Producing creative chess through chess engine selfplay

Wolf De Wulf
Computer Science
Vrije Universiteit Brussel
Computational Creativity Lab
wolf.de.wulf@vub.be

Abstract

This article presents preliminary work on a creative chess engine that can be used to produce creative chess games or sequences. The contribution in this article is the creation of a creative chess engine that is then pitted against itself to form a creative system that outputs chess games. The chess engine is an extension to an existing chess engine that consists of forcing the existing engine to play more creative moves. It is in no way an improvement when compared to existing chess engines, even though it is based on the world's best: Stockfish. Letting the supposedly creative chess engine play against itself forms a creative system that outputs chess games. Through analyzing these games it might be possible to discover new chess openings or principles.

Introduction

With the advancement of artificial intelligence we can more and more shift from Turing's question if computers can be seen as machines that can "think" (Turing 1950), to asking ourselves: "Can machines be creative?", or more specifically in this case: "Can chess engines be creative?" Every chess player has their own style of playing, but so do chess engines. For both, creativity can be very important. Bushinsky (2009) reasoned that there is a correlation between strength of play and creativity. Analyzing this and other observations regarding creativity in chess, inclines Bushinsky to conclude that at this point in time, chess engines are even more creative at playing chess than humans are.

The present creative system is an attempt at formalizing and proving what the majority of the chess community believes to be true: that chess engines can be and often are *creative*. The creative chess engine is in no way an improvement when compared to the current state-of-the-art. It is rather an *extension* of the current state-of-the-art – Stockfish – that consists of trying to force the engine to play more creative moves. Note that this means that measures that are able to capture the creativity of a chess move, had to be established. Inspiration was found in the work of Amatzia Avni (1998). Avni analyzed the roots of creativity in human play and reasons on the human creative process when playing chess as well as on general principles that are deemed creative in the game of chess.

Creativity in chess

An important chess concept that one needs to be familiar with to be able to understand creativity in chess, is the concept of *openings*. In chess it is common for players to memorize sequences of beginning moves. These sequences that have been played for a long time and have been analyzed and concluded to be good are called 'openings'. In recent years years, it has become more and more difficult for humans to find new openings, as a great deal has been tried already. Computer engines, on the other hand, are perfect 'analyzing machines'. There are multiple cases of computer engines finding new lines in existing openings or analyzing existing openings and concluding that they are sub-optimal when compared to others. As an example take Bushinsky's *Deep Junior* discovering a new line in the Sicilian defense that consisted of sacrificing three pawns. Also, in Silver et al. (2017), the authors of the well known *AlphaZero* engine explain how their engine analyzed the French defense for 2 hours and then stopped playing it completely as the engine concluded that its sequences of moves are sub-optimal.

As mentioned, chess engines have their own style of playing chess. Bushinsky reasons that the engine style is more objective as they do not bring prejudice to their play. They are said to "play the position" and this style of playing is often called the *concrete style*. Even though chess engines play more objectively and are better at assessing risk, this should not restrain them from playing creatively. An example is the sacrificial move. In the early days of chess engines it was thought to be impossible for them to ever come up with a move that sacrifices a piece with no immediate gain. In 2003, however, Deep Junior sacrifices its bishop in a game in New York against former world champion Garry Kasparov. The strategy of giving up one's own piece, that was formerly thought of as "god's gift to man" (Bushinsky 2009), was suddenly implemented by a machine. This occurrence alone should already spark the question if chess engines can in fact play as or more creative than human players.

According to Bushinsky, creativity in chess is not much different from creativity in other domains. He applies De Bono's (1973) *lateral thinking*, which consists of: nonconformism, provocation (in the positive sense), flexibility, casting doubt, thinking out of the box and transfer. Some of these more obviously applied to chess than others. Amatzia Avni, who is a psychologist and a chess master, links cre-

Proceedings of the 12th International Conference on Computational Creativity (ICCC '21)
ISBN: 978-989-54160-3-5

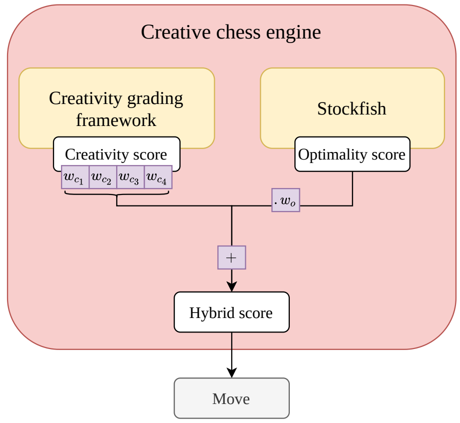


Figure 1: Graphical representation of the structure of the creative chess engine.

ativity in chess to the human's intelligent process of playing chess. He mentions the following four steps: gathering, synthesis, enlightenment and realization. In his book, called 'Creative chess', he explains some general creative elements in the game of chess: nonstandard positioning or functions of chess pieces, the removal of one's own piece, the breaching of theoretical principles, etc.

From the above, we establish creativity measures that try to capture creativity in chess moves. A chess move always takes into count the complete chess board. This means that if we consider a move on a chess board and we then consider that move on another chess board on which only a single piece has a different position, the two moves are different moves. The creativity that these measures capture is debatable. For the moment, the creative chess engine only uses the four measures below. It should be noted that new measures can easily be added.

Measure 1 (Unknown move)**.** *A chess move may be creative if it has never been played (in a tournament).*

Measure 2 (Low winrate move)**.** *A chess move may be creative if it has been played (in a tournament) and it has a low win to loss ratio.*

Measure 3 (Sub-optimal capture)**.** *A chess move that captures a piece may be creative if the captured piece is not the most valuable piece that could have been captured.*

Measure 4 (Sacrifice)**.** *A chess move may be creative if it is a sacrifice. A sacrifice is a move that when played, allows the opponent to capture an undefended piece.*

The first measure relates to novelty. A move that has never been played before, should be considered creative. The second measure is closely related to openings. When a move has been played before but its win to loss ratio is very low, the chess community and opening theory will evolve to consider it a bad move. However, this phenomenon does not imply that it is a bad move, as it could be that there exists some sequence of moves in which the move can be good. This sequence has just not been found yet. Therefore, playing such a move that is considered to be bad, can be creative. The last two measures are more closely connected to chess theory than to knowledge of played games. They relate to De Bono's *provocation* and *casting doubt* as they involve playing moves that lose material.

The creative chess engine

Figure 1 gives a graphical representation of the structure of the engine. The main job the creative chess engine (De Wulf 2021) has to perform is to calculate three types of scores: optimality scores, creativity scores and hybrid scores. When the creative chess engine plays a game, at each turn it calculates the hybrid scores for all the legal moves and it then plays the move that corresponds to the highest hybrid score.

Optimality scores These are the Stockfish scores but normalized by dividing by 100. An exception is when a move M is part of a sequence S that forces a *checkmate*. In this case its non-normalized score is calculated as follows:

$$score(M) = 999999 - length(S) * 100 \qquad (1)$$

By doing this we force the creative chess engine to play sequences that end in checkmate when they are possible but still allow it to choose between these sequences. For example, when one sequence could be considered more creative than the other, we allow the creative chess engine to choose the more creative one.

Creativity scores When a move complies with one of the creativity measures, its creativity score is awarded a weight w_{c_i} specific to that measure. A move's creativity score is hence the sum of weights w_{c_i}, for all the measures it complies with.

Hybrid scores The hybrid scores are the creative chess engine's combination of the optimality and creativity scores. The engine calculates a move's hybrid score by adding its creativity score, which is itself a summation of weights that correspond to the move's creativity, and its optimality score, multiplied by a weight w_o.

Proceedings of the 12th International
Conference on Computational Creativity (ICCC '21)
ISBN: 978-989-54160-3-5

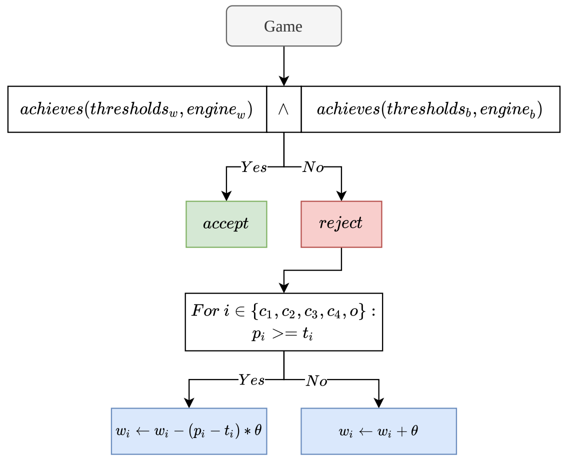


Figure 2: Internal evaluation and transformational creativity strategies.

Modelling the system

There are multiple possibilities for the output of a creative chess system. One could create a system that outputs single chess moves, sequences of moves or even complete games. In this case, the creative chess system is defined to be a system that outputs *complete* and *legal* chess games that were played by two of the creative chess engines. By modelling the creative chess system using the *Creative Systems Framework (CSF)* (Wiggins 2006a; 2006b), it becomes possible to compare strategy games like chess to more artistic and problem solving domains as creative endeavours.

We already defined the conceptual space C of the system to consist of all the possible *legal* and *complete* chess games that are produced by letting two of the creative chess engines play against each other. The universe U that contains C could then, for example, contain: partial games, games played by humans, games played by other chess engines, etc. The set of rules T that allows for the traversal of U around C, in the case of the creative chess system, consists of rules that define the values of the weights that correspond to each of the four creativity measures that were mentioned in the previous section, as well as the value of the optimality weight. We also need to be able to separate good concepts from bad concepts, or in our case, good chess games from bad chess games. For this, Wiggins introduces a set of rules E, written in L, which may be used to accept or reject concepts in terms of their quality. These rules are discussed in the next section.

In the case of the creative chess system, transformational creativity can be used to transform both R and T. Transforming the system's R can be done in two ways. Firstly, we can transform the 'selfplay' constraint to an 'otherplay' constraint. To do this, we need to pit the creative chess engine against any other chess engine (e.g. Stockfish or AlphaZero). Since every engine has its own style, we can expect the outputted games to be very different after transforming the system in such a manner. One could reason that after doing this transformation, the conceptual space C will still be the same. However, there surely are games and sequences of moves that some engines will never play. Therefore, we can safely state that there is a different conceptual space for each version of the creative chess system that uses a different chess engine. These conceptual spaces will almost never be completely indifferent but they will also never be completely equal. Secondly, we can change the rule that requires the outputted games to be complete games. Doing this can again be very interesting, as it could allow us to find chess principles that are not specific to the game itself but can be used in any game. Lastly, transforming the system's T can also be done in two ways. Firstly, we can change the creative chess engines' weights. This method of transformational creativity is natively implemented in the creative chess system and is discussed in the next section. Secondly, creativity measures can be added or removed from the creative chess engines. This method is not implemented in the system but can be seen as possible future work.

Evaluation

Figure 2 gives a graphical representation of the evaluation strategy of the creative chess system. In what follows, the evaluation rules are discussed in detail. Subsequently, their link with the system's transformational creativity strategy is explained.

Internal evaluation The internal evaluation strategy of the creative chess system should consist of rules that define which outputted *games* should be accepted as valuable and which should be rejected. To be able to come up with such rules, we first need to define *what* we find to be valuable in chess games. In this case, inspiration was found in the architecture of the creative chess engine. We have already defined creativity measures for chess moves. On top of that, we also know what moves are supposed to be very *optimal* moves, through the output of Stockfish. By letting the creative chess engine keep track of which moves it plays, we can, once a game is finished, get the percentage p_i for each of the move types to the total number of moves that were

Proceedings of the 12th International
Conference on Computational Creativity (ICCC '21)
ISBN: 978-989-54160-3-5

played. Also, since two creative chess engines are playing against each other, we get two personal percentage collections per game. Once we have those, writing rules that determine which outputted games are valuable, according to what we defined to be valuable, merely consists of writing rules that verify whether the percentages p_i are above certain thresholds t_i. Each type of move can have a separate threshold, allowing us to direct the system towards games that consist of a specific composition of moves of our liking. Anyone that is familiar with Ritchie's criteria (Ritchie 2007) will notice that this strategy of *achieving* thresholds is very similar to some of the criteria in Ritchie's model. As mentioned before, in this document the focus lies on the CSF but in future work, due to this resemblance, Ritchie's criteria could be implemented to further evaluate the creative chess system's creativity.

Even though this strategy of achieving thresholds as evaluation is a simple one, it can still have its nuances. We can, for example, require that only one of the creative chess engines achieves all of its thresholds while the other engine only needs to achieve some. Note that, if one of the engines achieves all of its thresholds while the other engine only achieves its optimality threshold t_o, the outputted game could still be a valuable one, since one of the engines played in a valuable way, while the other was not utterly useless.

External evaluation Evaluating the outputted chess games externally can be done similarly to the internal strategy. Instead of letting the creative chess engines explicitly count different types of moves, the outputted games can be analysed by a, preferably high-rated, chess player. Multiple approaches can be implemented when doing this. A first one consists of going over each and every move and each time asking the external evaluator for their opinion on the optimality and the creativity of the move. Aggregating the results for each move then results in measures similar to the percentages that are calculated by the creative chess engines. A second, more simple, approach, consists of letting the external evaluator analyse the game by themselves and after that asking them to rate the optimality and the creativity of the game as a whole, on a scale from 1 to 10. In both cases, accepting and rejecting can again be done by comparing the results to specific sets of thresholds.

Transformational creativity Closely related to the evaluation of a creative system is Boden's transformational creativity (Boden 2004). When a system determines an output to be invaluable, we can let that system *evolve* into a new system that is more likely to produce output that *is* valuable. Boden originally only described transformational creativity to be changing the rules in R, which means: changing a system's conceptual space C. Wiggins (2006a; 2006b) added to that that transformational creativity can also be applied to a system's T, which means: changing a system's strategy to explore its conceptual space C. In the case of the creative chess system, transformational creativity has only been implemented to transform T. As mentioned before, this can be done in two ways: by changing the values of the creative chess engines' weights and by changing the creativity measures for the creative chess engines' moves. The creative chess system only implements the former.

Each time the system produces a game that is rejected by the internal evaluation strategy, the weights w_i of every type of move are updated as follows:

- If the corresponding threshold t_i was achieved, the weight is subtracted a fraction of a given θ that is proportional to the surplus of the percentage p_i to the threshold t_i:

$$w_i \leftarrow w_i - (p_i - t_i) * \theta \qquad (2)$$

- On the other hand, if the corresponding threshold was not achieved, the weight gets θ added to it:

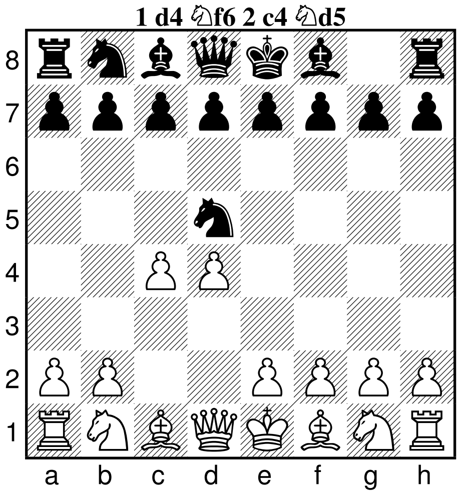
$$w_i \leftarrow w_i + \theta \qquad (3)$$

Results

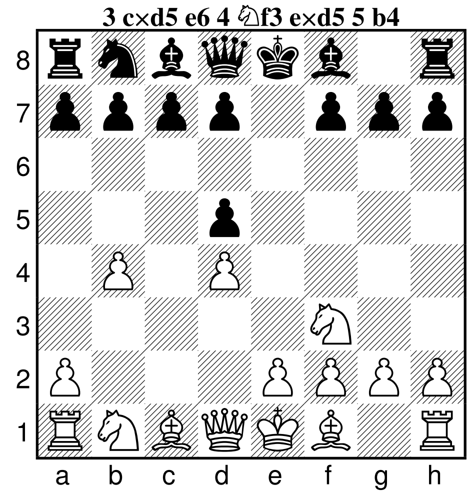
Figure 3 depicts two of multiple interesting positions that occurred in a game that was accepted by the creative chess system. In Figure 3a, the black engine sacrifices its knight with its second move. The creative chess engine classified this move as both an unknown move (Measure 1) and a sacrifice (Measure 4). In Figure 3b, the white engine sacrifices a pawn with its fifth move. The creative chess engine classified this move as both a low-winrate move (Measure 2) and a sacrifice.

This game and many others were analyzed and evaluated by a 2000-rated chess player. In their opinion, the games were creatively played but a recurring comment was that the engines tend to play a lot of sacrificial moves. While these moves can be creative, the context in which they are played is very important. There is also a big difference when humans play sacrificial moves compared to when such moves are played by engines. Computers are better at risk-assessment than humans are. We can not calculate a lot of steps a head, which makes playing sacrificial moves very precarious. Important to note is that, during the first few moves of a chess game, there are lots of positions that have been played before and thus will be stored in chess databases. In all of these positions, there is always a small probability of finding a position that has not been reached before or can be reached by playing a move that, most of the time, resulted in a lost game. However, once the game reaches its 8th to 10th move, a combinatorial explosion happens, making every move result in a position that has never been reached before. Of course, in these positions, firstly, low-winrate moves do not exist and secondly, since every move is technically an unknown move, 'real' unknown moves do also not exist. However, the fact that the moves that correspond to these two measures have a low occurrence rate does not make them useless. On the contrary, the combination of the two with the two other, theory-based measures, which naturally have a higher occurrence rate, can result in creative games. The external evaluator agreed with this statement and stated themselves that finding other, theory-based measures, to add to the system is possible and could, in combination with the other measures, improve the outputted games.

Proceedings of the 12th International Conference on Computational Creativity (ICCC '21)
ISBN: 978-989-54160-3-5



(a) Black sacrifices knight with *unknown* move 2. . . ♘d5.



(b) White sacrifices pawn with *low-winrate* move 5 b4.

Figure 3: Example accepted game, produced using:

- $thresholds_b = [0.04, 0.0, 0.04, 0.2, 0.7]$
- $thresholds_w = [0.0, 0.02, 0.04, 0.2, 0.7]$
- $weights_w = [3.0, 40.57, 10.52, 4.22, 4.51]$
- $weights_b = [31.78, 2.99, 10.21, 2.52, 3.38]$

Conclusions

A creative system that allows for generating creative chess games is presented. In recent years, chess has evolved to a phase where new principles and new openings are very hard to find, when looking for them is done by humans. In that aspect, the creative chess system can be a very handy tool to try to let chess engines do the looking for us. The resulting games can be analyzed by the system internally but for humans to find principles in them, analyzing them ourselves should be valuable. The system has an internal evaluation strategy that allows it to evaluate itself and learn to improve by directing its outputted games towards certain compositions of moves. The system was also externally evaluated, with very interesting results. The outputted games were found to be creative but the sacrificial nature of the chess engines was frowned upon. Sacrificing a lot of pieces can be interesting for chess engines as they are much better at risk-assessment than humans are. However, if humans are to learn new principles from the outputted games, the external evaluator stated that they should contain less sacrifices. In human play, sacrificing pieces is a very scary technique, as we humans can not easily calculate multiple moves ahead. As mentioned in the sections on transformational creativity, reducing this sacrificial nature of the chess engines can be done in two ways: making the corresponding weights smaller, or swapping out the sacrificial measure completely. Both are viable options. With respect to future work on the creative chess system, the latter should be the most interesting. By working together with professional chess players, multiple new creativity measures could be found and implemented in the system.

References

Avni, A. 1998. *Creative chess.* London: Everyman chess.

Boden, M. 2004. *The Creative Mind: Myths and Mechanisms.* London: Routledge.

Bushinsky, S. 2009. Deus ex machina — a higher creative species in the game of chess. *AI Magazine* 30(3):63.

De Bono, E. 1973. *Lateral thinking : creativity step by step.* New York: Harper Colophon Books.

De Wulf, W. 2021. Producing creative chess through chess engine selfplay. `https://github.com/wulfdewolf/creative-chess-producer`.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:76–99.

Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; Lillicrap, T. P.; Simonyan, K.; and Hassabis, D. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *CoRR* abs/1712.01815.

Turing, A. M. 1950. Computing machinery and intelligence. *Mind* LIX(236):433–460.

Wiggins, G. A. 2006a. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458. Creative Systems.

Wiggins, G. A. 2006b. Searching for computational creativity. *New Gener. Comput.* 24.

Proceedings of the 12th International Conference on Computational Creativity (ICCC '21)
ISBN: 978-989-54160-3-5