.

[2] T. Cazenave, J. Liu, and O. Teytaud, "The rectangular seeds of domineering," in *2015 IEEE Conference on Computational Intelligence and Games (CIG)*, Aug 2015, pp. 530–531.

[3] T. Cazenave, J. Liu, F. Teytaud, and O. Teytaud, "Learning opening books in partially observable games: using random seeds in phantom go," *arXiv preprint arXiv:1607.02431*, 2016.

[4] J. Liu, O. Teytaud, and T. Cazenave, "Fast seed-learning algorithms for games," in *International Conference on Computers and Games*. Springer, 2016, pp. 58–70.

[5] E. R. Berlekamp, "Program for double-dummy bridge problemsa new strategy for mechanical game playing," *Journal of the ACM (JACM)*, vol. 10, no. 3, pp. 357–364, 1963.

[6] A. I. Wasserman, "Realization of a skillful bridge bidding program," in *Proceedings of the November 17-19, 1970, fall joint computer conference*. ACM, 1970, pp. 433–444.

[7] S. J. Smith, D. S. Nau, and T. A. Throop, "Total-order multi-agent task-network planning for contract bridge," in *AAAI/IAAI, Vol. 1*. Citeseer, 1996, pp. 108–113.

[8] M. Ginsberg, "Partition search," *Proceedings Of The National Conference On Artificial Intelligence*, 1996.

[9] M. L. Ginsberg, "Gib: Imperfect information in a computationally challenging game," *Journal of Artificial Intelligence Research*, vol. 14, pp. 303–358, 2001.

[10] A. Amit and S. Markovitch, "Learning to bid in bridge," *Machine Learning*, vol. 63, no. 3, pp. 287–327, 2006.

[11] L. L. DeLooze and J. Downey, "Bridge bidding with imperfect information," in *Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on*. IEEE, 2007, pp. 368–373.

[12] M. Paul, "Bethe," *The state of automated bridge play*, 2010.

[13] M. Moravčík, M. Schmid, N. Burch, V. Lisỳ, D. Morrill, N. Bard, T. Davis, K. Waugh, M. Johanson, and M. Bowling, "Deepstack: Expert-level artificial intelligence in no-limit poker," *arXiv preprint arXiv:1701.01724*, 2017.

[14] C.-Y. Ho and H.-T. 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.

[15] C.-K. Yeh and H.-T. Lin, "Automatic bridge bidding using deep reinforcement learning," *arXiv preprint arXiv:1607.03290*, 2016.

[16] B. Yegnanarayana, D. Khemani, and M. Sarkar, "Neural networks for contract bridge bidding," *Sadhana*, vol. 21, no. 3, pp. 395–413, 1996.