 doi:10.20944/preprints202204.0227.v1

Allowing Differently Strong Chess Players to Play at An Equal Level

Arend Hintze[1,2*]

1 MicroData Analytics, Dalarna University, Falun, Dalarna, Sweden
2 BEACON Center for the Study of Evolution in Action, Michigan State University, East Lansing, Michigan, USA

* ahz@du.se

Abstract

Chess is an interesting game for artificial intelligence research and an entertaining hobby and sport for a growing number of people. However, humans differ greatly in their ability to play. Typically, the Elo rating is used to determine a player's skill and to predict who will win. When differences in Elo are too great the weaker player is almost guaranteed to lose. While on one hand, the Elo rating allows players to be matched to equally well-playing opponents, it also to some degree restricts the games to be played between equally strong opponents since otherwise the result is known beforehand. Here a handicap system where stronger players remove pieces at the start of the game is evaluated. Specifically, the effect each removed piece or combination of pieces have on a player. Interestingly, pieces do not always reduce the Elo of a player by a predefined amount, but their effect depends strongly on the player's current Elo. The results presented here are from playing the computer engine Stockfish because data about humans playing this scheme is limited. However, the results make direct predictions about the effect of piece removal on Elo.

Introduction

Chess is a fascinating game for humans and a challenge to artificial intelligence research [1, 2]. Recently the TV mini series "The Queen's Gambit" sparked an even higher interest in the game [3–5]. Even the recent Covid-19 pandemic with the need to stay home longer [6] increased the popularity of the game – even my children want to play more chess. However, the outcome of a chess game is easily predictable by the Elo ranking of players [7]. Simply put, players are assigned a number representing their strength. The likelihood of one player A winning against player B, given their ratings (R_A and R_B), can be predicted using Eq. 1:

$$E = \frac{1}{1 + 10^{(R_A - R_B)/400}} \tag{1}$$

Based on this prediction and the outcome of an actual match, the Elo ranking of each player becomes adapted. Over time, the Elo score arrives at an equilibrium reflecting a player's ability to play chess.

While it is hard to find objective evidence for this, but playing opponents with a higher Elo is a futile endeavor. Losing might serve as a learning experience, but the certainty of the outcome only depends on the difference in Elo. When playing against

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 April 2022 doi:10.20944/preprints202204.0227.v1

my children, who are relatively new to chess, while my skill is solid, the outcome is clear: I always win. A similar phenomenon should be observable in many other contexts and might lead to frustration and the demand for a "fairer" outcome. Fairness is a relative term here; clearly, the better player should win, but the demand is for a less predictable result, with the weaker player having a chance to win as well.

The game of golf remedies this issue by introducing a handicap. As a result, two players with different handicaps can play the same course competitively, and the player that scores proportionally better given their handicap wins, not the one that needed fewer total strokes. While widely accepted, it might still not be the fairest method [8]. However, applying such a handicap system to chess is desired, but its implementation is not apparent. Unlike golf, where two players play against a given course, in chess, the success of one player depends on the skill of the other; a simple point bonus would not help. Consequently, a different method needs to be devised to make games more equitable.

Two possibilities come to mind, either allowing players different amounts of time for their moves or removing pieces before the start of the game. Both methods are in practice already, and while it is hard to find scientific publications to support this claim, from personal experience, I know that chess hustlers do practice both to encourage opponents to take their chances by suggesting a less predictable outcome. As mentioned, objective data from humans playing these "odd games" allowing an analysis does not exist in sufficient quantities for either option, so any data to estimate how much time or which pieces would bridge an Elo gap needs to be generated using computers playing chess. If removing chess pieces impairs an algorithm's ability to win in the same way as it does a human player is an entirely different issue, however, a similar effect on play strength is assumed here. Therefore, the effect of removing different chess pieces on the likelihood of winning when competing chess algorithms with different strengths are used as a proxy for human games.

The general approach works as follows: The computer engine Stockfish is played against itself but setup with different start conditions. Specifically, at which depth each player should play and which pieces are removed for black and white. These setups will from now on be called *player setup* using a "depth-piece" nomenclature. "4-qp" for example, indicates a player computing four moves deep while stating with queen and a random pawn removed. The outcome of each match results in a probability for each player setup to beat another player setup. Then a tournament between all player setups is performed to determine the Elo rating for each player setup. This will result in a somewhat skewed Elo distribution, as so many player setups in this tournament will play with a handicap. This bias will be corrected by comparing player setups with a given depth and no piece removed to Stockfish playing with a predetermined Elo rating. The result is an estimate of how many pieces should be removed to allow two chess players with different Elo scores to play a game where the outcome is open.

Materials and methods

Removed Pieces

In theory, all pieces but the king (♔) can be removed before a game become impossible to play. However, only leaving the king or the king and a single bishop (♗) or knight (♘) makes winning impossible. Further, removing the queen (♕) or both rooks (♖♖) servery impacts winning chances, while removing a pawn (♙) is expected to have a low impact. Thus, here the removal of the following pieces or combinations of pieces was tested: pawn (♙), two pawns (♙♙), knight (♘), bishop (♗), pawn and knight (♙♘), pawn and bishop (♙♗), rook (♖), rook and pawn (♖♙), both knights (♘♘), bishop and

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 April 2022 doi:10.20944/preprints202204.0227.v1

knight (♗♘), both bishops (♗♗), rook and knight (♖♘), rook and bishop (♖♘), both rooks (♖♖), queen (♕), queen and one pawn (♕♙), queen and knight (♕♘), queen and bishop (♕♗), queen and rook (♕♖), and finally queen bishop and knight (♕♗♘) – icons are provided to simplify interpretations of figures later. The order here loosely follows the order suggested if each piece is assigned a single value (♙= 1, ♘= 3, ♗= 3,♖= 5, and ♕= 9) [9]. However, those values are highly debated [10], and the value of a single piece, among many other factors, depends greatly on its location on the board [11] and what other pieces are still around [12, 13]. Regardless, other combinations or more pieces could be removed, but the assumption is that this range of pieces is sufficient to allow weak players to win against excellent ones occasionally.

Fischer 960

Removing pieces has an effect on the play right from the start. While later gameplay depends strongly on a player's ability to play chess, it can be argued that early play is more dependent on a player's ability to memorize openings [14]. A phenomenon World Champion "Bobby" Fischer already criticized [15]. He thus invented a randomization scheme for the start position, such that learning openings became pointless. Chess algorithms take advantage of opening libraries, so in order to reduce this effect, Fischer 960 random openings were used. Every time a new game was run, one of the 960 Fisher openings was selected randomly.

Stockfish

While there are many different chess engines available, Stockfish is open source, easy to run, and most importantly, the Elo for different depths of computation is known. This will become important later when the results of Stockfish games need to be mapped to Elo ratings. Version 11.0 was used, as it was the most recent available version when the experiment started. However, Elo ranges can only be set from 1400 to 2800, which does not cover weaker players, for which it would this handicap method might be more interesting, allowing them to play against stronger opponents. Thus, the search depth was varied from 1 to 20 (max), and the best move, given the evaluation and search pruning algorithm of Stockfish, was always chosen without time limitation.

All possible matches

Stockfish was configured to play at all 20 different depth in combination with all 21 possible piece removals (None, ♙, ♙♙, ♘, ♗, ♙♗, ♙♘, ♖, ♗♗, ♗♘, ♘♘, ♖♙, ♖♘, ♖♗, ♖♖, ♕, ♕♙, ♕♗, ♕♘, ♕♖, ♕♘♗). This resulted in 420 possible player setups who where played starting with each color 9 times against each possible opponent, choosing a random Fisher 960 start configuration.

Data Analysis

All data analysis was performed using python and the python-chess, numpy, and scipy libraries which included the needed mathematical fitting functions and polynomials [1].

Elo calculations

For all Elo calculations equation 1 was used to determine the prediction about the likely winner E. To change the Elo (δ) of the involved players the following equation was used:

$$\delta = Elo + K(o - E) \tag{2}$$

[1] The code will be made publicly available on Github after the paper has been accepted.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 April 2022 doi:10.20944/preprints202204.0227.v1

Where $K = 2.0$ is a constant defining the effect size each game has and $o = [0.0, 0.5, 1.0]$ is defining the outcome of the game being lost, drawn, or won.

1 Tournament to determine the Elo

After playing all 420 player setups against each other, the likelihood of one winning against the other or drawing is known. From that, the Elo for each player setup needs to be determined so that later the losses in Elo per removed piece can be determined. For that, a tournament is played between all 420 player setups. Each player setup starts with an Elo of 1200. Ten million randomly picked matches are made, using the likelihoods computed before to determine the outcome of each match stochastically, followed by an update of each player's Elo. This method, however, leads to a very biased Elo rating (as high as 40.000, or below -5000, data not shown). The assumption is that here even excellent player setups are matched against very poor performing ones, and 400 of the 420 player setups play with a handicap, making it easy for none handicapped player setups to win overly often. Thus, only matches between player setups not further away than 400 Elo points were allowed, leading to a much more realistic range of outcomes (between 5000 and -2000). These results were corrected later. The tournament was repeated 50 times, and the results were averaged across replicates to reduce the effect of noise.

Elos needed to be adjusted, and for that, various functions needed to be fitted. Standard methods were used.

Stockfish can play with a given strength defined by a preset Elo from a range between 1400 and 2800. Such a preset adjusts a range of engine parameters resulting in a specific play strength. When playing Stockfish 11 with these presets, it will not play like a human with the same Elo, but at least comparably strong. These Elo presets have been tested against a range of other engines whose Elo is known (personal communication with Joost VandeVondele - Stockfish developer [16]).

Results

Players were set up using 20 different depths with 21 different piece removals. All these player setups were played against each other nine times in each color, with start conditions from Fischer 960. This results in a probability estimate for each player set up to win, draw, or lose against any other player setup. This did not differentiate between the different kinds of draws, as the FIDE rules consider a draw, a stalemate, a three-move repeat, or 50 moves of no pawn moves all as a 0.5/0.5 result.

Now the Elo of each player setup needed to be determined. For that, a tournament was played between all 420 setups, starting each with an Elo of 1200. Ten million games were played randomly, sampling two players as long as their Elo was not further apart than 400 points. The outcome of each game was determined randomly by sampling from the previous results. If, for example, one player setup won 15 out of 18 games before, the exact likelihood was applied now. This tournament was repeated 50 times, and results were averaged, resulting in a somewhat strange Elo distribution (see Figure 1).

Here Elo scores rank from 5000 to -2500, which is unrealistic for an actual population of human players. What needs to happen now is to translate or scale these unrealistic Elos into more realistic ones. Since some player setups did not have pieces removed, we can determine the Elo for those by playing them against Stockfish set known Elos. Since Stockfish can only be configured to play within a specific Elo range, only some search depths can be properly matched to Elo ratings; the remaining ones have to be predicted.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 April 2022 doi:10.20944/preprints202204.0227.v1

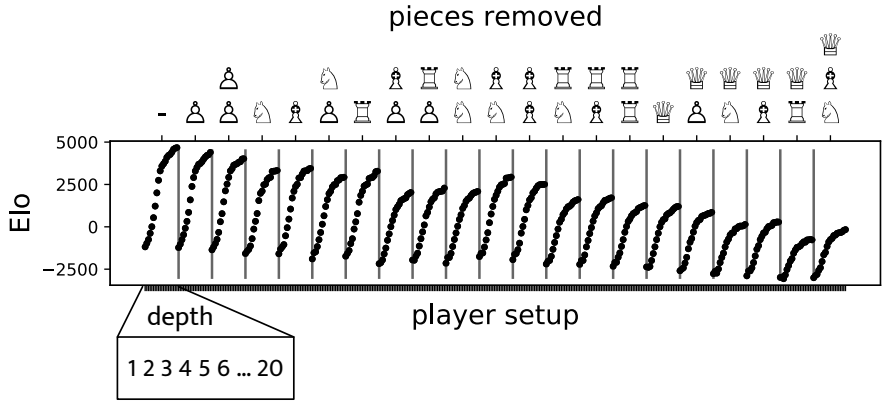

Fig 1. Average Elo scores for all different player setups after 50 tournaments with 10 million games played. Player setups on the x-axis are ordered in sections for all the 21 possible piece removal combinations divided by gray lines. Each section is then itself ordered by move depth. Measurement errors, such as 95% confidence intervals or standard errors, are neglectable due to their vanishing size and are thus not shown.

Once the Elo for all 20 player setups without a piece removed is known, the remaining Elos can be interpolated. How this was done is explained in the following paragraphs:

Stockfish 11.0 playing with all pieces and computing 20 moves running on a single CPU, as used here, has an Elo of 3400 [17]. Consequently, these Elo results need to be mapped to a realistic Elo rating. However, this has to be done for only those player setups that have no pieces removed. Thus, starting with each color and for all the possible 20 depths were matched against Stockfish 11, playing with a predefined Elo of 1400, 1600, 1800, 2000, 2200, 2400, 2600, and 2800. Each match was played 200 times. Using these results, the following sigmoid function predicting Elo (E) from depth (d):

$$E(d) = \frac{1.0}{1.0 + e^{-\alpha d + \alpha\beta}} \tag{3}$$

was fitted for each preset Elo (see Figure 2). The resulting α parameter indicates at which interpolated depth that given Elo wins in 50% of the cases ($P_{win} = 0.5$). At this point a given Elo plays equivalent to a given depths.

Performing all those matches between player setups with different depths versus defined Elos shows how Elo relates to search depth, at least for a few depths. These results were then fitted against a sigmoid to determine equally strong play ($P_{win} = 0.5$). This succeeds for depths 2 to 12. To estimate the relation between depths and Elo outside of this range, polynomials degrees 1 to 3 were fitted against the previous results (See Figure 3).

As it turns out the linear interpolation not only results in a very close fit, it also suggests that the best depth coincides with the highest Elo of 3400. The function defined by this line now allows the translation of player set up with a certain depth into a realistic Elo (which will now be called Elo_{engine}). See Figure 4 A for a mapping between search depths and Elo_{engine}. The more realistic Elo estimate for all player's setups with a defined depth and no pieces removed is known. From the initial tournament the $Elo_{tournament}$ is now known (or predicted) for all 20 depths (see Figure 4 B). When plotting the $Elo_{tournament}$ against the Elo_{engine} for all 20 different depth (see Figure 4 C red dots) another polynomial (degree 7) can be found that explains their relationship.

Using this last function, the remaining 400 player setups can be translated into

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 April 2022 doi:10.20944/preprints202204.0227.v1

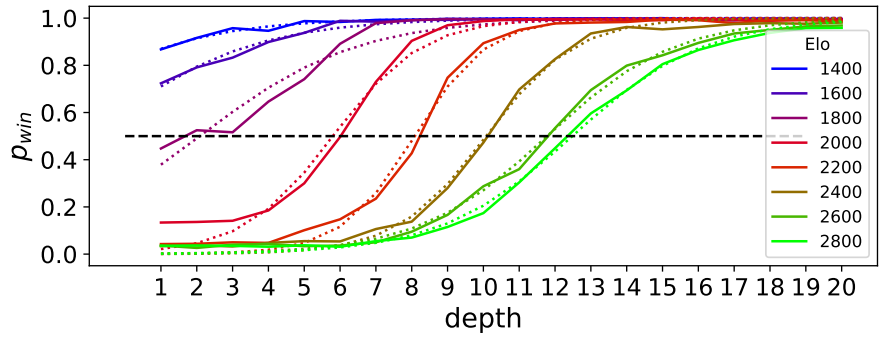

Fig 2. Results from a player setup with 20 different search depths against specific predefined Elo ratings ranging from 1400 to 2800. Each line represents the likelihood to win (P_{win}) of a different Elo rating versus all 20 depths (x-axis). See the legend for which Elo corresponds to each color. Dotted lines show the best fits of sigmoid functions (see Equation 3). The black dashed line indicates where two players would play equally strong. The α parameter from the fit corresponds to those values.

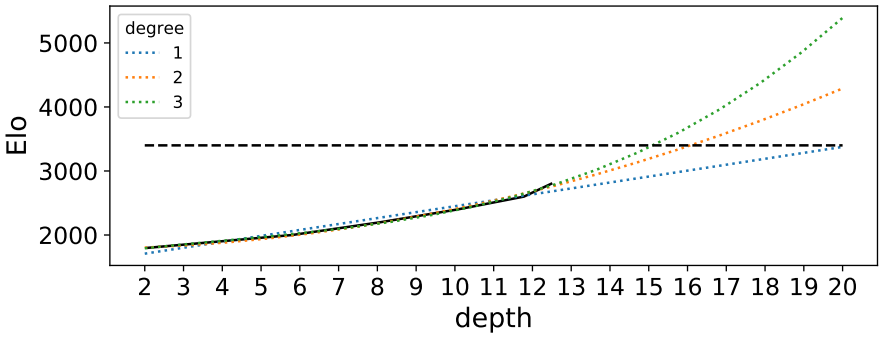

Fig 3. Results of fitting polynomials of different degrees (1-3, see legend for color). Functions were fit to predict the Elo from the engine by depth of the search. Results from Figure 2 are shown in black.

Elo_{engine} values for comparison. These results can be visualized in several ways[2] First the same data as shown in Figure 1 can be compared to properly interpolated Elo_{engine} values (see Figure 5).

Interestingly, the effect of removing a piece changes depending on each piece or a combination of pieces, but the effect also varies depending on the search depth. Thus, removing a piece does not cause a constant drop in Elo but depends on the player's strength. When taking this notion into account, the strength of each player setup can be visualized depending on the pieces removed differentiated for each search depth (see Figure 6).

While this visualization shows the playing strength, the loss in Elo given removed pieces dependent on player strength might illustrate another aspect of the results better (see Figure 7).

Several immediate observations can be made. The loss in Elo, as expected, depends on the removed pieces. The more or, the more valuable the pieces are, the greater the loss on Elo. However, weaker as well as stronger players are more affected by piece removal than average players - with respect to their Elo. This is in part explained by

[2]Typically, one would avoid showing the same data multiple times, but these plots are already hard to interpret and also serve different purposes, so an exception seems reasonable.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 April 2022 doi:10.20944/preprints202204.0227.v1

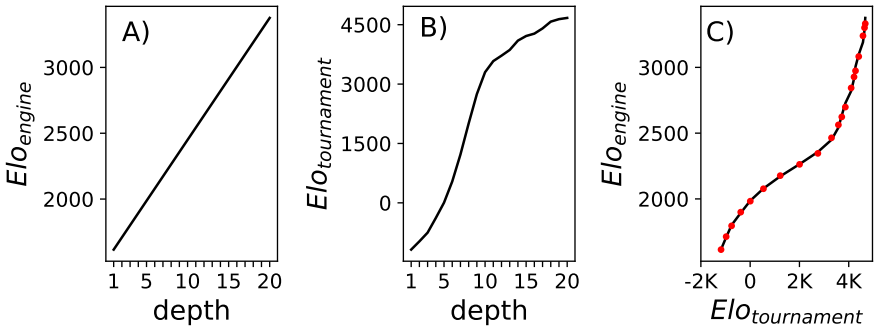

Fig 4

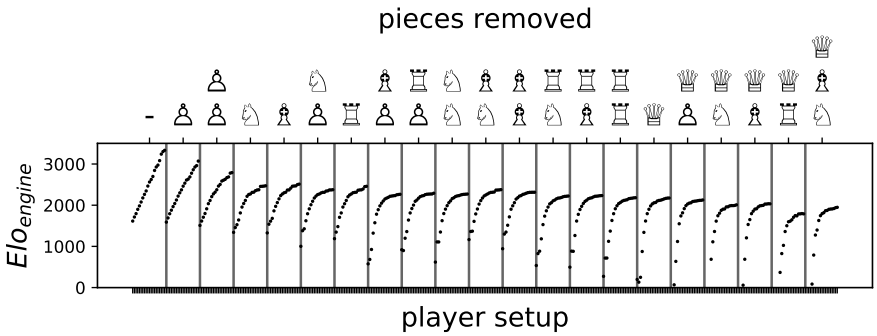

Fig 5. Corrected Elo_{engine} scores for all different player setups. Player setups on the x axis, ordered in sections for all the 21 possible piece removal combinations divided by gray lines.

the way Elos are computed, as it is a relative measure that approximates a normal distribution of Elos. To some degree, this overinflates high and low Elos compared to average ones. Thus, average players are not more or less affected than weaker or stronger players in an absolute or psychological sense, but only with respect to their Elo score. Still, engines playing with a small search depth are harder hit by piece removal than those playing with a deeper search depths. This is somewhat counter-intuitive. On the one hand, stronger players are probably better at dealing with the removal of a piece than weak players. Similarly, a strong player should be able to take advantage of a single piece much better than a weak player. However, in this tournament, as in real life, players of equal strength play against their peers (Elo not further apart than 400); yet, the effect a piece removal has on the Elo becomes smaller with increasing depth. This suggests that a weak player playing against other weak players can take advantage of an opponent's piece removed better than strong players playing against each other. This behavior might, in part, be also explained by players having different abilities to draw a game. Player setups with or without additional piece removals with high search depths draw more and against a broader range of other setups than those with a lower search depth (see Figure 8).

Lastly, for Figures 6 and 7 the order of pieces removed was arranged in such a way that Elo values for all player setups constantly dropped. As such, removing ♗♙has a more substantial effect than removing a ♖. None of the many ways to assign values to individual pieces suggests this [10].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 April 2022 doi:10.20944/preprints202204.0227.v1

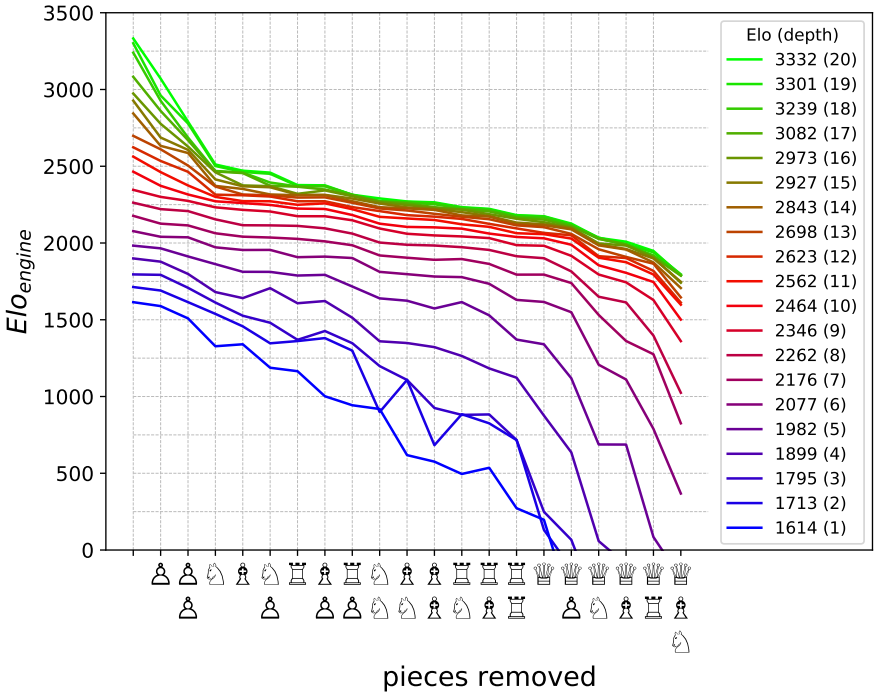

Fig 6. The effect of piece removal on Elo for differently strong players. The y-axis shows the predicted Elo playing strength given the removed pieces x-axis (staring with no piece removed). Each line represents a specific search depth and follows this depth (or estimated Elo) over all removed pieces tested here (see legend for colors). Pieces removed were ordered on the x-axis to avoid overlapping results.

Discussion and Conclusion

Playing the chess engine Stockfish 11.0 with different search depths in combination with different removed pieces gives a good insight into the probability of winning or losing games. From this, the Elo for each of these player setups can be determined. Still, a lot of fitting and mathematical transformations were necessary, potentialy adding errors or biases. Based on these results, the loss in Elo caused by removing pieces can be predicted. Interestingly, the effect each removed piece has on Elo strongly depends on the Elo of the player experiencing the loss. Figures 6 and 7 highlight these results and should allow players with different Elos to determine which pieces should be removed in order to have a more equitable outcome.

However, a significant degree of uncertainty has to be considered. No chess engine plays like a human, making a direct comparison difficult, even though an Elo to a chess engine can be assigned. Stockfish, even though one of the best chess engines, might play distinctly different from humans, and other using other chess engines might have different results. One surprising result is the advantage that ♗♙have over ♖, which is not explained by the values of the individual chess pieces. One future way to explore this matter is to match chess engines with different values for pieces and explore which values are advantageous. In particular, because AlphaZero, the currently best algorithm, trained by reinforcement learning also arrived at piece values that would rank a ♖higher than a ♗♙ [18]. Further, the Elo determined by the tournament used here is likely to incorporate a none neglectable error, and thus results should be considered a well-informed hypothesis and not the final answer. The hope is that using this method,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 April 2022 doi:10.20944/preprints202204.0227.v1

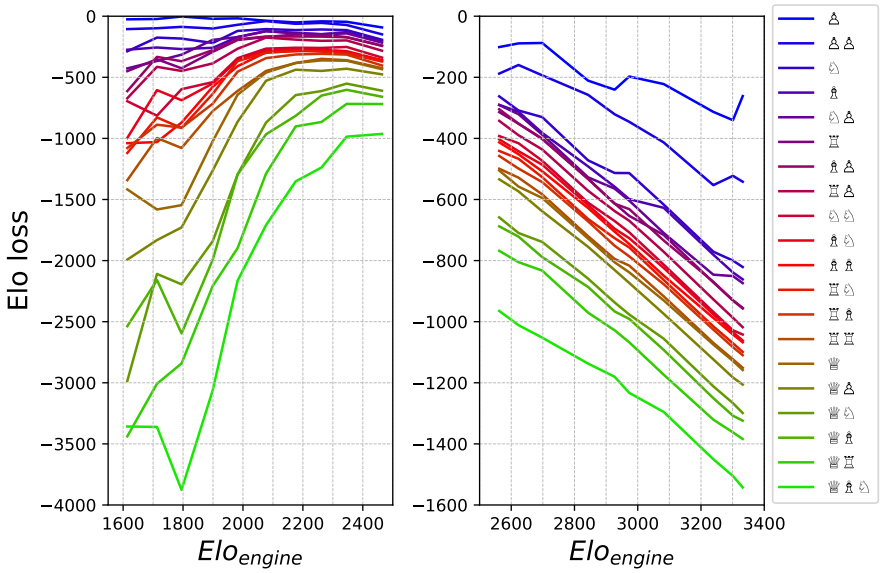

Fig 7. Drop in Elo given the strength of a player (x-axis) for different remove pieces (color code in the legend). The plot is split to allow for a better resolution of the y-axis for players with an Elo above 2500.

more "odd games" might be played in the future. Data collected from these games should allow for a better estimate of the relation between piece removal and the effect on Elo.

Acknowledgments

The computations were performed on resources provided by SNIC through Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX) under Project SNIC 2020-15-48 and by the Institute for Cyber-Enabled Research at Michigan State University supported by the BEACON Center for the Study of Evolution in Action at Michigan State University. I thank the chess.com team for their support, as

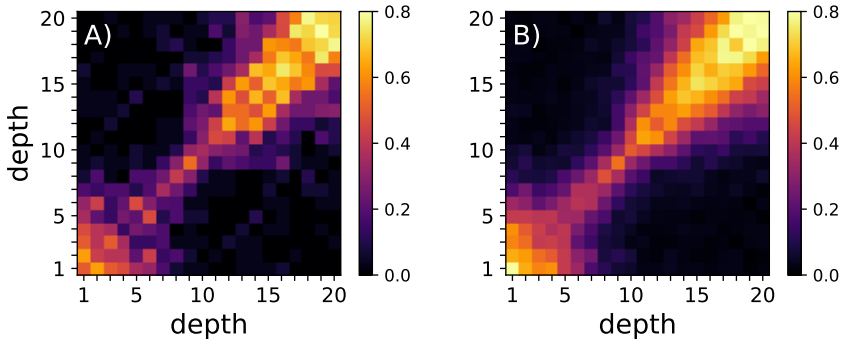

Fig 8. Fraction of games drawn between different player setups without additional pieces removed in panel A). Panel B) shows the same but now as an average including all possible pieces removed. Panels are symmetric across the diagonal, since the differences between playing white and black are neglectable (data not shown).

 doi:10.20944/preprints202204.0227.v1

well as the Stockfish developer community, especially Joost VandeVondele (vondele), for extremely valuable insights. Also, many thanks to GM Nils Axel Grandelius for a great discussion on this matter!

 doi:10.20944/preprints202204.0227.v1

References

1. Campbell M, Hoane Jr AJ, Hsu Fh. Deep blue. Artificial intelligence. 2002;134(1-2):57–83.
2. Hassabis D. Artificial intelligence: chess match of the century. Nature. 2017;544(7651):413–414.
3. Loges M, Aniceto M. The Queen's Gambit - TV Mini Series. Netflix, Flitcraft Ltd Wonderful Films; 2020.
4. Bilen E, Matros A. The Queen's Gambit: Explaining the Superstar Effect Using Evidence from Chess. Available at SSRN 3799396. 2021;.
5. Lowie FL. How big is the Netflix effect? The Queen's Gambit: The influence of Beth on the increase in chess numbers. Master's thesis - Utrecht University. 2021;.
6. Fuentes-García JP, Martínez Patiño MJ, Villafaina S, Clemente-Suárez VJ. The effect of COVID-19 confinement in behavioral, psychological, and training patterns of chess players. Frontiers in psychology. 2020;11:1812.
7. Elo AE. The rating of chessplayers, past and present. BT Batsford Limited; 1978.
8. Kupper LL, Hearne LB, Martin SL, Griffin JM. Is The USGA Golf Handicap System Equitable? Chance. 2001;14(1):30–35.
9. Shannon CE. Programming a computer for playing chess. In: first presented at the National IRE Convention, March 9, 1949, and also in Claude Elwood Shannon Collected Papers. IEEE Press; 1993. p. 637–656.
10. Wikipedia. Chess piece relative value — Wikipedia, The Free Encyclopedia; 2022. `http://en.wikipedia.org/w/index.php?title=Chess%20piece%20relative%20value&oldid=1080904823`.
11. Vázquez-Fernández E, Coello CAC, Troncoso FDS. Assessing the positional values of chess pieces by tuning neural networks' weights with an evolutionary algorithm. In: World Automation Congress 2012. IEEE; 2012. p. 1–6.
12. Pachman L, Palacio CA. Estrategia moderna en ajedrez. Martínez Roca; 1972.
13. Guid M, Možina M, Krivec J, Sadikov A, Bratko I. Learning positional features for annotating chess games: A case study. In: International Conference on Computers and Games. Springer; 2008. p. 192–204.
14. Matanovic A. Encyclopedia of Chess Openings (includes Korchnoi's analysis of the Open Defense to the Ruy Lopez). London: Batsford. 1974;.
15. Reem Ev. The Birth of Fischer Random Chess; May 31, 2001. Available from: `https://www.chessvariants.com/diffsetup.dir/fischerh.html`.
16. VandeVondele J. UCI ELO implementation by Vondele · pull request 2225 · Official-Stockfish/stockfish; 2019. Available from: `https://github.com/official-stockfish/Stockfish/pull/2225`.
17. CCRL team Ct. Computer Chess Rating Lists; 2005-2022. Available from: `https://ccrl.chessdom.com/ccrl/4040/rating_list_all.html`.
18. Tomašev N, Paquet U, Hassabis D, Kramnik V. Assessing game balance with alphazero: Exploring alternative rule sets in chess. arXiv preprint arXiv:200904374. 2020;.