On the Artificial Reasoning with Chess: A CBR vs PBR View

Zahira Ghalem[1,2] [a], Karima Berramla[3,4] [b], Thouraya Bouabana-Tebibel[1] [c] and Djamel Eddine Zegour[1] [d]

[1]*Laboratoire de la Communication dans les Systèmes Informatiques (LCSI), Ecole Nationale Supérieure d'Informatique (ESI), BP, 68M Oued-Smar,16270 Alger, Algeria*

[2]*Oran 2 University Ahmed Ben Ahmed, Algeria*

[3]*LAPECI Laboratory, Oran 1 University Ahmed Ben Bella, Algeria*

[4]*University of Science and Technology Mohamed Boudiaf, Algeria*

Keywords: Artificial Reasoning, Game Playing, Knowledge Generalization, Knowledge Representation.

Abstract: In the quest to advance artificial reasoning, this article delves into the contrasting realms of Case-Based Reasoning (CBR) and Pattern-Based Reasoning (PBR). Drawing inspiration from human thinking behavior in tackling novel problems. The study centers on the chess domain, exploring the intricacies of representation, generalization, and reasoning processes. It illuminates the fundamental trade-off between computational efficiency and decision quality in (PBR) systems. This comprehensive examination provides valuable insights into the adaptability of reasoning systems and the role of abstract knowledge bases in enhancing performance.

1 INTRODUCTION

Chess, often referred to as the touchstone of artificial intelligence (Ensmenger, 2012), has been extensively examined due to its accessibility and comprehensibility. From the historical tale of the Turk (Sajo et al., 2008), through the monumental clash between Deep Blue and Kasparov (Campbell et al., 2002), to the superhuman performance of Alphazero (Silver et al., 2017), machine mastery of the game has seen significant advancements. However, these achievements have predominantly relied on resource-intensive brute-force search techniques (Chaslot et al., 2008), complemented by heuristics like Alpha-beta pruning (Sato and Ikeda, 2016).

Intelligence, in a general sense, can be defined as the capacity to take actions that enhance the likelihood of problem-solving (Russell & Norvig, 2003). Given the computational speed of machines, this capability can be artificially replicated through brute-force computation, involving a systematic exploration of potential solutions. However, it is crucial to note that explainable artificial intelligence (AI)extends beyond mere computational power. It encompasses the ability to emulate the cognitive processes of human thinking in machines, enabling them to acquire knowledge, tackle complex problems (Ongsulee, 2017), and provide understandable, interpretable, and transparent explanations for their decisions and actions (Keane and Kenny, 2019). This paper is targeted at distinguished disciplines of existing artificial reasoning methods.

Since the pioneering work of Robert Shank (Shank,1982), case-based reasoning (CBR) has found its way into numerous computer applications leading to the development of successful CBR systems. Often touted for its ability to closely mimic human thought processes (Aamodt and Plaza, 1994), this approach hinges on the idea that solutions to new problems can be derived from the problem-solving experiences of similar, previously encountered issues. Likewise, pattern-based reasoning (PBR) involves eager generalization to extract patterns from a set of prior problems and construct a set of solutions-indicating rules.

[a] https://orcid.org/0000-0002-4070-1237
[b] https://orcid.org/0000-0002-2847-4895
[c] https://orcid.org/0000-0002-9944-3738
[d] https://orcid.org/0000-0001-9538-5895

Ghalem, Z., Berramla, K., Bouabana-Tebibel, T. and Zegour, D.
On the Artificial Reasoning with Chess: A CBR vs PBR View.
DOI: 10.5220/0012624000003645

In *Proceedings of the 12th International Conference on Model-Based Software and Systems Engineering (MODELSWARD 2024)*, pages 378-385
ISBN: 978-989-758-682-8; ISSN: 2184-4348

RBR and CBR represent two complementary paradigms for constructing artificially intelligent systems (Sun, 1995). This paper delves into their respective applications in the context of chess, emphasizing their distinct knowledge representation techniques and approaches to case generalization. It engages in a discussion of the research findings and conclusions in this area and underlines the potential value in adopting a combined perspective that leverages the strengths of both methods.

2 BACKGROUND AND PROBLEM STATEMENT

In addressing everyday problems, our natural inclination is to draw upon past experiences, compare them with new situations, and develop customized solutions. In turn, this process generates fresh knowledge that we can later recall and apply. As illustrated in Figure 1 (CBR) serves as a simulation of this human thinking behavior when tackling new problems.

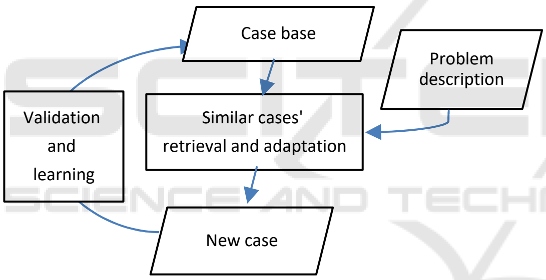

Figure1: CBR reasoning process.

PBR, on the other hand, relies on explicit pre-generalization and employs a more abstract knowledge representation through rules. When faced with a new problem, it selects relevant rules that have premises consistent with the problem description (see Figure 2). In larger domains, summarizing all the knowledge becomes increasingly challenging, and exact matching is seldom achievable. Consequently, rule selection is based on various contextual adaptations. This leads to the interchangeability of the terms "rule" and "pattern" (Reason, 1990).

In a fully automated environment, both CBR and PBR rely on a set of training examples to create generalizations. The key distinction between the two systems is that CBR generates (implicit) generalizations during the search and retrieval process by identifying similarities between base cases and target problems. In contrast, PBR systems make eager explicit generalizations by identifying shared characteristics with the same solution, which are then turned into rules applied to solve future problems.

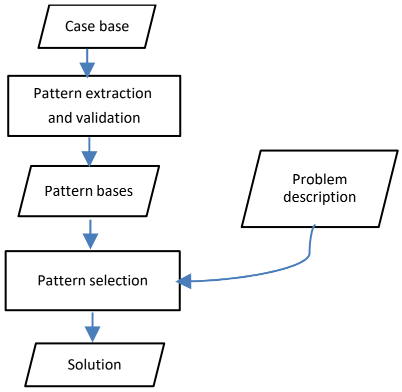

Figure 2: PBR process.

The performance of an artificial reasoning approach relies heavily on the quality of its knowledge base. But can it be influenced by its generalization method? To address this problem, we formulate the following questions:

RQ1: what are the advantages of each of the PBR and CBR approaches**?**

RQ2: how could their shortcomings be mitigated**?**

RQ3: could this lead to a new generalization approach?

3 CHESS GAME: CBR VS PBR

The fundamental challenge in knowledge-based approaches is to extract and present relevant knowledge in a usable form. In this paper, the surveyed research can be categorized into two primary directions: the first utilizes pattern-based, expert-level advice in the form of constraint rules, while the second involves creating case bases from expert-level gameplay.

3.1 Representation

From a CBR perspective, the knowledge base comprises games played at the expert level. In this view, a game can be seen as a series of distinct problems, each with its own solution. These problems involve the remaining pieces and their respective positions, referred to as the board position throughout this article. Due to the complexity of the game (approximately 10^{120} possible positions), the search function must be thoughtfully designed to balance search specificity and accuracy of selected solutions.

To address this challenge, various researchers have explored different approaches for the representation of chess board positions. For instance, (BRATKO et al., 1978) focused on studying endgames with pattern descriptions. Their representation includes a listing of the remaining pieces, their relative positions, and attack/defense relationships, along with defined goals. the educational system ICONCHESS (Lazzari, 1996) combines CBR with fuzzy logic to offer high-level playing advice. This system utilizes cases, drawn from games played by various experts and masters, along with their corresponding analyses, see (Figure 3). David Sinclair (Sinclair, 1998) employed Principal Component Analysis to condense 56 predictive features into 11. Another approach, as seen in (Ganguly et al., 2014) and, represents cases in a textual format, including precise piece positions and their potential interactions (attacks, defenses, counterattacks). Similarly, (Hesham et al., 2021) represent the board position in a simple textual format.

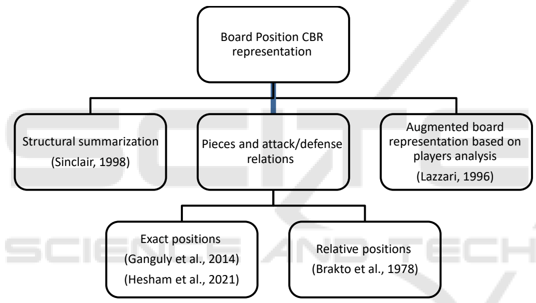

Figure 3: Board representation for CBR systems.

In contrast, knowledge can also be represented as a collection of conditional recommendations based on pattern extraction, suggesting potential winning moves. This approach is exemplified in (Kass, 1990) and (Kerner, 1995), where the concept of Explanation Models (XP) is introduced. An XP serves as a parametric explanation that can be adapted to elucidate new cases. A Multiple Explanation Model (MXP) comprises a collection of XPs, each representing a unique perspective on a given case. These XPs are assigned weights and assessments, contributing to the overall evaluation of the position.

CHUNKER (Berliner & Campbell 1984) employs abstract patterns stored in predefined libraries to assess pawn endgame positions. This approach has been further explored in SUPREM (Berliner & Ebeling, 1984) (Berliner & Ebeling, 1990), a pattern-based program implemented in the specialized machine/program HITECH. In this system, a board position is interpreted as a collection of patterns. Clamp (Cook, 2008), analyses middle-game positions to construct decisive piece groupings for move selection. Contributing to the development of piece-move-oriented chunk libraries.

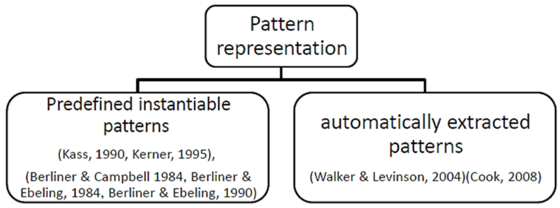

Figure 4: Rule representation for PBR chess system.

3.2 Reasoning and Generalization

The fact that analogous problems have analogous solutions is a cornerstone of CBR Systems. When it comes to a player's perspective, similar board positions often lead to similar moves. This raises the question of which features of a board position are crucial for move selection and how they affect the search process. (Lazzari, 1996) sought out similar positions, including reversed similarity, by evaluating both syntactic similarities (such as the exact location of pieces) and semantic similarities related to plans and similar strategic objectives. In (Sinclair, 1998), the researcher attempted to characterize each position in the case base by considering structural features like pawn formations and material. This approach led to similarity measurement based on the composite distance between these representations. (Ganguly et al., 2014) encoded the remaining pieces, their reachable squares, and attack/defense configurations, adopting an approximation search process that considered the piece's mobility and connectivity. With a simplistic textual representation of the chessboard position, (Hesham et al., 2021) employed base cases to illustrate potential moves for both the player and their opponent. Subsequently, these moves were input into a search algorithm (Plaat et al, 2014) employing alpha-beta pruning (Sato & Ikeda, 2016) to determine the optimal move.

Pattern-based systems, on the other hand, focus on identifying dominant patterns within a query, utilizing contextual adaptation mechanisms since the rule's condition part is expressed in a pattern-like form. In (Kass, 1990) and (Kerner, 1995), patterns with binary properties are used to extract fundamental explanation models from the board position, and the most dominant ones are selected. In CHUNKER Berliner & Campbell 1984), each model consists of instantiable properties, and each instance has a set of values for these properties. SUPREM (Berliner &

Ebeling, 1990) employs predefined pattern recognition in the form of rules that define temporary objectives for players and the necessary models to recognize these goals during the search process. Morph (Walker & Levinson, 2004), after being trained in various games, learns to associate chess piece formations with the possible winning moves. As for (Cook, 2008), when a query is submitted, piece groupings are extracted based on factors like attack, defense, proximity, and more. These groupings are then searched for in the position's legal move libraries, constructed through the piece's move-oriented chunk libraries.

3.3 Results and Insights

The efficiency of an artificial Reasoning system fundamentally hinges on two critical components: representation and similarity metrics. In this context, the dynamic interaction prompts a central question: How significant are the characteristics used for a problem representation?

Within this context, the study conducted by ICONCHESS, as presented in (Lazzari, 1996), places significant emphasis on specific factors that play a pivotal role in characterizing board positions. These factors encompass the positions and types of pieces and the intricate web of playing relations among them. The research underscores the importance of considering these elements when seeking to comprehensively define and understand the unique characteristics of board positions.

The research conducted by Sinclair (Sinclair, 1998) contributes valuable insights into this question. Sinclair's work reveals that the choice of similarity metrics plays a pivotal role in shaping the performance and outcomes of CBR chess systems:

Quality vs. Quantity Trade-off: Sinclair's observations demonstrate a fundamental trade-off. When employing strict similarity metrics, the cases retrieved exhibit a high level of quality. However, this precision often comes at the cost of quantity, as the number of results retrieved tends to be relatively low.

Summarization of Board Positions: Central to this discussion is the representation of board positions. The choice of which features to include, the number of features, and their respective weighting in the computational process can significantly affect the system's performance.

Furthermore, (Qvarford, 2015) investigated the performance of an AI agent that employed CBR with an extensive similarity metric. The outcomes revealed a subpar performance, with a low win rate across different case bases. This underperformance can be largely attributed to the utilization of a comprehensive similarity metric, which may have led to an overly strict matching criterion. The study's findings underline the potential advantage of employing, among other adjustments, a more abstract knowledge base. This could enhance an AI agent's overall performance, potentially leading to more successful outcomes.

However, it's worth noting that the studies discussed in this article exhibit substantial variations in terms of their training data, objectives, and the specific computing platforms on which they were implemented. This diversity makes it challenging to classify these papers solely based on the level of playing they address. A concise summary of the key aspects explored in these various research endeavors is presented in Table 1 for reference and clarity.

The majority of pattern-based systems discussed in this context were conceived and implemented with the primary objective of mitigating the branching factor challenges inherent in alpha-beta search algorithms (Sato & Ikeda, 2016). This challenging task of narrowing down the search space is crucial for achieving computational efficiency in AI systems. Some noteworthy examples include (BRATKO et al., 1978), CHUNKER Berliner & Campbell 1984), SUPREM (Berliner & Ebeling, 1990), (Ganguly et al., 2014) and (Hesham et al., 2021) which demonstrated high playing performances.

For instance, Clamp (Cook, 2008) introduced an approach that resulted in a substantial 50% reduction in the number of nodes examined during the search process. Although this achievement was commendable, Clamp had a relatively modest 17% success rate in selecting the optimal move, illustrating the intricate balance between computational efficiency and decision quality. In essence, it highlights the trade-off that many pattern-based systems encounter.

The case of Morph (Walker & Levinson, 2004) in the context of PBR systems provides valuable insights into the challenges and adaptability of an abductive approach. its noteworthy achievement was its ability to enhance pattern extraction efficiency over multiple games. However, Morph also faced persistent challenges when it came to understanding how to successfully conclude games and secure victory. This particular limitation highlights a key aspect of abductive PBR: the need for a comprehensive and well-structured knowledge base. It's not enough to identify patterns; the system must also know how to effectively apply these patterns to achieve a winning outcome.

The adaptability of an abductive PBR approach depends on several factors, including the quality and diversity of the training data, the sophistication of the pattern extraction algorithms, and the system's ability to derive actionable strategies from identified patterns. Over time, with access to more comprehensive and diverse data, an abductive PBR system may become increasingly adept at adapting to different gameplay scenarios and improving its overall performance.

The case of CHUNKER (Berliner & Campbell 1984) and SUPREM (Berliner & Ebeling, 1990) represents a perfect example of inductive (PBR). These systems, in contrast to purely abductive approaches, overcame the inherent challenges and exhibited the capability to play complete games at a master's level. Their achievement was underpinned by predefined pattern recognition, which essentially means that they were initially designed based on a foundation of hypothetical expert knowledge. The success of CHUNKER and SUPREM suggests the potential of a PBR approach in addressing complex gameplay problems and problem-solving in general. In the case of these systems, predefined patterns serve as a form of knowledge that guides their gameplay strategy. The study outlined in (Ganguly et al., 2014), hints at the tantalizing possibility of constructing a fully knowledge-based algorithm. This is contingent on the feasibility of implementing an automatic knowledge extraction process.

3.4 Theoretical Model Evaluation

The research in this area draws significantly from the work of Chase and Simon (Chase & Simon, 1973) and Gong et al. (Gong et al., 2015), who conducted studies focusing on the perceptual abilities of chess players. Their investigations aimed to gain insights into how players mentally perceive chess board positions. The key finding from their studies is that a

Table 1: Knowledge-based chess systems.

Knowledge representation	Reasoning and generalization	Goal	Game stage	Results
Relative piece position + attack defense relation (BRATKO et al., 1978)	Implementation of expert hypothesis on endgame	Elicitation of pattern-based representation for endgames	End game	Evidence that a more knowledge-based approach is required
Fuzzy logic using fixed patterns: material king protection, pawn structure (Lazzari, 1996)	customizable weighted function for classification	Human theory-based classification for board position evaluation	Middle game	Proof that joining CBR and fuzzy logic is valuable for the teaching of high-level chess strategies
Structural features representation with PCA (Sinclair, 1998)	K nearest neighbors based similarity for move selection	Quality of Results Assessment	Full game	The need to balance between quality and the number of results
Exact piece positions + attack/defense relation (Ganguly et al., 2014)	Piece's mobility and connectivity approximation for move selection	Search time sizing	Full game	Low runtime overhead
Exact piece position (Hesham et al., 2021)	potential moves for both players and their opponent	Downsizing the search space	Full game	Enhanced playing performances (using minimax algorithm and alpha-beta pruning)
Explanation Patterns (Kass, 1990, Kerner, 1995)	Pattern instantiation	chess expert system for game evaluation	Full game	Comprehensive board position evaluation
Construction of fixed-sized chunks based on attack, defense, and color (Cook, 2008)	The exact correspondence of board positions chunks	Investigating decisive chunk size and composition	Middle game	4 to 5 pieces attack defense chunk tend to be more decisive in move making
Abstract predefined pattern Berliner & Campbell 1984)	Guided pattern generation	Board position evaluation	Pawn endgame	Evaluation of entire board configurations based on predefined abstract pattern libraries
Predefined pattern (Berliner & Ebeling, 1984)	Interim goals and their defining pattern for recognition	Pattern-based advice for guiding Alpha-beta search	Full game	Playing a full game at a master's level

player's level of expertise is closely linked to their chunking abilities. This chunking process involves players breaking down a complex board position into manageable and meaningful "chunks."

These "chunks" are essentially cognitive units that encapsulate specific patterns and structures within the chessboard. Players establish these chunks based on various criteria, including pawn structures, color, attack and defense relationship, and local proximity. The chunking process allows players to efficiently process and remember complex board positions. They recognize recurring patterns and structures, which simplifies decision-making during a game.

León-Villagrá and Jäkel (Leon-Villagra & Jakel, 2013) have made contributions to this body of knowledge. Their research indicates that chess players do not rely on visual memory alone to think and remember game situations and features. Instead, players tend to think more abstractly, focusing on the underlying structures, patterns, and relationships between pieces. This abstract approach to thinking enables them to generalize their knowledge and apply it to a broader range of situations, ultimately contributing to their expertise.

The different implementations of these cognitive processes serve to answer **RQ1**, they underline the adaptability of CBR systems, promote the applicability of PBRs, and shed light on the relationship between case bases and pattern base extraction. Case bases serve as valuable sources of information that can potentially lead to knowledge base extraction. They provide the raw material from which generalizations and patterns are derived, ultimately contributing to the development of a knowledge base that enables the system to reason, strategize, and make decisions based on past experiences and expertise.

Here's how this connection works:

Case Bases as a Source of Cases: Case bases store collections of specific cases, each comprising a problem and its corresponding solution or outcome. These cases represent instances of real-world situations, often related to a particular domain, such as chess.

Generalization of Cases: In PBR, the process of generalization involves identifying patterns or commonalities among a set of cases. These patterns could be certain strategies, tactics, or recurring themes that emerge from analyzing multiple cases. The goal is to extract generalized rules or patterns from these individual cases.

Knowledge Base Extraction: The generalized patterns or rules extracted from the case base can be viewed as a form of knowledge base. These rules represent the distilled wisdom and expertise contained within the individual cases. They offer guidance and strategies for addressing similar problems or situations in the future. In essence, the knowledge base is created by summarizing and codifying the general principles that underlie the cases.

Application to New Problems: Once a knowledge base is constructed from the case base, it can be used to tackle new, previously unseen problems. When a new problem arises, the system can consult the knowledge base to identify relevant rules or patterns that apply to the current problem. This allows for informed decision-making and problem-solving.

Most advanced neural-network-based chess programs (He et al, 2018), (Sabatelli et al, 2018), share the overarching concept of learning from data and applying this learning to new problems, to evaluate positions and calculate strategies. Yet it has been proven that neural networks, particularly those involved in deep learning, tend to forget previously learned information upon learning new information (Babakniya et al, 2023). This phenomenon, known as catastrophic forgetting, is a significant barrier to effective generalization over time.

Psychological studies, particularly those conducted by Dingeman and DeGroot (Dingeman & DeGroot, 1965), have provided intriguing insights into the cognitive processes of players, highlighting the differences between experts and beginners. Key findings from these studies include:

Real-Time Decision-Making: Regardless of their expertise, players are observed to make their move decisions in the here and now, responding directly to the board position before them. This implies that even experts do not rely solely on pre-planned sequences of moves, dispelling the myth that chess experts have every move planned far in advance.

Contextual Analysis: To make these real-time move decisions, players engage in contextual analysis. They carefully evaluate the local situations on the board, identifying those that hold promise for their plans and moves. This emphasis on contextual analysis highlights the significance of a global contextual scan, which encompasses a broad assessment of the game situation.

Capacity for Memorization: Remarkably, Simon and Gilmartin (Simon & Gilmartin, 1973) have found that expert-level players possess an impressive capacity for memorization. They can commit a vast number of different game scenarios to memory, ranging from 10,000 to a staggering 100,000 unique situations. This ability to remember and recognize specific board positions further contributes to their expertise.

This can answer the question of why Morph (Walker & Levinson, 2004) couldn't successfully conclude games and the need for a comprehensive and well-structured knowledge base, thus solving **RQ2**. Retaining the cases that were used to generate rule bases can indeed be considered a constraint imposed to address the issue of rule validity. This approach serves several valuable purposes:

Rule Validation: Keeping the source cases allows for continuous validation and verification of the generated rules. By maintaining the original cases, reasoning systems can periodically check whether the rules are consistent with the actual experiences and expertise contained in the cases. This helps ensure that the rules remain valid and up to date.

Dynamic Adaptation: Cases are real-world instances and, as such, they capture a dynamic and evolving body of knowledge. New cases are added over time as more experiences are gained. By retaining these cases, reasoning systems can adapt and refine the rules as new information becomes available, enhancing the system's adaptability and accuracy.

Handling Exceptions: In complex domains like chess, there may be scenarios or exceptions that rules alone cannot adequately address. The original cases serve as a safety net to handle such exceptions. If a new problem or situation does not fit well with the existing rules, the system can fall back on the cases for guidance.

Explanation and Transparency: Maintaining the source cases offers transparency in rule generation. It allows system users to trace back to the original cases, making it easier to understand how and why specific rules were generated. This transparency can be crucial in critical applications or when users need to trust the system's decisions.

However, it's important to consider the trade-off between the advantages of retaining cases and the associated computational complexity. Managing a large number of cases can be resource-intensive. Therefore, Reasoning systems should strike a balance between retaining enough cases for validation and adaptability while ensuring efficient system performance, thus leading to a new generalization approach, thus treating the suggesting an answer for **RQ3**.

4 CONCLUSIONS AND PERSPECTIVES

In the realm of chess AI, the exploration of CBR and PBR uncovers the nuanced dynamics of knowledge representation and application. CBR mirrors human problem-solving behavior, but its applicability is challenged in the context of chess gameplay. PBR systems demonstrate the delicate balance between efficiency and decision quality, with trade-offs based on the choice of similarity metrics.

Psychological studies shed light on the real-time decision-making capabilities of chess players, offering insights into the cognitive processes that underpin expertise. The research underscores the importance of problem-specific characteristics and adaptability in PBR systems.

CBR systems are renowned for their scalability and are generally more approachable in design compared to rule-based systems. However, in practice, rule-based systems are often preferred over cases due to their greater applicability. Consequently, the challenge lies in automating the creation of rule bases, which can be viewed as a generalization of case bases.

Retaining cases used for rule generation emerges as a valuable constraint to ensure rule validation, dynamic adaptation, handling exceptions, and system transparency. Striking a balance between computational efficiency and the advantages of retaining cases remains a key consideration.

This comprehensive exploration of CBR and PBR in chess AI provides a deeper understanding of the challenges and accomplishments in building intelligent systems (Berramla et al, 2020) that navigate the complexities of chess gameplay and problem-solving in general. It opens doors to further research and development in artificial reasoning.

REFERENCES

Aamodt, A., & Plaza, E. (1994). Case-based reasoning: foundational issues, methodological variations, and system approach. AI Communications, 7(1), 39-59.

Babakniya, S., Fabian, Z., He, C., Soltanolkotabi, M., & Avestimehr, S. (2023). A Data-Free Approach to Mitigate Catastrophic Forgetting in Federated Class Incremental Learning for Vision Tasks. arXiv preprint arXiv:2311.07784.

Berliner, H.J., & Campbell, M. (1984). Using chunking to solve chess pawn endgames. Artificial Intelligence.

Berliner, H.J., & Ebeling, C. (1984). Pattern Knowledge and Search: The SUPREM Architecture. AI.

Berliner, H.J., & Ebeling, C. (1990). "Hitech". In Computers, Chess, and Cognition, T. Marsland and Schaeffer, Eds. Springer-Verlag, New York.

Berramla, K., El Abbassia Deba, J. W., Wu, J., Sahraoui, H. A., & Abou El Hassan Benyamina. (2020). Model Transformation by Example with Statistical Machine Translation. In MODELSWARD (pp. 76-83).

Bratko, I., Kopec, D., & Michie, D. (1978). Pattern-based representation of chess end-game knowledge. Comput.

Campbell, M., Hoane, A.J., & Hsu, F. (2002). Deep Blue. Artificial Intelligence, 134(1-2), 57-83.

Chase, W. G., & Simon, H. A. (1973). Perception in chess. Cognitive psychology, 4(1), 55-81.

Chaslot, G., Bakkes, S., Szita, I., & Spronck, P. (2008). Monte-carlo tree search: A new framework for game AI. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (Vol. 4, No. 1, pp. 216-217).

Cook, A. (2008). Chunk learning and move prompting: making moves in chess. School of Computer Science, University of Birmingham.

Dingeman, A., & DeGroot, A. (1965). Thought and Choice in Chess. Mouton Publishers, The Hague.

Ensmenger, N. (2012). Is chess the drosophila of artificial intelligence? A social history of an algorithm. Social Studies of Science, 42(1), 5-30.

Ganguly, D., Leveling. Johannes, Jones. Gareth J. F. (2014). Retrieval of Similar Chess Positions.

Gong, Y., Ericsson, K. A., & Moxley, J. H. (2015). Recall briefly presented chess positions and their relation to chess skills. PloS one, 10(3), e0118756.

He, J., Wu, D., Geng, X., & Jiang, Y. (2018). DeepChess: End-to-End Deep Neural Network for Automatic Learning in Chess. International Conference on Artificial Neural Networks (ICANN), Springer LNCS

Hesham, N., Abu Elnasr, O. & Elmougy, S. (2021). A New Action-Based Reasoning Approach for Playing Chess. Computers, Materials & Continua. 69. 175-190. 10.32604/cmc.2021.015168.

Kass, A. M. (1990). Developing Creative Hypotheses by Adapting Explanations. Technical Report #6, p. 9. Institute for the Learning Sciences, Northwestern University, U.S.A.

Keane, M. T., & Kenny, E. M. (2019). How case-based reasoning explains neural networks: A theoretical analysis of XAI using post-hoc explanation-by-example from a survey of ANN-CBR twin-systems. In Case-Based Reasoning Research and Development: 27th International Conference, ICCBR 2019, Otzenhausen, Germany, September 8–12, 2019, Proceedings 27. Springer International Publishing.

Kerner, Y. (1995). Learning Strategies for Explanation Patterns: Basic Game Patterns with Application to Chess. In Proceedings of the First International Conference, ICCBR-95, Lecture Notes in Artificial Intelligence, vol.1010.

Lazzari, S. G., & Heller, R. (1996). An Intelligent Consultant System for Chess. Computer Educ.

Leon-Villagra, P., & Jakel, F. (2013). Categorization and Abstract Similarity in Chess. Proceedings of the Annual Meeting of the Cognitive Science Society, 35(35).

Mardia, K.V., Kent, J., & Bibby, T.J.M. (1979). Multivariate Analysis. Academic Press.

Ongsulee, P. (2017). Artificial intelligence, machine learning, and deep learning. 2017 15th International Conference on ICT and Knowledge Engineering.

Plaat, A., Schaeffer, J., Pijls, W., & De Bruin, A. (2014). A new paradigm for minimax search. *arXiv preprint arXiv:1404.1515*.

Qvarford, J. (2015). A CASE-BASED REASONING APPROACH TO A CHESS AI USING SHALLOW SIMILARITY. Degree project in Computer Science Basic level 30 higher education.

Reason, J. (1990). Human Error. Cambridge University Press, New York.

Russell & Norvig. (2003). The intelligent agent paradigm.

Sabatelli, M., Bidoia, F., Codreanu, V., & Wiering, M. (2018). Learning to Evaluate Chess Positions with Deep Neural Networks and Limited Lookahead. https://doi.org/10.5220/0006535502760283

Sajo, L., Kovacs, G., & Fazekas, A. (2008). An application of multi-modal human-computer interaction — the chess player Turk 2. 2008 IEEE International Conference on Automation, Quality and Testing, Robotics.

Sato, N., & Ikeda, K. (2016). Three types of forward pruning techniques to apply the alpha beta algorithm to turn-based strategy games. In 2016 IEEE Conference on Computational Intelligence and Games (CIG) (pp. 1-8). IEEE.

Shank, R. (1982). Dynamic Memory: A Theory of Learning in Computers and People. Cambridge University Press, New York.

Simon, H., & Gilmartin, K. (1973). A simulation of memory for chess positions. Cognitive Psychology.

Sinclair, D. (1998). Using Example-Based Reasoning for Selective Move Generation in Two-Player Adversarial Games. Proceedings of the Fourth European Workshop on Case-Based Reasoning (EWCBR-98), pp. 126–135.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., ... & Hassabis, D. (2017). Mastering chess and shogi by self-play with a general reinforcement learning algorithm.

Sun, R. (1995). Robust reasoning: integrating rule-based and similarity-based reasoning. Artificial Intelligence.

Walker, D., & Levinson, R. (2004). The MORPH Project in 2004. ICGA Journal, 27(4), 226-232.