Any reason for using Computational Intelligence methods in classical mind board games?

Jacek Mańdziuk

Abstract—In the last two decades the advancement of AI/CI methods in classical board and card games (such as Chess, Checkers, Othello, Go, Poker, Bridge, ...) has been enormous. In nearly all “world famous” games the humans have been decisively conquered by the machines (actually Go remains almost the last redoubt of human supremacy). In the above perspective the natural question that comes to our minds is whether there is still any need for further development of CI methods in this area. What kind of goals can be achieved on this path? What are (if any) the challenging problems in this field? This paper tries to discuss these issues with respect to classical board mind games and provides partial (though highly subjective) answers to some of the open questions.

The main conclusion from the arguments specified in the paper is that one of the major, ultimate goals of CI in classical board game research concerns possessing by machines the ability to mimic human approach to game playing. This includes the human-specific learning methods (learning from scratch, pattern-based learning, multitask and unsupervised learning) and human-type reasoning and decision making (efficient position estimation, abstraction and generalization of game features, autonomous development of evaluation functions, effective pre-ordering of moves, and selective, contextual search). Three topics i.e. autonomous learning, knowledge discovery and intuition are discussed in this paper in more detail.

I. INTRODUCTION

Games for years have been a fascinating topic for Artificial Intelligence (AI). The first scientific attempts to invent “thinking machines” able to play mind board games began in the middle of the previous century. Cloude Shannon in his paper devoted to playing Chess [1] proposed the implementation of two types of computer players called Type A and Type B. The former algorithm searches the whole game tree to a certain depth e.g. using the min-max, while the latter one relies on generation of a subset of all potentially promising moves and exploration of them up to the quiescent (semi-stable) position, if time permits. Shannon also proposed numerical estimations of Chess figures’ values as well as certain positional features, e.g. pawn structure or mobility. Although the postulated values were not universally applicable (e.g. were not at all effective in endgame positions) it was an interesting attempt of quantifying tactical and positional strengths and weaknesses.

Alan Turing [2] discussed the problem of making a computer program to play tolerable Chess games. He proposed to selectively search the game tree to the depth of 2 moves (4 ply) and then estimate the values of the encountered positions. Turing proposed also certain numerical values as estimators of Chess figures and pointed out a few simple positional features worth being considered, e.g. the existence of capturing moves, overall mobility, defending pieces by other pieces, etc. He also introduced the notion of a “dead” position, i.e. the one which is stable (in some sense) and can be properly evaluated.

Newell, Shaw and Simon [3] surveyed the previous attempts to define a Chess playing program (including Shannon’s and Turing’s works) and presented their own proposal, which can be shortly described as goal oriented. Two examples of such goals, namely the “center control” and the “material balance” were considered in detail. Specific move generators associated with these predefined goals were also proposed in [3]. The authors underlined the importance of implementing heuristics which are generically similar to those used by humans.

Another great forerunner in the field of game playing was Arthur Samuel [4], who applied a variant of Temporal Difference Learning method\(^1\) to the game of Checkers. Samuel’s program (1959), equipped with a priori defined set of expert features presumably relevant for building board evaluation function, through a carefully designed process of self-playing was able to successfully define the appropriate subset of essential features together with their weights and form a polynomial evaluation function allowing for an advanced level of play.

The above cited seminal papers started the research which remained an interesting and challenging topic until today.

Why game playing research is interesting and important for the community? The reason is generally twofold: first of all, games provide cheap, reproducible environments suitable for testing new search algorithms, pattern-based evaluation methods or learning concepts. The second of all, people always tended to challenge themselves (by machines) in their mind abilities, including game playing. Hence, this “human aspect” of artificial game playing should not be underestimated. On the other hand the machines already surpassed the top human players in several challenging games, like Checkers, Othello or even Chess - regarded as “the Drosophila of Artificial Intelligence”.

In the light of undisputed AI achievements in machine game playing the natural question arises - “Are there any more reasons for pursuing the game playing research (especially with respect to the games in which the machines excel humans by a large margin)?”\(^2\). The paper attempts

---

\(^1\)The name Temporal Difference was coined 20 years later after Sutton’s paper [5].
to discuss this issue and provide arguments supporting the positive answer.

The focus is on the most popular two-player, zero-sum, perfect-information board mind games, in particular Chess, Checkers, Othello and Go. Certainly the above list is by no means complete. Shogi, Chinese Chess, Hex, Amazons, Octi are also examples of this category and playing them on a master level requires high intellectual competencies and a lot of experience. Nevertheless, the above mentioned four games are presumably the most popular among human players and - more importantly - for years have been targets for various AI/CI attempts of challenging human supremacy.

The following distinction between AI and CI methods (systems, approaches) will be made in the remainder of the paper: the label CI will refer to all approaches based on neural networks, genetic or evolutionary algorithms, fuzzy systems, reinforcement learning, Bayesian methods, probabilistic reasoning or rough sets capable of learning and autonomous improvement of their behavior. On the other hand, all methods (systems) applying extensive, sophisticated search methods, carefully tuned evaluation functions, without any ability to learn or autonomous improvement.

The goal of this paper is to express a belief that despite respectful accomplishments of traditional AI methods, alternative ways of developing “thinking machines” in board mind game domain are possible and needful. These methods include cognitive, knowledge-free approaches capable of learning from scratch based merely on an unguided training process (e.g. evolutionary or reinforcement-type) or based on agent’s experience obtained gradually through (self-) playing or in a supervised training process (e.g. with neural nets), but again without explicit implementation of human experts’ knowledge.

The paper is organized as follows. In the next section a brief description of state-of-the-art accomplishments in machine game playing in four games considered is presented. A discussion on the challenging issues in game playing domain and further motivation for pursuing this research topic is presented in Section III, in which particular challenges are also looked upon in more detail. The last section concludes the paper by summarizing its main theses.

II. STATE-OF-THE-ART PLAYING PROGRAMS

Chess. The most striking achievement of AI in games was, most probably, Deep Blue II’s victory over Garry Kasparov - the World Chess Champion at that time. The match was held in New York in May 1997 and Kasparov - one of the strongest Chess players in the history - was defeated by the score 3.5 : 2.5. This historical event ended a nearly 50-year era of Chess programming efforts which started by Shannon’s paper [1].

The success of Deep Blue II was owned to a very sophisticated evaluation function implemented straight in the hardware chips. 8,000 features were implemented in a single chip, and 480 such chips were used in a massively parallel, extremely effective search system build based on 30-node cluster. In effect Deep Blue II was capable of searching between 100 million and 330 million positions per second depending on their tactical complexity [6] or 50 billion positions in three minutes (which was the average time allotted for each move) [7].

Certainly, several problems, other than creation of a sophisticated evaluation function and taking care of demanding hardware issues, had to be resolved on the way to the final machine’s victory. These include, for example, implementation of highly-selective search methods or creation and analysis of the extended opening book and endgame database.

There in no doubt that Deep Blue II is a milestone achievement from an engineering and programming point of view [8], [6]. On the other hand, from a CI viewpoint much less can be said since the system did not take advantage of any learning or self-improvement mechanisms.

The success of Deep Blue II drew a thick line in human-machine Chess competition and began the era of computers supremacy in this game, but fortunately did not cause (as many expected) the immediate death of this rivalry. Several other Chess programs, e.g. Shredder, Deep Junior, Fritz or the recent Chess supercomputer - Hydra played successfully against human grandmasters after Deep Blue II’s victory. The latest computer achievements are the success of Deep Fritz over the World Chess Champion Vladimir Kramnik and the striking result of Rybka, which outpaces the other human and computer players by a large margin, approaching 3000 ELO points on a PC machine with single CPU!.

Checkers. The first computer program that won a human world championship was Chinook [9], [10], [11], which defeated Dr Marion Tinsley - the ultimate Checkers genius who was leading the scene for over 40 years.

Chinook was developed by Jonathan Schaeffer and his colleagues from the University of Alberta and similarly to Deep Blue II, is regarded as a large scale AI engineering project. The vital AI aspects of its design include efficient search, well-tuned evaluation function, opening book and endgame database. In 1994 re-match with Dr Tinsley, Chinook was equipped with a complete 7-piece endgame database and a 4 x 4 subset of an 8-piece database, i.e. all endings in which each side was left with exactly 4 pieces [10].

Special care was taken over specific tactical combinations, which were schematically coded in special tables. The evaluation function was linear and composed of over 20 major components, each of which having several heuristic parameters.

Since 1989 Schaeffer and his collaborators concentrated on the ultimate solving the game. The result was achieved and published in Science [12]. The game of Checkers was proved to be a draw (assuming perfect play of both sides). As part of the proof a complete 10-piece end-game database was developed. Checkers are the most challenging board game
Othello. Another popular mind game in which computers outperformed humans is Othello. In 1997 Michael Buro’s program Logistello [13], [14] decisively defeated the then Othello World Champion Takeshi Murakami with the score 6 : 0. Taking into account that Logistello was running on a single PC machine and considering the post-mortem analysis of the games, which was very favorable for Othello, makes Buro’s achievement even more respectful.

The main contribution to this strong victory is attributed to the three following ideas. First, the new efficient way of feature selection for the evaluation function was implemented [15]. Each feature was a logical combination of some appropriately chosen predefined, atomic features. Second, a new forward pruning method ProbCut (and Multi-ProbCut) capable of cutting out the most probably irrelevant subtrees with predefined confidence was applied [16]. This method generalizes the shallow search results by using statistical estimations of the relation between shallow and deep min-max search. Third, automatic opening book development mechanism was used. It took advantage of the search results along the promising lines not actually played during the game, leading to potentially interesting opening alternatives in the future [17].

It is worth to underline that all three above mentioned ideas are to large extent game independent and can be generally applied to other two person, perfect information games. They are also in line with human way of playing or learning how to play the game. Constructing the evaluation function as a combination of primitive features, performing selective search or looking for new variants in already known game openings is a typically human behavior.

Go. It is generally agreed that Go remains the most demanding, grand AI/CI challenge in game domain. The rules of Go a fairly simple and can be easily explained to a few year old kid. There is no pieces differentiation in Go and, at first sight, the game seems to be much simpler than Chess. However, despite this seeming simplicity, playing the game well by machines is still an unattainable task. The most advanced Go programs can still be beaten by strong amateur human players.

There are several reasons for this situation. First, Go has a very high branching factor, which effectively eliminates brute-force-type exhaustive search methods. Another very distinctive feature that separates Go from other popular board games is the orders of magnitude slower static positional analysis of the board is this game compared to other games [18]. Additionally, proper positional board judgement requires performing several auxiliary tactical searches oriented on particular tactical issues [18]. Considering the variety of positional features and tactical threats it is highly probable that, as stated in [18], “no simple yet reasonable evaluation function will ever be found for Go”. Another impediment in efficient machine playing is the “pattern nature” of the game. On the contrary to humans, who posses strong pattern analysis abilities, computer players are very inefficient in this task. This is mainly caused by the lack of mechanisms (either predefined or developed) that allow flexible subtask separation. Ideally, solutions obtained for these separated subtasks should then be aggregated - considering complex mutual relations between them - at a higher level and ultimately provide the final estimation of the board position. So far, in the currently playing programs, only relatively simple pattern matching techniques are implemented [19], [18].

Since Go playing programs are still not in a mature phase of development (as of human standards) it is hard to point out the stable leader among them. Instead, there exists a group of about ten programs occupying top positions in the major computer Go tournaments. playing on a more or less comparable level. These include: MoGo, Crazy Stone, Aya, GNU Go, Handtalk, Gointellect, Explorer, Indigo and a few more. A detailed discussion on the development of Go playing programs can be found in [20], [18].

III. CHALLENGES FOR CI IN MIND GAMES

Recent AI advances in board games, which gave rise to spectacular challenging the human supremacy in Chess, Checkers, Othello or Backgammon provoke the question: “Quo vadis mind game research?”. Is there any more reason for pursuing research in this area or maybe by defeating the human world champions the ultimate goal was already achieved? Posing this question another way: what kind of knowledge or experience can be possibly attained from continuation of the AI/CI research in this area and what are the short-term and long-term demanding and satisfying problems worth to work on?

Certainly, if the sole reference point in the man-machine competition were the scale of the computer supremacy over humans then the only thing that might be expected is the extension of the upper hand that machines already have, since it is very doubtful that humans would be able to regain the throne or even narrow the gap. But is it really a unique motivation?

I think there are a few good reasons for continuation of human efforts is CI-based game playing. All of them are connected with the way in which high playing competency is accomplished by machines.

There are generally two possible approaches to building a world-champion playing programs. On the one hand there are extremely powerful AI methods, which include carefully designed evaluation functions, look-up tables, perfect endgame databases, opening databases, grandmaster game repositories, enormously fast search methods and many other predefined, knowledge-based tools and techniques that allow making extremely high quality moves. For this type of approaches the goal is to achieve the highest possible level of play, but not necessarily to follow the human way of playing/learning.

On the other hand there are soft, CI-based methods, which include knowledge-free approaches (e.g. learning from scratch, learning based exclusively on the final outcomes of the games played), extensive training methods (e.g. reinforcement learning, neural networks, self-playing), generalization
from shallow search results to deeper depths, autonomous construction of the evaluation function (either by unguided choice of features from a pre-defined pool or by gradual development process based on gained experience). These methods - to some extent - try to resemble the human way of playing/learning, naturally on a general, functional level.

Development and application of these “human-type” techniques pose several challenging questions which are in short presented in the remainder of this section. In particular three fundamental topics, namely autonomous learning, knowledge discovery and intuition are briefly discussed. A more comprehensive presentation of the challenging issues can be found in [21].

A. Autonomous Learning

One of the most distinctive features of CI-based systems is the ability to improve themselves through a (self)-learning process or evolution given only some initial basic knowledge.

Construction of game playing agents of this type is one of the most interesting and demanding problems. There are several notable examples of such systems e.g. Tesauro’s Neurogammon [22] and TD-Gammon [23], [24], [25] - in both cases the evaluation function was implemented as a multilayered perceptron trained by backpropagation and Temporal Difference (TD), respectively; Baxter et al.’s KnightCap [26], [27], [28] - utilizing the TDLeaf(λ) method for the game of Chess; Schaeffer et al.’s TDL-Chinook [29] - also applying the TDLeaf(λ) learning but in Checkers domain; Thrun’s NeuroChess [30] - which combines TD learning with Explanation-Based Neural Network learning (EBNN) in order to learn the evaluation function for Chess based solely on the final outcomes of the training games; Chellapilla and Fogel’s Anaconda alias Blondie24 [31], [32] - an evolutionary experiment over an ensemble of suitably designed neural networks, each representing an evaluation function for the game of Checkers, having as a sole input the positions of pieces on the board and the sum of all board inputs (reflecting the cumulative difference in material); Blondie25 [33] - a twin neuro-evolutionary experiment in Chess.

Certainly the systems mentioned above by no means pretend to be a complete catalogue of CI achievements in this area. They are rather a partial collection of milestone accomplishments subjectively chosen by the author. Due to a space limit there is no possibility to discuss these and other notable examples of autonomously learning systems in more detail. Please see the cited papers or [21] and citations therein for further discussion.

The common feature of all playing agents described above is the ability to autonomously improve their playing strength based on experience (games played). This improvement is achieved either in the self-playing regime or by playing against external opponents. Interestingly, the above mentioned experiments and other works presented in the literature are inconclusive with regard to whether it is more profitably to favor self-playing or rather to train with external opponents. Hence, one of the interesting and challenging issues is further investigation and formalization of the strengths and weaknesses of both training approaches.

In the case of training with external opponents additional key issue is the choice of the opponent players and the scheme of training [26], [34]. In particular, in TD learning one may consider updating weights of the evaluation function after each game or only after the games lost or drawn. Another possibility is to update the weights regardless of the game’s outcome, but with elimination of weak moves which most probably may be misleading for the training process [26], [27]. One may also consider playing against stronger opponent a few times in a raw if only the learner keeps losing against that opponent [35]. Other learning schemes, e.g. the tournament choice of the opponents, can also be considered. The issue of how to define the optimal training scheme deserves further investigation and hopefully new conclusions across various game domains will come into light.

Another important issue is the choice of initial weights in the evaluation function. Regardless of the training method (being either TD, neural nets or evolutionary approach) the choice of the starting point is in most cases crucial for the final outcome of the learning process. Usually these initial settings are based on human expert knowledge. Another possibility would be to define a universal, game-independent procedure allowing the development of “reasonable” initial settings that approximate the relative importance of particular features or their combinations.

Naturally, the problem of how to obtain/learn close to optimal evaluation function for a particular game is also a challenge (see e.g. [36] for Othello. A more demanding question would be how to define the semi-autonomous and game-independent process that would have led to the construction of suboptimal set of board (game) features. This issue was discussed by Paul Utgoff who stated in [37]: “Constructing good features is a major development bottleneck in building a system that will make high quality decisions. We must continue to study how to enhance our ability to automate this process”.

A related challenge concerns autonomous learning with zero initial knowledge (except for the rules of the game). A notable example of successful learning without built-in human knowledge using evolutionary techniques is Anaconda, but possibly other approaches are also worth investigation.

B. Knowledge Discovery

One of the long-term goals of CI in game playing is development of creativity mechanisms which implemented in the playing program might lead to spontaneous knowledge discovery.

Some examples of such “emerging knowledge discovery” have already been presented in the literature however, to my knowledge, all of them were merely “side effects” of the training process. The most famous example is Tesauro’s TD-Gammon. According to former Backgammon World Champion Robertie, the program came up with genuinely novel strategies that no one had used before. TD-Gammon’s play caused revision in human positional judgement in this game
leading, for example, to invention of new opening moves - proposed by TD-Gammon and subsequently proved (in exhaustive, statistical analysis as well as tournament play) to be successful. Another interesting observation concerning TD-Gammon is the development of spatial weight patterns in the MLP, responsible for representation of particular game concepts, which were not explicitly presented in the course of training [23].

Knowledge discovery in the game of Chess was reported in the MORPH experiment [39], [40] relying on pattern based learning with the weights of patterns being modified through the TD(λ) method combined with simulated annealing. Although the strength of MORPH was far inferior to GNU Chess, the patterns learned by the system were consistent with human Chess knowledge. In particular MORPH was able to play openings on a reasonable level, despite the fact that no information about the significance of development or controlling the center of the board in the opening phase was presented to the system. On the other hand, one of the weaknesses of MORPH was poor scalability with respect to the number of patterns, due to the lack of efficient selection mechanisms. Nevertheless, the system was able to defeat human novices while searching only 1-ply.

A general approach to automatic feature generation was presented by Fawcett and Utgoff [41]. Given only domain theory and the ability to solve problems in this domain the system called Zenith was able to automatically generate a set of relevant domain features. The system started with a single feature created automatically from the problem’s goal (e.g. “win of white”) and by using four predefined transformations: decomposition, abstraction, regression and specialization, gradually extended the set of features in iterative manner. Zenith was applied to Othello with promising results. The system, for example, autonomously discovered the importance of pieces, which cannot be reversed [41].

Another well-known example of independent feature discovery in games is Anaconda, mentioned in previous subsection, which received its name due to the “snake-like” way of playing - in many of the games won by the program the opponent was blocked and consequently forced to make a weak move. The importance of the concept of mobility was “autonomously invented” by the system or more precisely by the evolutionary process, which guided Anaconda’s development.

The potential strength of neuro-evolutionary approach was also reported in Othello [42]. The evolved networks “discovered” positional features and advanced mobility issues indispensable for high-profile tournament play.

All the above examples led to discovering new features, previously unknown to the system, induced from the training data. The ultimate goal that can be put forward in this context is autonomous discovery of all relevant components of the evaluation function in a way allowing their separation.

3 A similar effect of spontaneous representation of game specific concepts in the MLP’s weight space was also observed by the author in the system trained to solve the Double Dummy Bridge Problem [38].

and explanation. Such a requirement goes beyond Anaconda experiment and other neural or neuro-evolutionary type approaches that resulted in efficient numerical approximation of the board state, but lack the feature-based formulation of the evaluation function.

C. Intuition

Implementation of the concept of intuition is definitely one of the greatest challenges in computer games and also in computer science in general.

One of the most salient research studies focused on understanding and implementation of intuition was performed by Herbert Simon who concluded that intuition is nothing mysterious or extraordinary and simply relates to a subconscious pattern recognition process able to immediately provide appropriate pattern(s) among those stored in the memory, based on our knowledge and experience [43].

In many board games intuition plays a leading role at master level of play. Consider for example Chess. With a branching factor of about 30, in a 50 move game there are about $10^{147}$ contingencies, which is an enormous number for any human being (grandmasters are believed to search no more than a few hundred contingencies during the entire game). How then is it possible that Chess champions are able to play at such a high level? One of the factors is intuition which allows them to perform highly selective search in this huge space, although in many cases they are not able to explain why they have chosen to search a particular contingency and skipped the others.

Another aspect of intuition in board games is the ability of almost instantaneous recognition of strengths and weaknesses of a given position. A grandmaster usually needs only a few seconds of board analysis in order to tell which side is in the winning or favorable position. One of the possible psychological explanations of this phenomenon is the ability of advanced players to link the new position with previously explored familiar ones and consequently focus on moves and plans associated with these, already known, positions [44].

In majority of mind games, intuition is one of the main factors contributing to the beauty and attraction of the game. Its application, for example, often leads to long term material sacrifices without apparent possibility of its recovery. A well known example in Chess is the immortal game played in London in 1851 by Adolf Anderssen and Lionel Kieseritzky. Another interesting example of intuitive sacrifice occurred in the game played between two great archenemies: Anatoly Karpov and Garry Kasparov in the New York match in 1990. In the middle-game position Kasparov sacrificed queen for a rook and knight on moves 16 – 17 (see Fig. 1) and this sacrifice was clearly positional with no immediate material threats. The game continued up to 53th move, when players agreed for a draw. Human-type intuition in machine playing may possibly emerge also as a “side effect” of using very efficient, close to optimal evaluation function. Examples of “intuition” of such origin have been observed in the famous Kasparov vs Deep Blue re-match, in which some of the machine’s moves were described by grandmasters...
commenting on the match as *phenomenal* or extremely human.

One of very few attempts focusing on formalization of intuitive concepts was recently described by Arbiser [45] for the game of Chess. The author proposed the way of formalizing such concepts as capture, attack, threat, sacrifice, etc. as well as the notion of *style of opponent’s play*, i.e. aggressive, defensive, conservative, tactical or positional. The underlying idea is based on generalization of the null-move heuristic in such a way that instead of hypothetical opponent’s moving twice in a row, the opponent is allowed to virtually change one of his or our pieces or add/delete a piece and then make a move. For example the notion of aggressive play will be implemented by exchanging one of the opponent’s or our pieces into a strong opponent’s piece before making the actual move. Such an exchange would most probably cause immediate threats to us thus forcing the choice of an appropriate response. Although the description of the method raises several questions concerning its time complexity as well as the omitted implementation details, overall the algorithm seems to be an interesting proposition.

Understanding and furthermore implementation of the mechanism of intuition in artificial players is one of the main challenges for CI in games and efficient and general approach to this wonderful human ability is yet to be specified. I would argue that unless programs (machines) capable of making intuitive moves (in the above described sense) in Chess and other mind board games are created, we should be very cautious about announcing the end of the human era in this area.

## IV. Conclusions

The underlying statement of the paper is that further development of Computational Intelligence methods in game domain is important and desirable. The ultimate goal that can be put forward is building a truly autonomous, human-like multi-game playing agent. In order to achieve this goal several challenging problems have to be addressed and solved on the way.

One of the most interesting issues is implementation of mechanisms of autonomous knowledge discovery that would lead to creation of new game features and new playing strategies. In particular a very challenging task is autonomous choice of board features that compose efficient (close to optimal), descriptive game representation allowing adequate evaluation of board positions. At the moment the development of a world-class playing program requires that the set of features be predefined by human experts. Even though there exist a few notable examples of learning how to play certain games without human expertise, there is still a lot of work ahead.

Another fundamental issue is the ability to improve artificial player’s behavior through the learning process resting solely on the experience-based knowledge acquired from previously played games. An interesting, open problem in this area is analysis of pros and cons of two main learning schemes used so far, i.e. playing against external opponents vs self-playing. In the former case, additional open problems concern the optimal choice of the training opponents and the training schedule. Both types of learning were successfully applied to various board games especially with TD learning algorithms. Further exploration of these two directions may reveal some game-independent, general insights.

Yet another formidable challenge is implementation of intuition in the game-playing systems or, more precisely, implementation of mechanisms that would efficiently pretend human-type intuitive behavior. Such achievement would straightforwardly lead to efficacious search-free pre-selection of moves and instantaneous estimation of position strength as well as the ability to play strong positional moves relying on shallow search only. All three above mentioned skills are typical for experienced human players, but still generally non-attainable for machines.

## References
