Revista Brasileira de Computação Aplicada, Abril, 2019

DOI: 10.5335/rbca.v11i1.9111
Vol. 11, Nº 1, pp. 96–103
Homepage: seer.upf.br/index.php/rbca/index

ISSN 2176-6649

ORIGINAL PAPER

Solving a hypothetical chess problem: a comparative analysis of computational methods and human reasoning

Léo Pasqualini de Andrade[1], Augusto Cláudio Santa Brígida Tirado[1], Valério Brusamolin[1] and Mateus das Neves Gomes[1]

[1]Instituto Federal do Paraná, campus Paranaguá

*leopasq10@gmail.com; augustotirado@gmail.com; valerio.brusamolin@ifpr.edu.br; mateus.gomes@ifpr.edu.br

Received: 2019-02-15. **Revised:** 2019-02-27. **Accepted:** 2019-03-15.

Abstract

Computational modeling has enabled researchers to simulate tasks which are very often impossible in practice, such as deciphering the working of the human mind, and chess is used by many cognitive scientists as an investigative tool in studies on intelligence, behavioral patterns and cognitive development and rehabilitation. Computer analysis of databases with millions of chess games allows players' cognitive development to be predicted and their behavioral patterns to be investigated. However, computers are not yet able to solve chess problems in which human intelligence analyzes and evaluates abstractly without the need for many concrete calculations. The aim of this article is to describe and simulate a chess problem situation proposed by the British mathematician Sir Roger Penrose and thus provide an opportunity for a comparative discussion by society of human and artificial intelligence. To this end, a specialist chess computer program, Fritz 12, was used to simulate possible moves for the proposed problem. The program calculated the variations and reached a different result from that an amateur chess player would reach after analyzing the problem for only a short time. New simulation paradigms are needed to understand how abstract human thinking works.

Key words: Chess Software; Cognition; Computational Modeling

Resumo

A modelagem computacional tem possibilitado aos pesquisadores realizar simulações de tarefas onde muitas vezes se torna impossível a experiência prática, como por exemplo, nas buscas em decifrar o funcionamento do cérebro humano. O jogo de xadrez é utilizado por muitos cientistas cognitivos como ferramenta de investigação no campo de estudos sobre a inteligência, padrões de comportamento, desenvolvimento e reabilitação cognitiva. Através da análise computacional em banco de dados com milhões de partidas de xadrez é possível predizer o desenvolvimento cognitivo e averiguar padrões de comportamento de jogadores. Porém, o computador ainda não consegue resolver problemas de xadrez onde a inteligência humana analisa e avalia de forma abstrata, sem a necessidade de muitos cálculos concretos. O objetivo deste artigo é simular, observar e descrever uma situação problema de xadrez proposta pelo matemático britânico Penrose, abrindo espaço de discussão para a sociedade sobre a inteligência humana em comparação com a artificial. Para tanto foi empregado um programa de computador especialista em xadrez, o Fritz12, que simulou as jogadas possíveis do problema proposto. A conclusão foi de que o programa de computador calculou as variantes e avaliou resultado diferente do que um amador de xadrez conclui em pouco tempo de análise da posição problema, sendo ainda necessários novos paradigmas de simulação para se entender como funciona o pensamento abstrato humano.

Palavras-Chave: Cognição; Modelagem Computacional; Softwares de Xadrez

1 Introduction

Computational modeling and chess are widely used by cognitive scientists for research purposes. Cognitive scientists investigate human intelligence in an interdisciplinary field of knowledge involving neuroscience, psychology, the philosophy of mathematics and computer science.

> "In recent years, with the rapid development in computer software and hardware technologies, big data and the artificial intelligence (AI), cognitive computing has received considerable attention in both academic and industry. In the academic, the IEEE Technical Activity for cognitive computing defines it as 'an interdisciplinary research and application field', which 'uses methods from psychology, biology, signal processing, physics, information theory, mathematics, and statistics' in an attempt to construct 'machines that will have reasoning abilities analogous to a human brain'." (Chen et al.; 2018).

Chess, which is believed to have originated in India, dates from the sixth century and has evolved over the years (Murray; 1913). It is played on a checkered board with sixty-four squares and two equal armies, each of which has a king and queen, two rooks, two bishops, two knights and eight pawns (Fide Team; 2018). The players make moves in turn, and the aim of the game is to checkmate the opponent's king. The essence of chess lies in its embodiment of various mathematical concepts, such as arithmetic, algebra, geometry and logical reasoning, as well as spatial vision (Saarilouma; 2001).

Chess programs are very popular with chess lovers because they can be used to play against human competitors on the Internet as well as against the programs themselves. The games are stored in databases which contain millions of chess games, allowing research into human behavior, cognitive development and intelligence. These databases can also be used to train players and, with the aid of software, identify a particular person's playing style. By anticipating in this way what the person's moves would be, a game against that player can be simulated.

In studies of human intelligence, however, it has not yet been possible to simulate factors involved in thinking that allow an understanding of abstract situations. Artificial intelligence (AI) is not yet able to solve experimental chess problems that require the power of abstraction rather than pure machine calculation. This was demonstrated with a chess problem proposed by scientists at the Penrose Institute and described in Section 4 (Results). The computer calculated the possible solutions for more than fourteen hours, but predicted different results from those identified by humans after analyzing the problem for only a short time.

The aim of this article is to report a simulation in which the Fritz 12 chess program (ChessBase Team; 2010) was used to solve the problem proposed by Penrose and to describe the results in order to provide an opportunity for a comparative discussion by society and the scientific community of aspects of human and artificial intelligence. Studies in this area seek to understand how humans solve chess problems other than by merely performing concrete calculations of the variations.

The article is organized in **five sections** in addition to this Introduction. Section 2 describes chess in the context of computing. Section 3 describes the methodology used here. Section 4 presents the results, which are then analyzed in Section 5. Finally, some conclusions are drawn in Section 6.

2 Chess in the Context of Computing

A study of the behavioral patterns of novice chess players was carried out recently by Leone et al. (2014). The authors used a database with chess games played by novices and experienced players. Using numerical simulation, they identified inexperienced chess players' behavioral patterns, such as repeating moves several times with the same piece in squares close to the last move and attempting to simplify the game by exchanging pieces (captures).

Gaschler et al. (2014) used the German Chess Federation database and computational modeling techniques to predict children's performance after at least ten years participation in tournaments. They made predictions of players' expertise acquisition and how they evolved over time. The German Chess Federation database is recommended as a research vehicle in psychology by Vaci and Bilalić (2017), who encourage other researchers to register on a website where the database is available.

A computational model can be used with the information in the database to calculate players' performances and compare them with the results of other types of models, as well as to carry out, for example, gender-related studies Vaci and Bilalić (2017).

The possibilities of computational models and self-learning in AI as used in chess-playing software such as AlphaZero were criticized by Bratko (2018). The author questions whether the resulting computational models can be applied to other types of human activity.

Cognitive computing models (Chen et al.; 2018) and multimodal observations of humans (Guntz et al.; 2018) based on chess have been described in recent studies. The aim of these authors was to use their studies to help with patient treatment and rehabilitation as well as to detect medical problems.

In Kujala and Saariluoma (2018), the concept of cognitive mimetics for computational models of intelligence was introduced. According to the authors, this concept helps to understand how cognitive processes work in the human mind and how they can be mimicked with computational modeling. The authors provide a historical review of the mimicking of intelligence, its potential uses, methods, applications and successful tests. Models cited by the authors include AlphaZero (which learned chess on its own) and AlphaGo (which learned the game of Go on its own) from DeepMind, as well as IBM's Watson, which played Jeopardy (a general-knowledge TV game show) against humans and won. They conclude that cognitive mimetics can complement pattern-matching and machine-learning based design of AI to solve design problems.

In 2018, Sir Roger Penrose, of the Mathematical Institute, University of Oxford, devised a challenge: he developed a chess problem that existing software could not solve even after many hours of processing. He argued that the AI developed in recent years

simply "does not understand" what happens in practice, as in a chess match, for example. Penrose (2018) used a program, Fritz, that is very popular among chess players. He notes that the software did not manage to find the solution to the proposed problem as it decided there was an easy win for black although the only outcome was in fact a draw. According to Penrose, the software did not understand what was happening on the board.

> "Many examples of highly effective algorithmic systems, such as AI devices, have been constructed in recent years. We have computer-controlled machines like self-driving cars and algorithmic systems that play chess and GO at levels that can outperform even the best of human players. But do such devices actually "understand" what they are doing, in any reasonable sense of that word? I argue that they do not, and as an illustrative example I present a recently composed chess position that a human chess player, after briefly examining it, would correctly conclude that it is an obviously drawn position. Nevertheless, when it is presented to the top-level chess-playing program Fritz, set at grandmaster level, Fritz incorrectly claims that it is a win for the black pieces and eventually Fritz blunders dreadfully (though "correctly" according to its algorithm) to be soon checkmated by white. This demonstrates Fritz's remarkable lack of any actual understanding of the game of chess, despite its vast computational abilities." (Penrose, 2018).

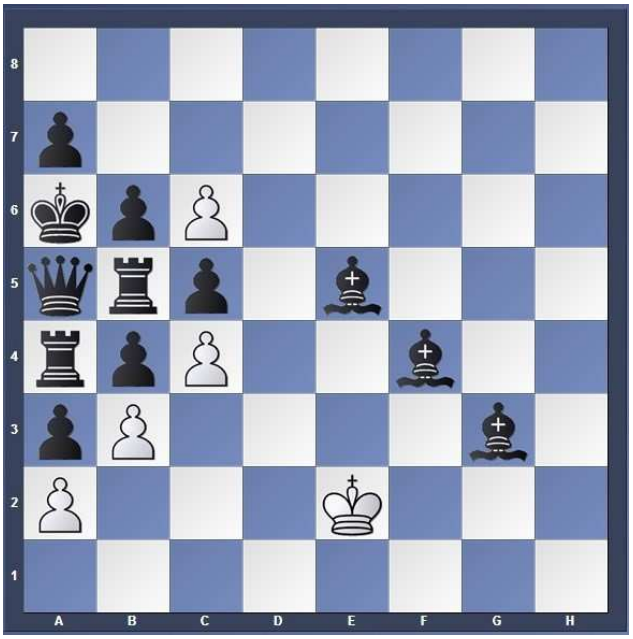

Figure 1: The problem devised by Penrose (2018). Source: recreated by the authors

3 Methodology

The first step consisted of a literature search to identify articles in scientific journals and specialized sites. The following terms were used to search the PubMed database: chess, computer, cognitive development, expertise, visuospatial perception, artificial intelligence.

To check the results obtained by Penrose (2018), the Fritz 12 chess program was used to analyze the hypothetical problem that he drew up. The results were recorded and analyzed in light of the articles identified in the literature search.

4 Results

The chess problem devised by Penrose (Fig. 1) was input into the Fritz 12 program. The program was set to "infinite analysis" mode (in which there is no time limit for processing the variations) and processed the problem uninterruptedly for more than fourteen hours to assess the position. Although black is at an advantage, his position is static as he is caught between white's pawns and unable to move, apart from the bishops on e5, f4 and g3. An amateur chess player will notice that the bishops on their own cannot do anything to checkmate the white king, which can move undisturbed on the white squares.

In Russel and Norvig (2013), the chess programs analyzed use the minimax algorithm to make decisions. This algorithm seeks to minimize the maximum loss and maximize the minimum gain in order to determine the best move. The program achieves this using various technical concepts, from the value of the chess pieces through the pawn structure and dynamic possibilities to checkmate calculations.

The modules used in Fritz 12 assess the position with the help of a numerical value. The assessment is expressed using the pawn as the unit, always from the perspective of white. When the program assigns a value of +1.30, this means that white's position is 1.30 pawns more than black's. If the advantage were only one pawn, the remaining 0.30 would correspond to the value of conceptual position-related factors (mobility, position of the pieces, safety of the kings, pawn structure etc.). A value of -3.00 means that white has three pawns or one piece less (a bishop and knight are each worth three pawns). A rook is worth 5 pawns, and a queen 9 pawns. The king's value is defined as infinite, so loss of this piece results in immediate defeat.

Russel and Norvig (2013) note that the half-move, or ply, is important in programs that use alpha-beta pruning as it corresponds to one level of the decision tree, which is used to determine the risk associated with a move. The alpha-beta algorithm is considered a significant improvement on the minimax search algorithm as it substantially reduces the search tree by using the branch-and-prune technique. This means that when a piece of software is analyzing a game, all it has to do is find a winning alternative. In this case, there is no need to analyze other variations even if they are better. When a winning alternative is found, the others are discarded.

When it is processing the problem conceived by Penrose, Fritz 12 calculates millions of variations in depth with more than forty moves in the decision tree. Based on the calculations it has performed, the program concludes that black will win the game. The result of the calculations is a score of -33.56 for black (Fig. 2). Positive scores (for white, or negative for black) of more than two points signify a decisive advantage in terms of being able to checkmate the opponent's king.

Note that the software's decision tree involves seven variations (Fig. 2). Variations with moves that only involve white's king (variations 1 to 4 in Fig. 2) are calculated to have values of -33.56, i.e., a significant advantage for black. However, they do

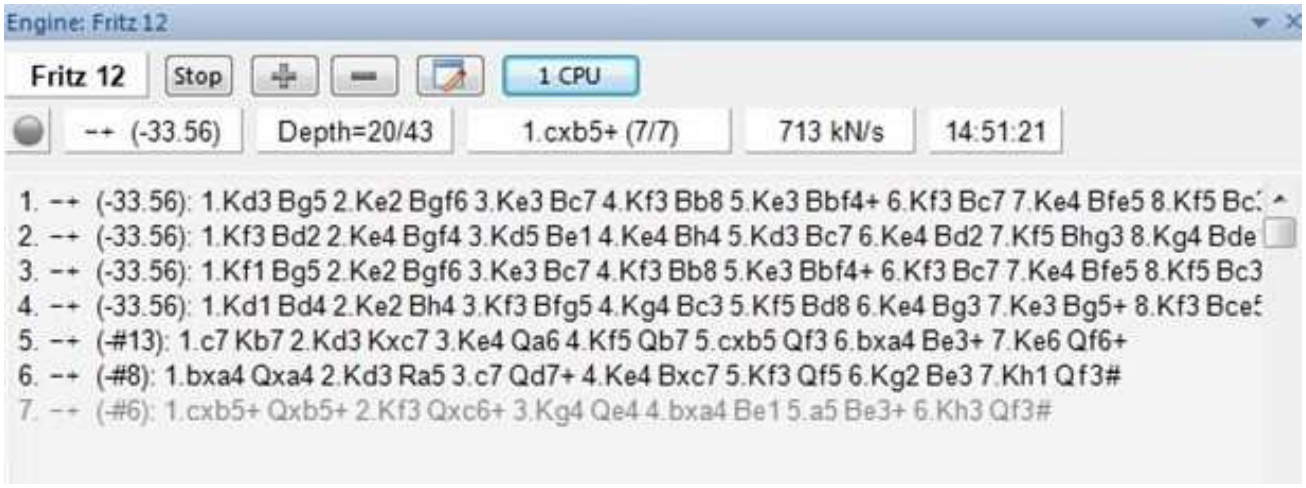

Figure 2: Simulation of the solution to the problem using Fritz 12. Source: the authors

not reach checkmate. In variations 5, 6 and 7 (Fig. 2), the result is checkmate for black in 13, 8 and 6 moves, respectively. It should be noted that white is not obliged to make the moves in these variations.

5 Analysis of Results

The results of the analysis performed by Fritz 12 can be considered equivalent to those obtained by Penrose (2018). This clearly indicates the difficulty the software faced assessing the position correctly. More importantly, other programs on the market have the same difficulty solving the problem even when the configuration of the hardware on which the program is running is superior to that tested here.

It cannot be denied that chess programs suffer from limitations; nevertheless, these relate not to their playing capacity but to their assessment of a position. For these issues to be overcome would require the development of software that is both prohibitively expensive and requires an impractical amount of processing power. It is therefore important to improve decision-tree pruning algorithms to select moves and shorten analyses.

An amateur chess player realizes after analysis of the problem in Figure 1 for only a short while that if he moves the king along the white squares on the board, black cannot force a checkmate. This type of heuristic analysis is intrinsic to humans' mental processes and has not yet been translated into any type of algorithm.

The artificial chess problem proposed by Penrose is unlikely to arise in practice and is the fruit of human creativity, which enables a person to explore possibilities he knows can be easily solved, unlike a specialist program, which cannot make abstract assessments and therefore cannot conclude that the result is different from the result of its calculations.

The process used by humans to perform calculations differs from that used by software in that humans can simplify and select variations to include in the concrete calculation. This simplification is a result of the ability of humans to imagine the decisive position, to understand the limitations on the opponent's moves and to filter irrational moves. The problem faced by scientists is to implement in an algorithm this human abstraction, which, with only a little analysis and a few calculations, discovers that there is no other way to continue the game and that there is no need to go down as many moves in the analysis of the decision tree as Fritz 12 did.

Chess programs have evolved greatly since 1997, when Deep Blue, an IBM chess program running on a computer with 256 interconnected processors, beat the then world chess champion, Gary Kasparov (Marshall; 2014). Countless programs have been developed since, and Fritz 12 is one of this new generation.

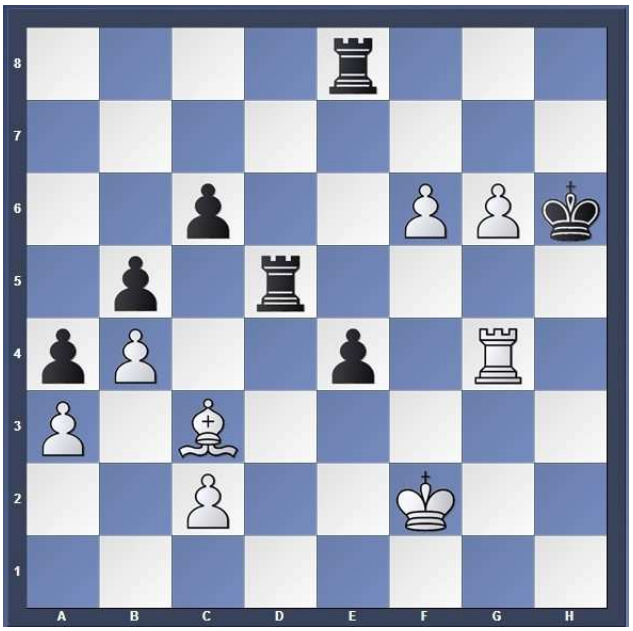

Figure 3: Position in the first match between Kasparov and Deep Blue in 1997. Source: the authors

For the purposes of comparison with current software, we input a position from the first game between Kasparov and Deep Blue in 1997 into Fritz 12. The game was won by Kasparov. Deep Blue's (black's)

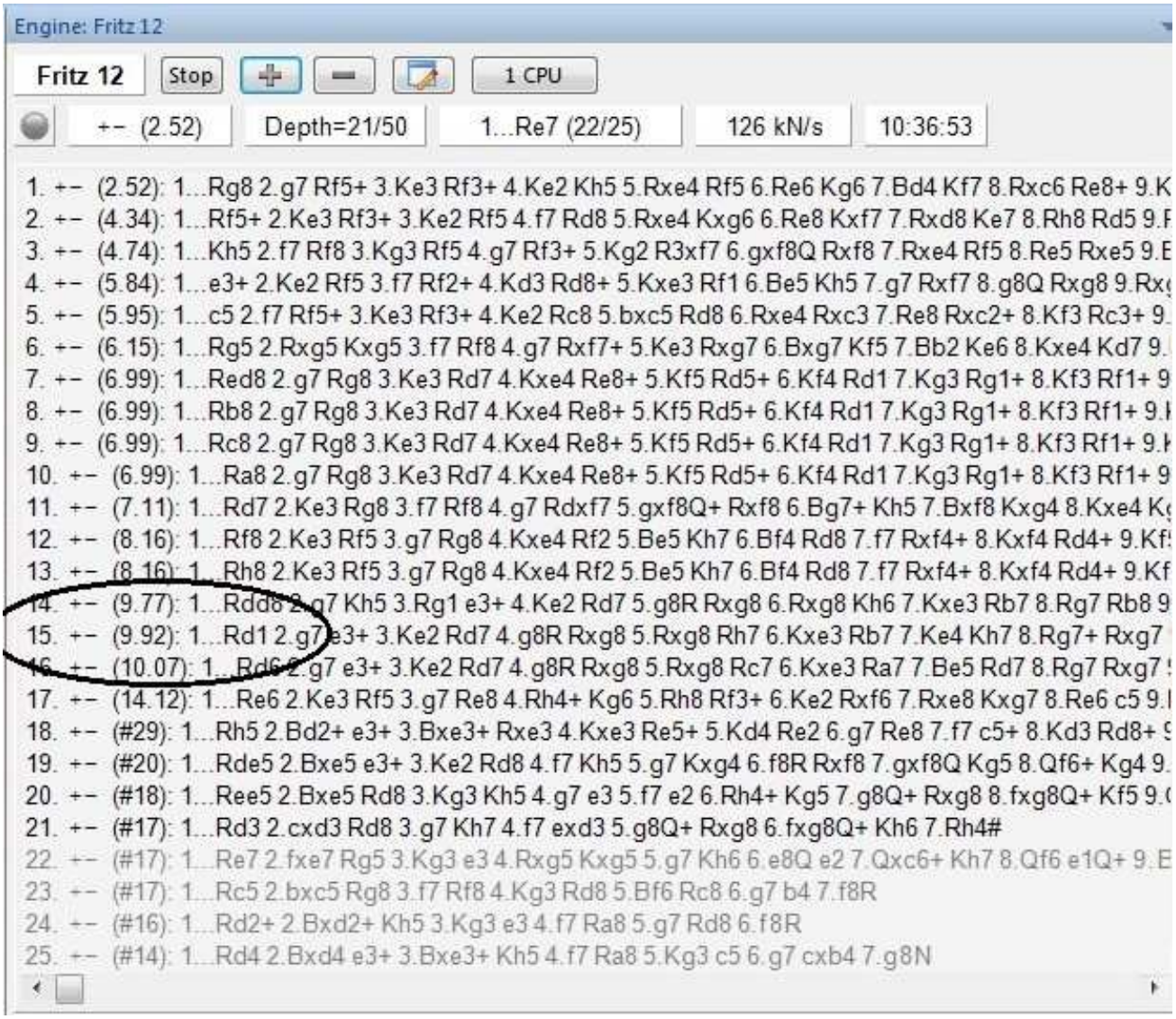

Figure 4: Simulation with Fritz 12 of Deep Blue's move. Source: the authors

44th move (Fig. 3) was a strange one that surprised Kasparov, who began to mistrust the machine for the first time, saying that there was human involvement in the moves it was making. Kasparov soon realized that the move was very bad and enabled him to win sooner. It was only in 2014 that a documentary was released by ESPN in which it was confirmed that there was a bug in the software (Marshall; 2014). Interestingly, in the simulation with Fritz 12, Deep Blue's choice was only the 15th choice for Fritz 12 (Fig. 4).

In the same match in 1997, in the sixth and last game, Kasparov made a provocative move, certain in the knowledge that Deep Blue would not sacrifice the knight on e6 because he believed the software, after coldly calculating the numbers, would not tolerate being at a numerical disadvantage. Kasparov was very surprised by Deep Blue's move, which was precisely to sacrifice the knight on e6. We put the same position into Fritz 12 (Fig. 5).

Even though some twenty years have passed since the original game, the result simulated by Fritz 12 was different from the choice made by Deep Blue (Fig. 6):

As can be seen in the simulation, Ne4 got a better grade in Fritz 12, Deep Blue preferred the second variation (2 = (0.23): 1.Nxe6) to the first (1 = (0.25): 1.Ne4), which would result in a higher score for the machine (Fig. 6). Because of these moves, Kasparov became suspicious of the software and suggested that a good player was "helping" Deep Blue.

Deepening the analysis or the position after the Knight's sacrifice in "e6" (Fig. 7), Fritz 12 shows even worse evaluations (Fig. 8). The program do not "understand" that the black king does not have a safe place to go and white has a winning attack. As in the events horizon of the simulation did not appear a checkmate or any material gain, the algorithm has no reason to choose that sacrifice.

Fritz 12 prefers to return the Knight to "e4" 9), This is the position that Kasparov thought a computer should choose. But Deep Blue has played the former position 7), (Kasparov; 2018) later said:

> "Machines are not speculative attackers. They need to see the return on their investment in their search before they invest material. I knew that Deep Blue would decide to retreat its knight

Figure 5: Position in the sixth game between Kasparov and Deep Blue in 1997 after 7... - h6 by Kasparov. Source: the authors

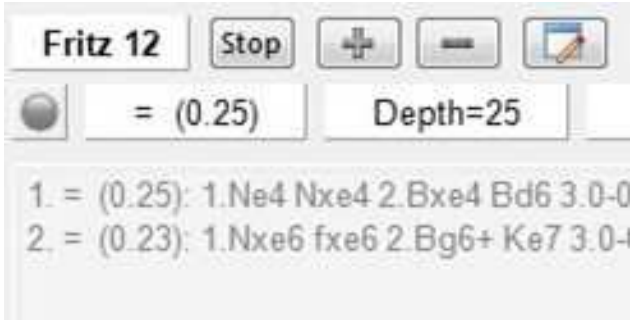

Figure 6: Simulation with Fritz 12 of the move by Deep Blue in the sixth game after Kasparovs move (h6). Source: the authors

> instead of playing the sacrifice, after which my position would be fine"

The position evaluation after "Ne4" (Fig. 10), has a better evaluation than after "Nxe6". But why Deep Blue played a typical human move? Many years later (Kasparov; 2018) got the answer:

> "Here once more is Deep Blue coach Miguel Illescas in his 2009 interview, speaking about the fateful sixth game: 'We were looking at all kinds of rubbish, such as 1.e4 a6 or 1.e4 b6, giving as many forced moves to the computer as we could. On this same morning we also introduced the move Knight takes e6 in the Caro-Kann, on the same day that Kasparov played it. That very morning we told Deep Blue, if Garry plays h6, take on e6 and don't check the database. Just play, don't think.... This was his bet, that the machine would never like this piece sacrifice for a pawn. And indeed, if we had given freedom to Deep Blue to choose, it would never have played it.'"

Although new chess programs have evolved substantially and implement algorithms that make use of heuristics to work out the moves in a game, they are not yet able to solve the problem created

Figure 7: Position in the sixth game between Kasparov and Deep Blue in 1997 after 8-Nxe6 by Deep Blue. Source: the authors

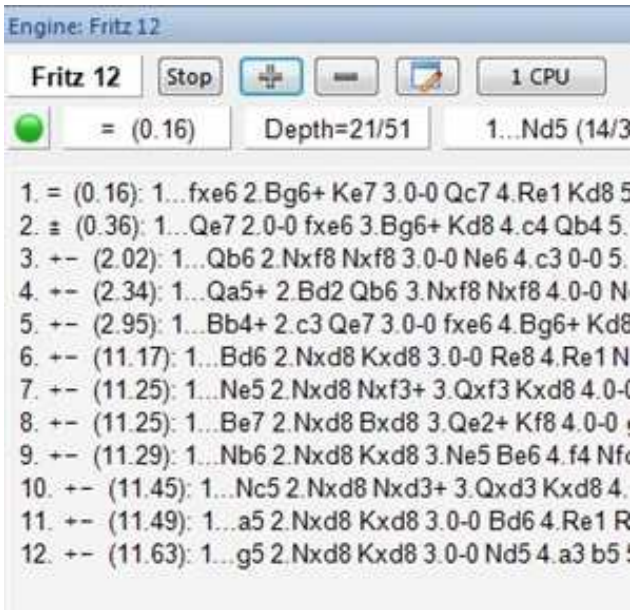

Figure 8: Simulation with Fritz 12 of the move by Deep Blue in the sixth game after 8.Nxe6. Source: the authors

by Penrose. From a technical point of view, chess programs have superior playing strength to humans, a characteristic that can be attributed to their great processing power and access to large databases of information. These factors, nevertheless, do not endow programs with intuition and an ability to learn from their own mistakes (Razmov, V.; 2010). In spite of these limitations, programs such as AlphaZero would appear to suggest that these last barriers described by Razmov have now been overcome.

Specialized programs like AlphaZero and AlphaGo

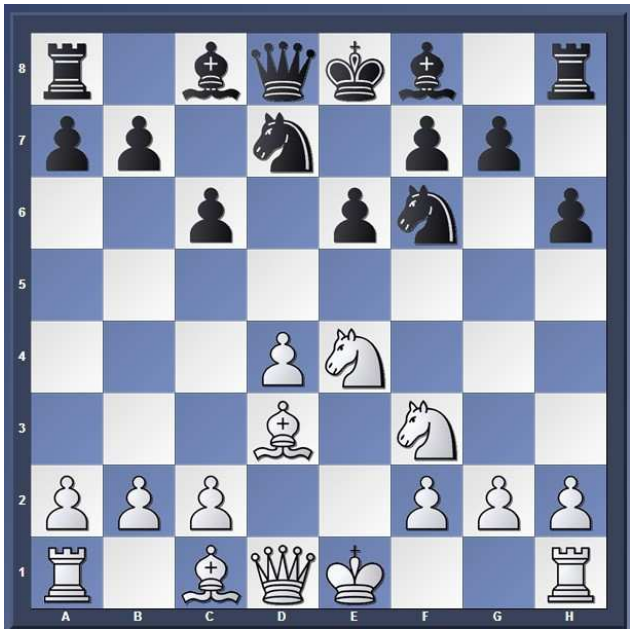

Figure 9: Position in the sixth game between Kasparov and Deep Blue in 1997 after 8-Ne4 by Fritz 12. Source: the authors

Figure 10: Simulation with Fritz 12 of the possible move by Deep Blue in the sixth game after 8.Ne4. Source: the authors

learn using machine-learning algorithms, and their moves are very similar to those made by humans. Like humans, AlphaZero reduces the number of variations processed when solving a problem and uses fewer branches in the decision tree, allowing it to perform calculations in greater depth and find unexpected solutions (Illescas; 2018). Indeed, this new generation of programs may soon be able to solve problems like that designed by Penrose.

Specialized programs that can solve problems which are difficult for humans and find better solutions evolve as the results of studies are shared by the scientific community. The effects of this evolutionary progress will be felt in all sectors and will pose new challenges in terms of reorienting society's priorities.

6 Conclusion

Penrose (2018) composed a chess problem of easy solution for human players, but too hard for computers, because the software do not understands what is happening on the board. In a position that one side has a huge material advantage, the software algorithm is unable to perceive that there is no way to victory.

Analysis of the performance of Fritz 12 reveals that even though some elements of heuristics have been implemented in the software, key problems have yet to be overcome before human intuition can be mimicked. A simulation performed by Fritz 12 at Penrose position obtained the same results, showing the difficulty the software faced assessing the position, and that chess programs suffer from limitations.

Kasparov, former World chess champion was aware of chess programs limitations when played against Deep Blue in 1997. He got very suspicious that there was a human helping the computer, because the machine made a qualitative sacrifice that could not be translated into numbers. An algorithm would not tolerate being at a numerical disadvantage without a predictable winning. This position was also simulated at Fritz 12, confirming the reasoning of Kasparov, that a machine would not choose the move that Deep Blue has made.

Computational modeling has helped researchers in different fields of knowledge and, together with chess, can be an invaluable aid for research into the cognitive sciences. A range of models have been developed for various purposes, especially research into how the human brain learns, and new paradigms have emerged at a speed and on a scale typical of the information technology era.

Although that simulations of Penrose position and Kasparovs game exposed programs limitations, AlphaZero is a new specialized chess program, that can make moves very similar to those made by humans. Maybe AlphaZero can solve Penrose problem as easy as humans and choose the same move that Kasparov said a machine would not make. This subject is suggested for other simulations.

Discussions of these new ideas involve profound reflection on science and technology. Society is bombarded with information and very often is not aware of the network of actors—including scientists and the State—that exert an enormous influence on the resulting technologies. Society must become aware of the real power it can exert in this complex network and discuss the findings and experiences made possible by information technology to ensure the validity of future experiments.

Existing computational models help with many tasks involving large numbers of calculations that are unproductive for humans because of the time required. Although such models are important tools for performing tasks of this nature, they are not yet fully reliable, and any important decisions taken

by them must therefore be scrutinized by a human observer.

References

Bratko, I. (2018). Alphazero–what's missing?, *Informatica* **42**(1). Available at http://www.informatica.si/index.php/informatica/article/view/2226 (Accessed 2 January 2019).

Chen, M., Herrera, F. and Hwang, K. (2018). Cognitive computing: architecture, technologies and intelligent applications, *IEEE Access* **6**: 19774–19783. Available at https://ieeexplore.ieee.org/document/8259243/?reload=true (Accessed 2 January 2019).

ChessBase Team (2010). *Fritz 12*, ChessBase, Hamburg, Germany. Available at https://shop.chessbase.com/en/categories/chessprogramms-fritz (Accessed 15 August 2018).

Fide Team (2018). *Fide Laws of Chess taking effect from 1 January 2018*, FIDE, Athens, Grecia. Available at https://shop.chessbase.com/en/categories/chessprogramms-fritz (Accessed 15 August 2018).

Gaschler, R., Progscha, J., Smallbone, K., Ram, N. and Bilalić, M. (2014). Playing off the curve-testing quantitative predictions of skill acquisition theories in development of chess performance, *Frontiers in psychology* **5**: 923. Available at https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4141457/ (Accessed 2 January 2019).

Guntz, T., Balzarini, R., Vaufreydaz, D. and Crowley, J. (2018). Multimodal observation and classification of people engaged in problem solving: Application to chess players, *Multimodal Technologies and Interaction* **2**(2): 11. Available at https://www.mdpi.com/2414-4088/2/2/11 (Accessed 2 January 2019).

Illescas, M. A. (2018). ¿máquinas o dioses? peon de rey, pp. 10 – 23. janeiro/ fevereiro.

Kasparov, G. (2018). Deep thinking: Where machine intelligence ends and human creativity begins, p. 216. PublicAffairs.

Kujala, T. and Saariluoma, P. (2018). Cognitive mimetics for designing intelligent technologies, *Advances in Human-Computer Interaction* **2018**. Available at https://www.hindawi.com/journals/ahci/2018/9215863/abs/ (Accessed 2 January 2019).

Leone, M. J., Fernandez Slezak, D., Cecchi, G. A. and Sigman, M. (2014). The geometry of expertise, *Frontiers in psychology* **5**: 47. Available at https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3913042/ (Accessed 2 January 2019).

Marshall, F. (2014). Signals: The man vs. the machine., *ESPN Films: The Making of a Sports Media Empire* .

Murray, H. J. R. (1913). *A history of chess*, Clarendon Press. Available at https://archive.org/details/AHistoryOfChess (Accessed 2 January 2019).

Penrose, R. (2018). Can machines be conscious?, *The Science of Consciousness (25th annual The Science of Consciousness, 2018, Tucson, USA*, pp. 136–158. Available at http://www.consciousness.arizona.edu/documents/TSC2018AbstractBookfinal3.pdf (Accessed 2 January 2019).

Razmov, V. (2010). *How Computers Think in Chess*, CHESS.com. Available at https://https://www.chess.com/blog/_valentin_/how-computers-quotthinkquot-in-chess (Accessed 30 August 2018).

Russel, S. and Norvig, P. (2013). Inteligência artificial.

Saarilouma, P. (2001). Chess and content-oriented psychology of thinking, *Psicológica* **22**(1). Available at https://www.redalyc.org/html/169/16922108/ (Accessed 2 January 2019).

Vaci, N. and Bilalić, M. (2017). Chess databases as a research vehicle in psychology: Modeling large data, *Behavior research methods* **49**(4): 1227–1240. Available at https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5541096/ (Accessed 2 January 2019).