

# Bluffing computer? Computer strategies to the Perudo game

Norbert BOROS  
Széchenyi University, Győr  
email: [bnorby@sze.hu](mailto:bnorby@sze.hu)

Gábor KALLÓS  
Széchenyi University, Győr  
email: [kallós@sze.hu](mailto:kallós@sze.hu)

**Abstract.** In this paper we present the elaborated computer strategies which we developed to a less-known dice game, Perudo. Moreover, the related analysis is presented, with the results of our machine players against human players. The main contribution of our paper is a strategy which involves bluffing. An important goal of this work is the remembrance to John v. Neumann; with our means we pay a tribute to the activity of the "founding father"; on the occasion of his 110<sup>th</sup> birthday.

## 1 Computer and human thinking

At the time of designing the first electronic calculators it came up as a straightforward possibility to try to copy/build in some—that time already known—modus operandi of the human brain (John v. Neumann et al., about 1946). To make the process operational it was necessary to prepare a simplified model of the brain and to analyse several very important fields like data storage (capacity), data transfer, arithmetic precision, and error-free operation (in computers and in brain); in wider meaning similarities and differences in human and computer thinking, respectively.

John v. Neumann turned to this field from the 1940s with special interest. He presented in this topic several memorable talks and published valuable

---

**Computing Classification System 1998:** I.2.1

**Mathematics Subject Classification 2010:** 68T20

**Key words and phrases:** games, machine strategies, bluff, Perudo

**DOI:**10.2478/ausi-2014-0018

papers. Unfortunately—because of his serious illness and early death—he was not able to finish one of his most interesting books (The Computer and the Brain [8]), but this work is even so extremely remarkable and can serve as a source of interesting ideas even for today’s researchers.

The observations can be summarized in brief as follows. The human brain—contrary to computers—does not use the “language of mathematics”, its operation is not digital, but essentially statistical; its important characteristic is the lower arithmetic accuracy, and extremely high logical reliability ([8], 1959).

The examination of similarities and differences in human and computer thinking is even today an actively investigated multidisciplinary top-field. In this “front” an interested researcher has obvious opportunities. In this paper within the main trend we “venture” to a smaller branch—which is proved to be very interesting, however—we present the strategies (and the connected analysis) which were elaborated to a less-known dice game, Perudo (first related publication by the first author: [2]).

## 2 Games and strategies

Playfulness has been always a characteristic of human nature. Challenge, victory, competition, tempting the fortune—these are factors which affect every human. We can take it for granted that for achievement of victory an interested, thinking player can take some theoretical considerations—over and above the given game—, and of course, it was the situation even for centuries/thousand years.

In this work we have not intended to describe the theory of games in detail; we always present the necessary background to the chosen game(s) (and to the problems which have actually arisen). For the readers interested in game theory deeply a rich literature is available (e.g.: [5, 7]).

The world of the game-area—separated partly from the reality of the real world—can be generally characterized as follows. A game “consists of” players, rules and results [6]. Usually we can assume that keeping the rules is obligatory for every actor (i.e. deception is not allowed); and also that players know the rules. The positive result (victory, winning the round or other favourable outcome) involves for the players winning (money, token, other real or fictitious objects); this is called simply *payoff*. So, the goal of the game is to reach the most favourable payoff.

By evaluating the results the players—according to their knowledge/skills—

set up a value order among the possible outcomes foreseen (step by step, on longer run or even globally) and make decisions in this way. When a player makes such a plan which prescribes the answer in every decision situation, then we talk about *strategy*.

Formally, a game can be identified by a vector (or list)  $G = (S_1, S_2, \dots, S_n; \varphi_1, \varphi_2, \dots, \varphi_n)$ , where  $n$  is the number of players,  $S_1, S_2, \dots, S_n$  denote the strategy-sets of the players, and  $\varphi_1, \varphi_2, \dots, \varphi_n$  are the payoff functions—here the list above is called as the normal form representation [5]. The domain of the payoff function is  $S_1 \times S_2 \times \dots \times S_n$ , its range is a certain subset of real numbers, so  $\varphi_i(x_1, x_2, \dots, x_n)$  can be computed assuming that all players have a fixed strategy. (It should be noted that a player usually does not know the strategy of the others.)

The best possible strategy is the *winning strategy*. If in a particular game such a strategy exists, and the player who moves first knows it and adheres to it, then—regardless of the moves of the other players—he/she will always win.<sup>1</sup> The other extremity is given in games where we have no information about the will of the other players; in this case obviously we cannot perform any analysis. An example of such a game is rock-paper-scissors; in this case we can trust essentially only in luck. Card games represent an interesting middle course, where information is usually not complete, but we have possibility for certain deductions.

While a lot of games can be described as deterministic processes, and can be evaluated so, for economic processes this property usually does not hold. The optimal solution here depends mostly on a series of external factors which cannot be necessarily foreseen and we have to apply here certain simplifications in order to reach the good resolution [3]. Essentially, this is the main problem of the game theory. The game theory science was established by John v. Neumann, who realized the minimax theorem by analysing the two-player zero-sum games, according to which with a reasonable game style of the players, one actor always gets an average  $v$  winning. The fact that game theory is widely applicable was made clear by his book published with Oskar Morgenstern [6].<sup>2 3</sup>

---

<sup>1</sup>We note that in many "complicated" deterministic classic games the winning strategy is usually not known or—based on our knowledge for the time being—we are not able even to decide if such exists [3] (chess, chequers).

<sup>2</sup>An excellent round-up was presented about the activity of John v. Neumann by L. Á. Kóczy in [6].

<sup>3</sup>We note that in connection with the 2008 financial crisis it was heard from many experts that economic game theory is essentially unusable. This is, of course, a strong exaggeration,

### 3 Classification of games

Analysing well-known games we find that devising a strategy good enough is usually not an easy task. At the beginning of this process we have to answer e.g. the following questions:

- Do we know completely the standing of the other players, their move or choice, and objective circumstances of their decisions?
- Does luck play a role in the given game?
- Is the collaboration of the players allowed?

We know that many times in the deterministic case the extremely complicated game-space causes a big problem; while in games which comprise random elements we cannot reach even theoretically the complete mastery (however, some additional knowledge can help even in this case).

By an extensive analysis of the factors presented above, we can group the games into the following categories (here we used the book [5] as a source):

- Strategy game: fortune plays no role in the game, the decisions of the players can be analysed in a fully deterministic way (example: chess, noughts and crosses);
- Game of chance (gamble): the result of the game is determined by pure accidental factors; these games can be managed with the means of probability theory (example: lottery);
- Mixed game (partly strategy, partly gamble): between the two categories listed above (example: poker);
- Zero-sum game: a given player can increase his/her winnings only to the detriment of the other players (example: nim, noughts and crosses);
- Non zero-sum game: players can use even external sources to increase their winnings (example: roulette);
- Cooperative game: players can cooperate in coalitions to reach a common goal (example: monopoly);
- Non-cooperative game: keeping the game-rules players have no possibility to cooperate (example: chess);
- Game with complete information: players see the standing and the moves of the opponent(s) exactly, and—assuming appropriate game skills and knowledge—they can conclude precisely the reasons of the decisions of the other actor(s) (namely the payoff function of every player is a common knowledge for all players; example: chess);

---

but it is a fact where lessons should be deducted.

- Game with incomplete information: for one or more player(s) there exists some kind of uncertainty about the payoff function of the other players—standing, strategy (example: poker, rock-paper-scissors);
- Static game: the strategy of the players is fixed, does not change during the game (example: lottery);
- Dynamic game: the momentary strategy of the players is an element of a big strategy-set, the choice is determined by the current game-situation (example: poker, chess).

The classification according to the categories—with possible connections—is presented in Figure 1.

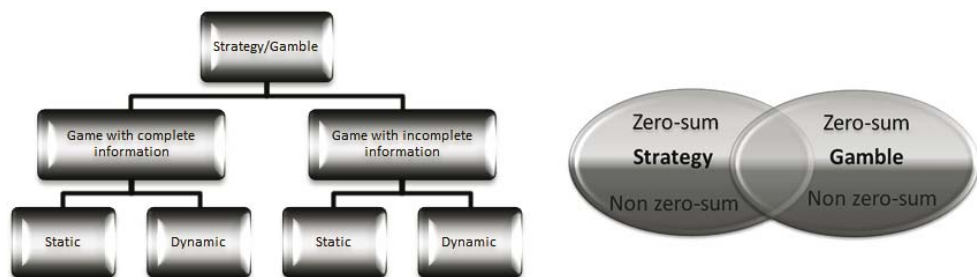


Figure 1: Classification of the games

## 4 Analysis of the game of poker

In the following—according to our game, Perudo—we will be interested in games located on the borderline of strategy games and gambles which are zero-sum, non-cooperative and dynamic with incomplete information. We will use poker as a kind of standard which is similar to Perudo in several aspects, but is better known and much more analyzed.

John v. Neumann liked to play poker himself, and he was deeply interested in finding the way of reaching victory with a good chance—regardless of dealing luck which cannot be influenced assuming a fair game style [6]. Achieving this, bluff can help most of all and the understanding of the connected human behaviour. He tried to describe the bluff methods with the means of mathematics and later he generalized this analysis into several directions (game theory, machine and human thinking, see the introduction section).

Poker is a game which is even nowadays very actively analysed—because its goal is essentially "making money". It is thought-provoking that some kinds

of people are in this game more successful and the winners of big competitions come from a well-limited group. About the theory of poker David Sklansky published a "bible-like" detailed writing [9] in which—among others—he presents the mathematical background and the "fundamental theorem of poker" (not a real theorem, principle; it refers to strategy based gambles, too), the good game strategies, and the possibilities of recognition and evasion of bluff. The book was written mostly for professional players, it is not an easy reading; but shows the enquiring reader ins and outs of the game with a method of a high standard, trying to reach completeness.

In order to algorithmize the game of poker various approaches were applied; adaptive learning-, probability-, deterministic rule-based- and balanced equilibrium methods. The theory of the machine techniques is presented in detail by Darse Billings in his interesting PhD thesis [1]. However, each method has its shortcomings, too. In summary, we can say that for the time being there is no chance for the best machine players to beat the best human players (comparing with chess it can be a bit surprising).

For the implementation of the machine strategies for Perudo we took into account mostly probability rules. It is important to emphasize that although the algorithmization of Perudo (variants) was dealt with by other programmers already (example: online Perudo, <http://www.perudo.com>), but in these realizations machine strategies *have not been created* (they have made online games, where people can play against each other). So to the best of our knowledge, the results published in this paper are new (stand: July 2013).

## 5 Presentation of the game of Perudo

The game of Perudo—as we know it today—derives from the pre-Columbian South-America and was very popular among the Inca Indians. According to tradition, emperor Atahualpa had taught Pizarro to play Perudo (during the period of his imprisonment in 1532), and after it the game spread in Europe.<sup>4</sup> Perudo can be characterized as the exciting game of logic, bluff and guess.

The game can be played by two or more players. *Initially*, all of the players have 5 dices (under a cup). The game consists of *rounds*, and at the end of each round some of the players lose one dice. *Finally*, the play is won by the player, who has at least one dice at the end.

---

<sup>4</sup>We note that Perudo has several versions. Here we highlighted one version of them.

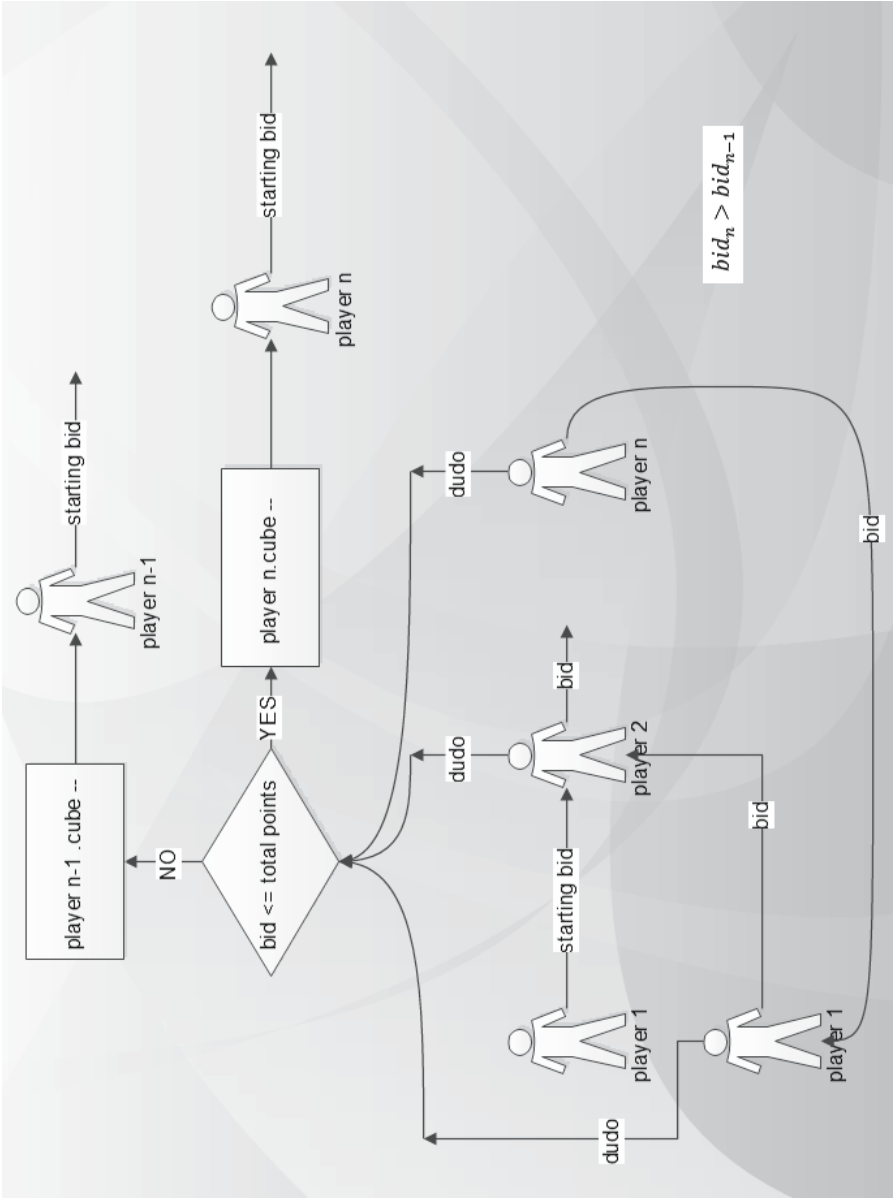


Figure 2: The course of the game of Perudo (without joker bid)

In one round the players—hidden from each other—roll their dices and then begins the continuous bid. Bidding consists of two parts; the number and the value of the dice belonging to it. The player who moves first estimates at least how many dices are on the table with that value. The number of ones is added to the number of the pieces, because they are counted as *joker*. The value of the bid is determined by the following formula:

$$L = \text{number of the dice}(s) \cdot 10 + \text{value of the dice}(s).$$

The next player has two choices: he/she can overcall the bid (at such time he/she can bluff, too), or says "dudo" (in Spanish: I doubt)—in this case he/she considers the former bid as a not real one. By the call of dudo, players reveal their dices and determine the number of the dices with the given value (including jokers). If the last bid was valid, then the player saying dudo has to throw away one of his/her dices, while if the player with the last bid bluffed then he/she loses one dice (Figure 2).

It is a twist in the game that the player whose turn it is can call a joker bid, too. At such time he/she estimates the number of jokers in the play currently (of course with this he/she can bluff, too). In a joker bid at least half of the number of pieces appearing in the last bid must be called (rounded upwards). The possibilities for the next player are: he/she can say dudo, can increase the number of jokers in the bid, or can return to normal bid (in this case the number of dices in the bid must be doubled and at least one additional dice must be added). Using formulas (for the number of the dices, ND):

$$\text{Joker bid}_{\text{ND}} \geq \lceil \text{Normal bid}_{\text{ND}} / 2 \rceil, \text{ and } \text{Normal bid}_{\text{ND}} \geq \text{Joker bid}_{\text{ND}} \cdot 2 + 1.$$

## 6 The machine strategies implemented

It was our emphasized goal to create such machine strategies which possibly can cope successfully with human players, too. The realization was accomplished step by step, from the simplest strategies toward the more sophisticated ones. Machine and programming environment: table PC with Intel Core 2 Duo CPU and 4 GB RAM, Windows operating system, Netbeans integrated development environment, JAVA language.

The first programmed method was named as basic strategy. This strategy decides and says bid on the basis of analysing expected values. The strategy has an adjustable parameter, the tolerance, with which we can set the upper bound—exceeding the expected value—acceptable in bids. (In this case joker bid is said.)





Figure 3: The starting screen of the program

```
decision(bid_got)
```

```
1 expected = expected_value(bid_dice_value_got)
```

```
2 if (bid_dice_number_got < expected)
```

```
3     bid() // lines 2–3: normal bid
```

```
4 else if (bid_dice_number_got <= expected + tolerance)
```

```
5     joker_bid() // lines 4–5: joker bid
```

```
6 else
```

```
7     dudo()
```

Following David Sklansky, the good player plays so that he/she would know the cards/dices of the other player(s)/opponent(s) [9]. In the extended strategy—based on basic strategy—this idea was applied: the code was supplemented with a method which deduces back—from the initial bids—to the dices of the players and stores them. The effect of this deduction resulted that the extended strategy has won—using a 100 thousand-round test—6-times more rounds than the basic strategy (Figure 4).

The reason for this is that basic strategy tried saying dudo roughly for half of the times as the opponent, and almost the half of its trials were unsuccessful; whilst the extended one said many times successful dudo, than unsuccessful.

The *foreseeing strategy* accomplished thirdly, —similarly to the extended one—has a `bidLoad()` method using which it keeps the records of dices of the

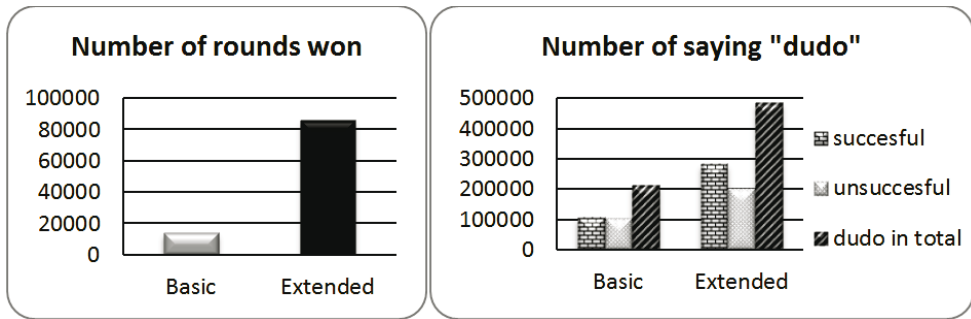


Figure 4: Basic vs. extended strategy

other players. However, by the deduction it takes into account all bids, so, it can calculate more exactly.

```

bidLoad(bid[ ], number_of_bids)
1 player[bid[0].player_id][bid[0].dice_value - 1]
  += number_of_dices(starting_bid)
2 for(i = 1; i < number_of_bids; i++)
3 player[bid[i].player_id][bid[i].dice_value - 1]
  += number_of_dices(bid[i], bid[i - 1])

```

It is true for the foreseeing strategy, too, that—due to the more refined deduction—it plays better than the basic one. Interestingly, the basic strategy against the extended one said only a little bit fewer successful dudos than unsuccessful ones, on the other hand, against the foreseeing strategy the number of saying unsuccessful dudos was definitely more (Figure 5).

Strategies were tested in various situations because we wanted to know whether the result of the game is influenced by such factors as which strategy starts the bid, how many times they can initiate a joker bid, or how the players sit side by side (seating), respectively. The modification of the first two factors had brought no change; however, we got surprising a result by altering the seating.

In the following the result of two complex simulations will be presented. In both examinations five machine players took part, four basic and one foreseeing. In the first case the seating changed randomly, in the second case it was fixed. In both simulations the foreseeing strategy won mostly, but the winning-distribution of the basic machines is interesting.

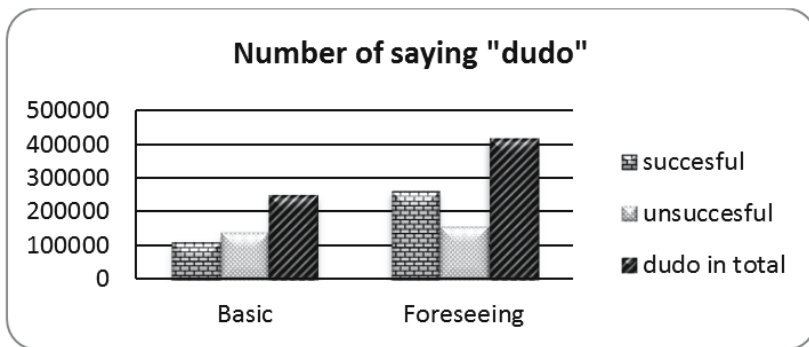


Figure 5: Basic strategy vs. the foreseeing one

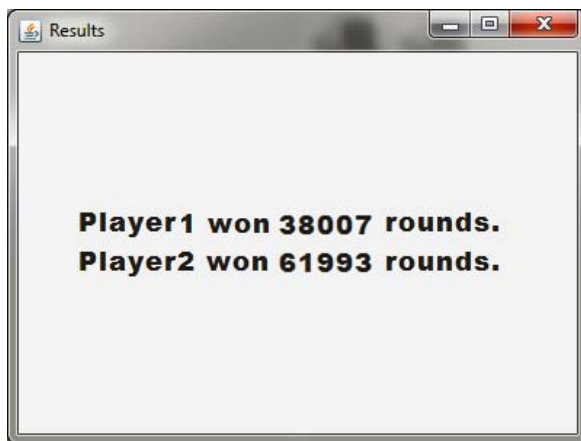


Figure 6: Displaying the result of a machine test (duel of strategies, program window)

With fixed seating—gradually—the basic machine following the foreseeing one won most often, and after it each less and less; the basic strategy right before the foreseeing one the least. The reason for this is that the foreseeing machine tends to say dudo at the right time to the strategy right before it, and the basic machines before the foreseeing say—often—at the wrong time dudo to the foreseeing strategy. So, the basic machine right before the foreseeing one drops out of the competition first. With random seating every basic machine won roughly the same number of rounds—this meets the expectations (Figure 7).

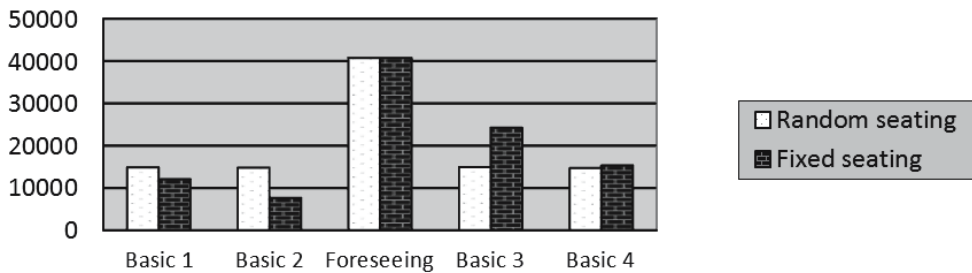


Figure 7: Basic machines vs. the foreseeing one: winnings with fixed and changing seating

After this, the tests with human players were organized. In the course of the trials 3 volunteers played 15 rounds against the machine players. The scenario of the games was similar: at the beginning almost exclusively the machine had won, later this equalized and slowly the human players gained the dominance. The reason of this phenomenon is that human players came to know the machine players. All three participants of the trial found out how the machine gave the bid and exploiting this they bluffed. The machine players played rationally, they believed the bluff and so finally lost.

## 7 Bluffing strategy

The parties against the human players have brought the clear experience that in certain situations it is impossible to win with rational behaviour; so our machine players—accomplished so far—has had a serious disability. As a "theorem", we can draw up that a good player has to be unrecognizable, because if he/she decides to always use the same strategy, then after a while the others saw through it and defeated him/her.

In the Perudo game bluff is an important factor since without it we would have essentially a simple gamble which would be won by the player who throws larger. Bluff is a form of inscrutableness, the "allowed lie" in the game, the tool of the deception of the opponent.<sup>5</sup> To the success of deception it is indispensably important to be disguised: in the behaviour of the player the "deceitful purpose" should not be visible. In the case of human players bluff should be said with the same poker face as the other bids; besides this a good player has

<sup>5</sup>As it was already mentioned, John v. Neumann came to the mathematical establishment of the game theory from the analysis of the bluff.

to be able to detect the deception of the others. The one, who wants to be really successful in this "genre", is worth getting to know thoroughly the psychological background, too: which are the usual telling/revealing characteristic of the lie [4].

Bluff must be set up by following a strategy, too (when and how we should bluff; concerning the poker game: [9]), winning is not for one game, but in the long run. Greedy strategy is not practical here: if a player always or almost always bluffs then although in the first few rounds he/she will win with better chance, but later when the opponents see through his/her habits, it is almost sure that he/she will lose. Similarly to poker—possibly—it is practical to make an analysis about the playmates: who in what manner reacts to the bluff, how credulous or doubting he/she is.

In machine environment clearly it should be implemented otherwise (at least partly) since—taking into account our current possibilities—considering the behaviour, gestures and facial features bluff should not be found out because a computer cannot recognize these, and such manifestations computer cannot own.

By creating the *bluffing* strategy, our goal was to evolve such an algorithm which can be efficient against both of the credulous and doubting player, too. To make itself less predictable, strategy "builds" from several elements. With bluffing it considers not always its best theoretical bid: it can happen that it bids with a dice from which it has minimum or just sends the value from which it has maximum, but playing out not all of them. Occasionally it can transmit a bid generated randomly which may not exist. In non-existing bids, however, we made sure it should be believable to every opponent (strategy), i.e. based upon probabilities theoretically it could be on the table.

By forming of the bids we have applied case separation, elaborating scenarios for various situations. Such cases are, for example, when the player or the opponent has only one dice; or the adversary has a lot of dices, but have told only a one-valued bid, and so on.

Similarly to the extended and foreseeing strategies, the bluffing strategy analyses the bids of the opponents and tries to deduce back to their dices. However, it takes the deduction not for sure (assuming that the other players can bluff, too). According to the current realization the strategy is efficient only in the case when it has only one opponent, but we are planning to implement the multi-opponent version, too.

With the bluffing strategy we performed three 100 000-round tests, against the basic, extended and foreseeing strategies. The following result was arrived at: the new method was successful against all of the older ones, however, in

varying degrees; it won the most parties against the foreseeing strategy, and the minimum against the basic strategy (Figure 8)!

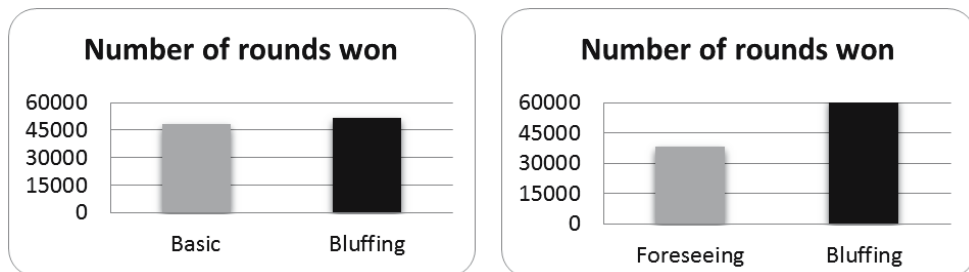


Figure 8: The bluffing strategy vs. the basic and foreseeing strategies ("credulous" and "doubting" opponents)

The reason for this—maybe surprising—result can be that foreseeing strategy deduces back the better way to the dices of the opponents, that's why it can be cheated the most easily (so, the foreseeing machine can be considered as a "credulous" player). Basic strategy performs no such analysis, therefore it is not gullible (so, it can be counted as a "doubting" player). Analysis of course helps without bluff, too: that's why the bluffing strategy could won in majority against the "doubting" basic strategy.

The bluffing strategy was tested against human players, too. Several rounds were played with three human players—the Perudo skills of whom could be classified at average level. Now we cannot deduce how the human players came to know the machine player. Results are rather scattered, almost random (Table 1); definite superiority is not visible from either slide—so, the *bluffing strategy plays with a similar efficiency as a typical man/woman*.

| Player | Winner              |
|--------|---------------------|
| 1      | HHM MHM MMH MHM HMM |
| 2      | MMM HMH MMH MHH MHM |
| 3      | MHH MHH HHM HMM HHH |

Table 1: Bluffing machine strategy (M) vs. human (H) players

We can conclude that predictability was removed from the machine play with this strategy, but still, our bluffing strategy—implemented so far—cannot be classified as a really successful and "clever" player.

We continue the work: our goal is to implement such a machine player that will be able to defeat clearly an average human player, and with the hope of success can enter the fight with real professional human players, too.

## Acknowledgements

We thank Gábor Takács, our excellent colleague, who contributed significantly with valuable advices and useful comments to the writing of this paper.

## References

- [1] D. Billings, *Algorithms and assessment in computer poker*, PhD Thesis, Alberta University, 2006.  $\Rightarrow$  61
- [2] N. Boros, *Machine strategies in the game of Perudo*, Univ. Thesis (in Hungarian; supervisor: Gábor Takács), Széchenyi University, Győr, 2009.  $\Rightarrow$  57
- [3] M. Eigen, R. Winkler, *Das Spiel – Naturgesetze steuern den Zufall*, R. Piper & Co. Verlag, München, 1975.  $\Rightarrow$  58
- [4] A. Furnham, *50 Psychology Ideas You Really Need to Know*, Quercus Publishing, London, 2009.  $\Rightarrow$  68
- [5] R. Gibbons, *A Primer in Game Theory*, Pearson Acad. Publisher, Harlow, 1992.  $\Rightarrow$  57, 58, 59
- [6] L. Á. Kóczy, A Neumann-féle játékelmélet, *Közgazdasági Szemle* **53**, 1 (2006) 31–45.  $\Rightarrow$  57, 58, 60
- [7] S. Molnár, F. Szidarovszky, *Game Theory* (in Hungarian), ComputerBooks, Budapest, 2011.  $\Rightarrow$  57
- [8] J. von Neumann, *The Computer and the Brain*, Yale Univ. Press, York, PA, 1959.  $\Rightarrow$  57
- [9] D. Sklansky, *The Theory of Poker*, Creel Printing Co., Las Vegas, NV, 2007.  $\Rightarrow$  61, 64, 68

*Received: August 28, 2013 • Revised: March 24, 2014*