Aesthetic Chess Puzzle Generation using Reinforcement Learning

Deep Mehta
New York University
New York, USA
dmehta@nyu.edu

Jayesh Gaur
New York University
New York, USA
jjg9777@nyu.edu

***Abstract*—Chess is one of the well-studied OTB (over-the-board) puzzles because of its simple rule-set and finite states. With DeepMind taking over the SOTA results after training itself with self-learning in 2017, there has been little progress in the domain. While we built very strong chess engines to play the game, the use of AI to build Chess Puzzles has always been an overlooked topic. Most puzzles available today are filtered through the database of all played games by looking for particular patterns. Chesthetica, a chess puzzle generator, builds puzzles from scratch, but only works for the "Mate in X moves" category. Moreover, it was worked upon before the rise of Deep Learning in Games. It follows a very traditional approach with limitations that we look to overcome. We propose a novel solution where we leverage the available computational resources as well as incorporate the Aesthetic heuristics to train an agent for generating "beautiful puzzles" using a self-learning model.**

I. Introduction

Chess has been one of the board games to receive immense attention from a research perspective to test algorithms against. The victory of IBM's Deep Blue against the former chess champion, Gary Kasparov, changed the perspective of how the game was played [1]. A move that surprised everyone was a Knight sacrifice by Deep Blue for a compensation of a pawn and positional advantage [2]. Until now, sacrifices were rarely seen in professional chess plays unless there was a clear intention. After this event, tremendous progress was seen in the use of computers for chess and its analysis.

In 2017, DeepMind announced its new engine, *AlphaZero*, that used Neural Networks combined with MCTS to further improve the difficulty of chess engines [3]. While the previous methods involved heuristics formulated and tested by masters, experts and previous databases, *AlphaZero* made use of self-learning. The only input to the agent was the basic rules of chess, and the training was done through self-learning by playing over 44M games against itself. AlphaZero defeated Stockfish - the best chess engine in 2017 - by a score of 28W-0L-72D [4]. Since 1997, no human has been able to beat a computer chess engine.

However rich the advancement in state-of-the-art chess engines is, there is an area of chess that is overlooked by these engines - Chess Puzzles. A chess puzzle refers to a particular state in chess that consists of a solution that gives a strong advantage to one of the players. Solving chess puzzles is considered an inseparable part of Chess practice. As we will find further, a good chess puzzle can be evaluated based on a few heuristics. If we can come further to such an extent with chess engines, why is it so difficult to build chess puzzles?

To understand this, we need to go through what makes a puzzle. For the basics, a puzzle needs to have: A) Focus on a particular theme for analysis of the interaction between chess pieces B) Provides advantage to the solver over the opponent C) A single solution regardless of what the opponent plays D) No unnecessary moves - the solution has to be in the minimum moves possible

Azlan Iqbal summarizes all the puzzle essentials into 21 points in their work [5]. Such puzzles are either manually composed by chess composers or searched through a huge database of played games following particular patterns. What makes it so hard for a computer to do the same is that it is considered as a form of art, and with any other art model, AI is far from achieving human-like behaviour. Whether chess qualifies as a form of art or not is out of scope for our goal, but a lot of research is done over the topic [6].

II. Related Work

While a chess puzzle can be defined by the above-mentioned objectives, a puzzle also has an Aesthetic aspect, or in the chess world, "beauty of chess". For a computer program, every move is a move, regardless of how ugly it may be for humans. This limits generation of a quality puzzle with the use of computers. Beauty or aesthetics is a highly subjective heuristic and involves subjective responses that are affected by personal tastes. However, we attempt to encapsulate them in a few objectives, based on the common patterns observed throughout the puzzles [5] [7]: A) violate basic heuristics - sacrifice, silent key moves, disguise, B) emphasize on the most powerful piece, C) play moves that prepare other pieces for a follow-up

An Aesthetic puzzle is not about the randomness or difficulty of the puzzle, but the set of moves that are not so natural under regular positions. Moreover, an Aesthetically pleasing puzzle may not occur in real gameplay.

So far, we have only seen one advancement in the domain of chess puzzle composing computer program - Chesthetica [8]. It looks at the Aesthetic heuristics of the game and uses an approach called Digital Synaptic Neural Substrate (DSNS) [9]. DSNS allows the AI to leverage computational aesthetics. It generates multiple positions using the puzzle essentials which are then evaluated against the theme and these heuristics - while many assume DSNS to be a Neural-Network based approach, it is not. It combines data from unrelated domains into strings to enable regeneration of creative elements [8]. However, Chesthetica comes with a few limitations:

- Uses traditional approaches: During the time Chesthetica was being developed - 2006, Deep Learning approaches were uncommon in the game domain.
- Supports one type of puzzle: These puzzles are targeted for a specific kind of puzzle - Mate in X moves, where X is between 2 to 5. It does not consider puzzles where the goal of the puzzle may be to win material or improve the position immensely.
- White has a clear advantage: One common observation with Chesthetica puzzles is that White always has a material advantage over black.
- Puzzles have low stakes: Most puzzles involve a position where White is winning. A good puzzle is where one wrong move from white can turn the game in black's favor, which frequently occurs in real games.

Thus, while there is no definitive definition for a great puzzle, it can be written as a perfect combination of Composition (difficulty) and Aesthetics (beauty).

III. Method

Chess engines like AlphaZero leverage the computation power to rapidly evaluate multiple search states. On the other hand, solely relying on Aesthetics limits the scope of the puzzles. We can leverage the best of both worlds to build a reverse chess puzzle builder.

To begin with, we first have to decide how we decide the Aesthetics. For the same, we select the most common themes as heuristics along with the classical evaluations.

1) Classical position evaluation - We use Stockfish's open-source evaluation to find the zero-sum advantage [10]
2) One heuristic each for: sacrifice, weakest piece, fork, pin, disguise [7]
3) One heuristic each for: economy, position sparsity, number of moves

We use 2) to retrieve the theme of the puzzle and using 3), we can further optimize the puzzle by reducing the irrelevant pieces and moves. As we develop further, we can modify these heuristics to adapt to diverse themes.

For the exploration part, we plan to use AlphaZero's self-learning approach, however, in reverse order. i.e. We will start with a random position such that one player is at an advantage and find the best "undo" moves within the range of 5 until the classical evaluation of the position is averaged [10]. Using this, we can generate a set of possible legal lines (set of moves) which when played in order, makes a puzzle. Once we get a set of such lines, we can trim out the irrelevant moves and filter out the good puzzles using the puzzle objectives mentioned above.

Thus, to summarize the process:

- Begin with an agent that knows the rules of chess - similar to AlphaZero, but with an additional rule: Player can "un-capture" an opponent piece that is not on the board*
- Start with a random position** such that one color has a better classical evaluation
- Generate all possible states constrained by the number of moves and evaluate (filter) them using the heuristics
- With the filtered puzzles, optimize the weights for each heuristic allowing the agent to exploit the knowledge of the previous positions

* *In a general chess engine, this will explode the search space. However, these puzzles are limited to only a few moves allowing to explore such states.*

** *A random position allows to explore a wide variety of puzzle themes. If we want "mate in X" puzzles, we start with a position where black is already in a checkmate.*

Additionally, we can make use of the puzzle databases [11] [12] [13] to search for interesting heuristics that can be further used to optimize the puzzle generation.

IV. Conclusion

Through this project, we aim to shed light on the overshadowed domain of Chess and the use of computers in creating chess puzzles. High computation and immense studies over the Aesthetics of chess puzzles allows us to build an agent that can leverage the resources to generate great puzzles. These puzzles can be analyzed to further explore the beauty of chess and unveil some interesting ideas.

If this project is successful, it will allow an AI to solve the art of chess compositions. Similar approaches can be applied to other games like Go, Checkers, Shogi, etc.

References

[1] M. Campbell, A. Hoane, and F. hsiung Hsu, "Deep blue," *Artificial Intelligence*, vol. 134, no. 1, pp. 57–83, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370201001291

[2] H. Editors, "Deep blue defeats garry kasparov in chess match," 1997. [Online]. Available: https://www.history.com/this-day-in-history/deep-blue-defeats-garry-kasparov-in-chess-match

[3] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, "Mastering chess and shogi by self-play with a general reinforcement learning algorithm," 2017.

[4] MikeKlein, "Google's alphazero destroys stockfish in 100-game match," 2017. [Online]. Available: https://www.chess.com/news/view/google-s-alphazero-destroys-stockfish-in-100-game-match

[5] A. Iqbal, "A systematic and discrete view of aesthetics in chess," *Journal of Comparative Literature and Aesthetics*, vol. XXIX, pp. 53–65, 01 2006.

[6] A. Iqbal and M. Yaacob, "Computational aesthetics and chess as an art form," *Journal of Comparative Literature and Aesthetics*, vol. XXVIII, pp. 49–59, 01 2005.

[7] Y. M. Iqbal, Azlan, "Advanced computer recognition of aesthetics in the game of chess," *WSEAS Transactions on Computers*, vol. 7, pp. 497–510, 05 2008.

[8] A. Iqbal, "A discrete computational aesthetics model for a zero-sum perfect information game," Ph.D. dissertation, 09 2008.

[9] A. Iqbal, M. Guid, S. Colton, J. Krivec, S. Azman, and B. Haghighi, "The digital synaptic neural substrate: A new approach to computational creativity," 2016.

[10] "Stockfish." [Online]. Available: https://github.com/official-stockfish/Stockfish

[11] "Lichess puzzle database." [Online]. Available: https://database.lichess.org/

[12] “Yet another chess problem database.” [Online]. Available: https://www.yacpdb.org/

[13] “Chess blunders - endless database of chess puzzles.” [Online]. Available: https://chessblunders.org/