

A Proposed Heuristic for a Computer Chess Program

(c)2008 John L. Jerz

Software Engineer, Lockheed Martin

Abstract. How might we create an evaluation function for a computer chess program that plays a stronger positional game of chess? A new heuristic for estimating the positional pressure produced by chess pieces is proposed. We calculate and maintain a database of potential mobility for each chess piece 3 moves into the future, for each position we evaluate in our search tree. We update this piece mobility database dynamically as we evaluate each new chess position (this database allows us to reward chess pieces for specific objectives they can accomplish in the future). We determine the restrictions placed on the future mobility of the pieces based on the mobility of the lower-valued enemy pieces. The central idea is that a better level of insight is obtained by this method. Initial results are presented.

1 Introduction

This paper is concerned with heuristic algorithms. According to Koen [9] a heuristic is anything that provides a plausible aid or direction in the solution of a problem but is in the final analysis unjustified, incapable of justification, and potentially fallible. Heuristics help solve unsolvable problems or reduce the time needed to find a satisfactory solution.

A new heuristic is proposed which offers better insight on the positional placement of the pieces to a chess-playing computer program. The heuristic will have usefulness in the evaluation function of a computer program, or as part of a teaching tool which explains to a human user the reasons that one side or the other has an advantage in a chess game.

The heuristic involves constructing a table of the future mobility for each piece, taking into account the other pieces on the board, as well as the likely *constraints* that these other pieces place on this future movement. The heuristic concept is described, and then examples are presented from a software application constructed to demonstrate this concept.

Computer chess programs have historically been weak in understanding concepts relating to positional issues. The proposed heuristic offers a method to potentially play a stronger positional game of chess.

2 Principles of Positional Chess

Understanding the principles of positional chess is a necessary starting point before designing concepts useful for a machine implementation. We select the relevant concepts of positional chess which have been addressed by multiple authors.

Stein [17] declares that the most important single feature of a chess position is the activity of the pieces and that the primary constraint on a piece's activity is the pawn structure. Stein defines a weak pawn as one which cannot be protected by another pawn, therefore requiring support from its own pieces. This is the ability to be protected by another pawn, not necessarily the existence of such protection. Stein declares that the pawn structure has a certain capacity for efficiently accomodating pieces and that exceeding that capacity hurts their ability to work together.

Aggaard [1] declares that all positional chess is related to the existence of weakness in either your or the opponent's position. This weakness becomes real when it is possible for the weakness to be attacked. The pieces on the board and their interactions define how attackable these weaknesses are.

Emms [6] declares that is an advantage if a piece is performing several important functions at once, while a disadvantage if a piece is not participating effectively in the game. Emms teaches that doubled pawns can be weak if they are attackable or if they otherwise reduce the mobility of the pawns. Doubled pawns can control vital squares, which might also mean denying mobility to enemy pieces. Isolated pawns require the presence of pieces to defend them if attacked.

Dvoretsky and Yusupov [5] argue that creating multiple threats is a good starting point for forming a plan. Improving the performance of the weakest piece is proposed as a good way to improve your position as a whole.

McDonald [11] gives an example of *good* doubled pawns which operate to restrict the mobility of the opponent's pieces and are not easily attackable. His view is that every position needs to be evaluated according to the unique features present.

Albus and Meystel [3] have written that the key to building practical intelligent systems lies in our ability to focus attention on what is important and to ignore what is not. Julio Kaplan in [8] says that it is important to focus attention on the few moves that are relevant and to spend little time on the rest.

The positional style is distinguished by positional goals and an evaluation which rewards pieces for their future potential to accomplish objectives. Katsenelinboigen declares [18]

the positional player is occupied, first and foremost, with the elaboration of the position that will allow him to develop in the unknown future. In playing the positional style, the player must evaluate relational and material parameters as independent variables. ...The positional style gives the player the opportunity to develop a position until it becomes pregnant with a combination. ...The positional style merely prepares the transformation to a combination when the latter becomes feasible.

3 The Positional Evaluation Function

We construct an evaluation function with the goal of making our machine more knowledgeable with regard to the positional concepts discussed above.

In order to more accurately estimate the distant positional pressure produced by the chess pieces, as well as to predict the future capability of the pieces in a basic form of planning [10] [15] we create the software equivalent of a *diagnostic probe* which performs a heuristic estimate of the ability of each piece to accomplish positional objectives. These objectives (as a starting point) will be attacking enemy pieces and supporting friendly pieces (especially those pieces that are weak). We calculate and maintain this database of potential mobility for each chess piece 3 moves into the future, for each position we evaluate.

We update this piece mobility database dynamically as we evaluate each new leaf position in our search tree. This database lets us reward chess pieces for the specific positional objectives they can attack or support in the future (such as 2 moves away from defending a piece or 3 moves away from attacking a square next to the enemy king). Note that the piece *mobility* we calculate is not itself the desired objective - it is the *means* through which we determine the pressure the piece can exert on a distant objective.

We reduce our bonus for each move that it takes the piece to accomplish the desired objective. We then consider *restrictions* which are *likely* to constrain the piece as it attempts to make moves on the board.

For example, let's consider the pieces in the starting position (Figure 1). What squares can our knight on g1 influence in 3 moves, and which squares from this set are likely off-limits due to potential constraints from the enemy pieces?

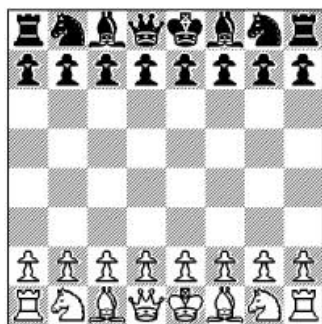


Fig. 1. The chess pieces at the starting position

We now introduce the concept of the *influence diagram* (from [15]) and the *engagement diagram* (Figure 2, a new concept), which are interpreted in the following way. If a piece is on our influence diagram for the knight, then it is possible to attack it or defend it in 3 moves (this includes waiting moves or

			3		3		3
		3		3		3	
	3		3	2	3	2	3
		3	2	3	2	3	2
	3	3	3		1	3	1
		3	2	1	2	3	2
		3		2		N	

			X		X		X
		X		X		X	
	3		3	2	3	2	3
		3	2	3	2	3	2
	3	3	3		1	3	1
		3	2	1	2	3	2
		3		2		N	

Fig. 2. Influence Diagram and Engagement Diagram, Ng1 at starting position

moves which move a piece out of the way). We label this kind of map an *influence diagram* because it accurately shows the squares that the piece can influence in 3 moves, provided that it is unconstrained in movement by the enemy.

Keep in mind that we need to take into account the location of the other pieces on the chessboard when we generate these diagrams for each piece. If we trace mobility through a friendly piece, we must consider whether or not we can move this piece out of the way before we can continue to trace mobility in that particular direction. If we trace mobility through an enemy piece, we must first be able to spend 1 move capturing that piece.

Comparing this 3-move map with a diagram of the starting position, we can determine that the white knight on g1 can potentially attack 3 enemy pieces in 3 moves (black pawns on d7, f7 and h7). We can defend 8 of our own pieces in 3 moves (the knight cannot defend itself).

We decide to reward pieces for their ability to accomplish certain types of worthwhile positional objectives: attacking enemy pieces, defending friendly pieces, attacking squares near our opponents king (especially involving collaboration), minimizing our opponents ability to attack squares near our own king, attacking pieces that are not defended or pawns that cannot be defended by neighboring pawns, restricting the mobility of enemy pieces (specifically, their ability to accomplish objectives), etc. In this way, we are getting real about what the piece can do. The bonus we give the piece is 1. a more precise estimate of the piece's ability to accomplish positional objectives that have value and 2. operationally based on real things present on the chessboard. In this way, our positional evaluation function will obtain insight not usually obtained by a computer chess program. It is still an estimate, but the goal here is to focus our search efforts on likely moves in a positional style of play, and to evaluate positions from a more positional point of view.

What does the evaluation function look like for the proposed heuristic?

We model (and therefore estimate) the positional pressure of our pieces, by following a two-step process:

1. We determine the *unrestricted future mobility* of each chess piece 3 moves into the future, then

2. We estimate the *operating range* or level of *engagement* of the pieces by determining the limiting factors or constraints that bound the unrestricted mobility. The concept of using limiting factors is briefly mentioned by Blanchard and Fabrycky in the context of Systems Engineering [4]. The consideration of constraints is a part of the *decision protocol of Orasanu and Connolly* [13] and [14] which also includes the identification of resources and goals facing the decision maker. We therefore reduce the bonus for accomplishing objectives (such as, attacking an enemy piece or defending a friendly piece) if the required moves can only be traced through squares that are *likely* to result in the piece being captured before it can accomplish its objective.

We also reduce the engagement bonus for mobility traced through squares where the piece is attacked but not defended.

We may use another scheme (such as probability) for determining mobility reduction for piece movement through squares attacked both by friendly and enemy pieces where we cannot easily resolve whether or not a piece can trace mobility through the square in question.

We reward each piece for its predicted ability to accomplish strategic objectives, exert positional pressure, and restrict the mobility of enemy pieces, based on the current set of pieces on the chess board at the time we are calling our evaluation function. van Wezel, Jorna and Meystel [19] say that anticipation (as a strategy) can be costly and is limited by time constraints. It can hurt our performance if it is not done with competence. An efficient compromise between anticipative and reactive strategies would seem to maximize performance.

We give a piece an offensive score based on the number and type of enemy pieces we can attack in 3 moves. We give a piece a defensive score based on how many of our own pieces it can move to defend in 3 moves. Again, this bonus is reduced for each move it takes to potentially defend such piece. This information is derived from the influence diagram and engagement diagram we just calculated. Extra points can be given for weak or undefended pieces that we can threaten.

The proposed heuristic determines also king safety from these future mobility move maps. We penalize our king if our opponent can move pieces into the 9-square template around our king within a 3 move window. The penalty is larger if the piece can make it there in 1 or 2 moves, or if the piece is a queen or rook. We penalize our king if multiple enemy pieces can attack the same square near our king. Our king is free to move to the center of the board - as long as the enemy cannot mount an attack. The incentive to castle our king will not be a fixed value, such as a quarter pawn for castling, but rather the reduction obtained in the enemy's ability to move pieces near our king (the rook involved in the castling maneuver will likely see increased mobility after castling is performed) The king will come out of hiding naturally when the number of pieces on the board is reduced and the enemy does not have the potential to move these reduced number of pieces near our king. We are likewise free to advance the

pawns protecting our king, again as long as the enemy cannot mount an attack on the monarch. The potential ability of our opponent to mount an attack on our king is the heuristic we use as the basis for king safety. Optionally, we will consider realistic restrictions that our own pieces can make to our opponent's ability to move pieces near our king, as discussed in point 6 below.

Pawns are rewarded based on their chance to reach the last rank, and what they can do (pieces attacked and defended in 3 moves, whether or not they are blocked or moveable). The piece mobility tables we generate should help us identify pawns that cannot be defended by other pawns, or other pieces - it is this weakness that we should penalize. Doubled or isolated pawns that cannot be potentially attacked blockaded or constrained by our opponent should not be penalized. Pawns can be awarded a bonus based on the future mobility and offensive/ defensive potential of a queen that would result if it made it to the back rank, and of course this bonus is reduced by each move it would take the pawn to get there.

The information present in the future mobility maps (and the constraints that exist on the board for the movement of these pieces) allow us to better estimate the positional pressure produced by the chess pieces. From these calculations we can make a reasonably accurate estimate of the winning potential of a position, or estimate the presence of positional compensation from a piece sacrifice. This evaluation score also helps steer the search process, as the positional score is also a measure of how interesting the position is and helps us determine the positions we would like to search first.

In summary, we have created a model of positional pressure which can be used in the evaluation function of a computer chess program. Michalewicz and Fogel [12] have this to say about using models to solve problems:

Whenever you solve a real-world problem, you have to create a model of that problem first. It is critical to make the distinction that the model that you work with isn't the same as the problem. Every model leaves something out. It has to - otherwise it would be as complicated and unwieldy as the real-world itself. We always work with simplifications of how things really are. We have to accept that. Every solution we create is, to be precise, a solution only to the model that we postulate as being a useful representation of some real-world setting that we want to capture.

Starfield, Smith and Bleloch [16] emphasize that problem solving and thinking revolve around the model we have created of the process under study. We can use the proposed model of positional pressure to direct the machine to focus the search efforts on moves with well-placed pieces. For our search efforts, we desire a proper balance between an anticipatory and a reactive planning strategy. Using a suggestion of van Wezel, Jorna, Meystel [19] we desire our forecast of each piece's abilities to help us anticipate its effectiveness in the game, instead of just reacting to the consequences of the moves.

4 Results

We have created software to implement the proposed heuristic and now examine several positions to see if we can obtain a better positional understanding of how well the pieces are likely performing.

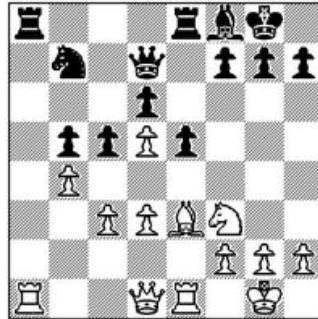


Fig. 3. Emms-Miralles (1998), Black to move

John Emms, in his book *Simple Chess*, reached Figure 3 as white with the idea of restricting the mobility of black’s knight on b7. How fully engaged is this piece in the game? Let’s see what the *influence diagram* and *engagement diagram* from the proposed heuristic show us:

R							
R							
R		P		P			M
P		P		M		M	
R	P	P	P	P	M		M
R	Q		Q	P	P	P	P
R		Q	M	R	M	K	M
Q	R	M	R	M	R	M	K

r	r	r	m	r	r		k
r	q	q		m	q	m	k
r		q	m	p	p	p	p
m	q	p		p	q		
p	p	p	p		p	q	
r							q
r							
r							

Fig. 4. Emms-Miralles Constraint maps

We generate the *constraint maps* as in Figure 4 in order to estimate the squares that the knight on b7 is likely to be denied access.¹ We then apply the

¹ The simplified version of a *constraint map* assumes that bishop and knight are of equal value, and we use *M* or *m* to represent a white or black *minor* piece.

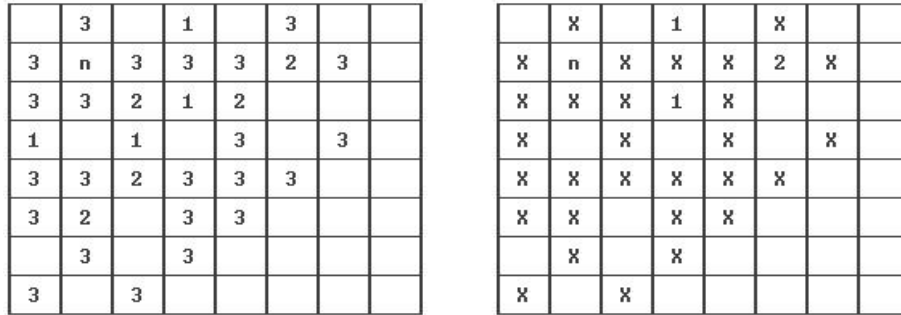


Fig. 5. Influence Diagram and Engagement Diagram for Nb7

constraint maps to the influence diagram to create the engagement diagram. We can see from Figure 5 that the movement of the piece on b7 has been constrained.

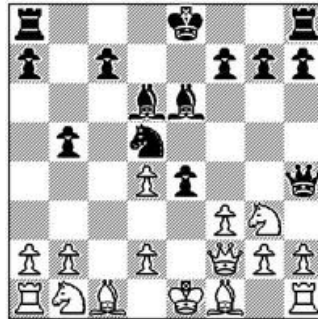


Fig. 6. Estrin-Berliner analysis (1965-68 corr.) after 12.Qe2 Be6 13.Qf2, Black to move

Figure 6 examines a sideline from Estrin-Berliner (1965-68 corr.) after 12.Qe2 Be6 13.Qf2. How fully engaged is the white Bishop on c1? We generate constraint maps and influence diagram as before in order to generate the engagement diagram. We see that the bishop on c1 can enter the game after moving a pawn out of the way.

The computer can use the knowledge present in the influence diagram and engagement diagram to estimate how *fully engaged* each piece is in the game. The maps are useful in 1) steering our search efforts 2) deciding which positions are the most promising from a positional point of view and 3) predicting the winning potential at the end points of our search tree

Figure 8 is from Nickel-Hydra (2004 corr.) after 18.g4 was played.

The black knight on h6 has been *constrained* to stay on the edge of the board by the white pawns and the location of the other pieces. It is unlikely that Hydra was aware of the long term consequences of the constraints placed

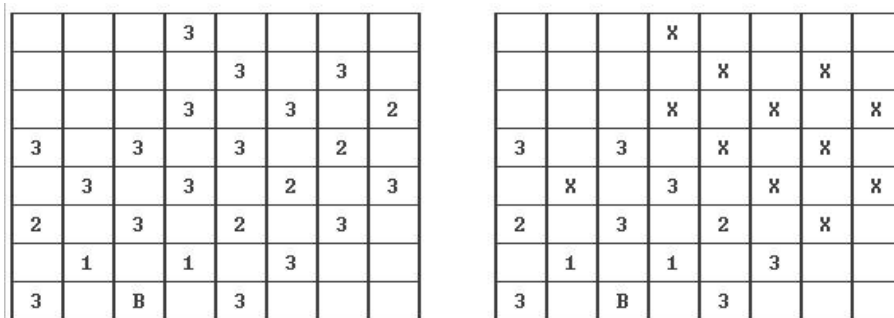


Fig. 7. Influence Diagram and Engagement Diagram for Bc1

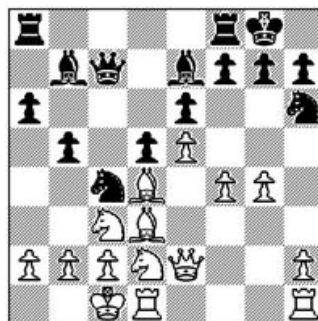


Fig. 8. Nickel-Hydra (2004 corr.) Black to move

on the Black knight on h6. In fact, the knight on h6 never moved the remainder of the game, which white won. The influence diagram and engagement diagram corresponding to the knight on h6 (Figure 9 strongly suggest that this piece has been effectively constrained by correspondence GM Nickel and therefore made ineffective in the game.

Figure 10 is from Fischer-Petrosian (1971). Fischer was criticized for playing Nxd7 and giving up a well-placed knight for a Bishop which seems to be *bad*. We can use the proposed heuristic to examine the positional placement of the d7 Bishop. The influence diagram and engagement diagram of Figure 11 indicate that the d7 Bishop has influence over much of the board, although it has been constrained generally to a defensive role on the queenside. Perhaps Fischer used a similar insight before deciding to exchange the piece.

5 Conclusions

The proposed heuristic offers insight on the ability of the chess pieces to accomplish strategic objectives. It offers promise as a component of an evaluation function for a computer chess program. It also offers promise as a component of

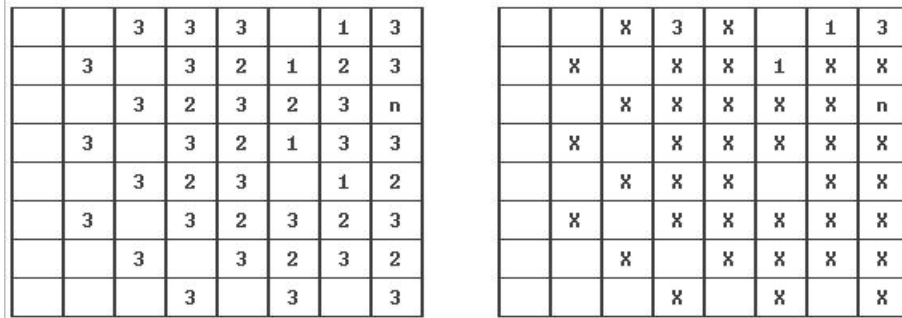


Fig. 9. Influence Diagram and Engagement Diagram for Nh6

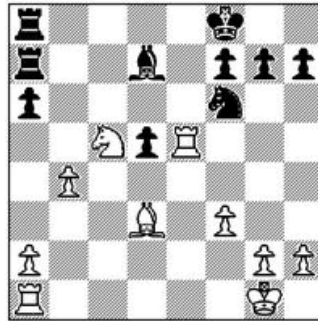


Fig. 10. Fischer-Petrosian (1971), white to move

a chess tutor, as it can offer insight into the ability of the pieces to accomplish positional objectives. The presented results demonstrate this insight for several test positions. Perhaps chess is more than just calculation [2], but the day may come sooner than we think when computers use heuristics to play a positional game of chess at skill levels equal to their current strong tactical play.

Future work will involve construction of a full positional evaluation function as part of a computer chess program.

References

1. Aagaard, J.: Excelling at Positional Chess, pp.35-36. Gloucester Publishers plc, London (2003)
2. Aagaard, J.: Inside the Chess Mind, pp.157. Gloucester Publishers plc, London (2004)
3. Albus, J., Meystel, A.: Engineering of Mind, pp.80. John Wiley and Sons, New York (2001)
4. Blanchard, B., Fabrycky, W.: Systems Engineering and Analysis, Fourth Edition, pp.162-163. Prentice Hall, Upper Saddle River, New Jersey (2006)

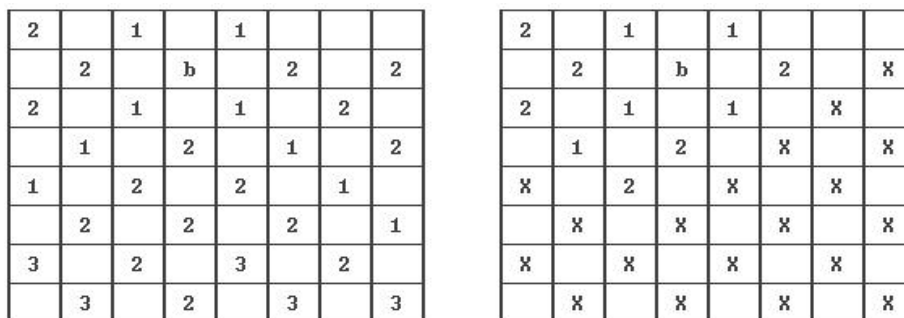


Fig. 11. Influence Diagram and Engagement Diagram for Bd7

5. Dvoretzky, M., Yusupov, A.: Positional Play, pp.111,121. Henry Holt and Company, New York (1996)
6. Emms, J.: Simple Chess, pp.7,63,67,70. Everyman Publishers plc, London (2001)
7. Kahneman, D., Slovic, P., Tversky, A.: Judgment under Uncertainty: Heuristics and Biases pp.69. Cambridge University Press, New York (1982)
8. Kaplan, J.: The ABCs of Chess. In: Hochberg, B., Chess Life and Review pp.370, New Windsor, NY July (1978)
9. Koen, B.V.: Discussion of the Method, pp.28. Oxford University Press, New York (2003)
10. Lakein, A.: How to Get Control of Your Time and Your Life, pp.25. Signet, New York (1974)
11. McDonald, N.: The Art of Planning in Chess, pp. 35,86. Batsford, London (2006)
12. Michalewicz, Z., Fogel, D.: How to Solve It: Modern Heuristics, pp.31. Springer, New York (2004)
13. Orasanu, J., and Connolly, T.: The Reinvention of Decision Making. In: Klein, G.A., Orasanu, J., Calderwood, R. and Zsombok, C.E. (eds.) Decision Making in Action: Models and Methods, pp. 3-20, Ablex, Norwood, New Jersey (1993)
14. Plessner, H., Betsch, C., Betsch, T.: Intuition in Judgment and Decision Making, pp.120-121. Lawrence Erlbaum Associates, New York (2008)
15. Shoemaker, P.: Forecasting and Scenario Planning: The Challenges of Uncertainty and Complexity. In: Koehler, D., Harvey, N. (eds.) Blackwell Handbook of Judgment and Decision Making, pp.281,283, Blackwell Publishing, Malden MA (2007)
16. Starfield, A., Smith, K., Bleloch, A.: How to Model It: Problem Solving for the Computer Age, pp.21. Burgess International Group, New York (1994)
17. Stean, M.: Simple Chess, pp.7,39,42,132. Dover Publications, Mineola, New York (2002)
18. Ulea, V.: The Concept of Dramatic Genre and The Comedy of A New Type, pp.17-18. Chess, Literature, and Film. Southern Illinois University Press, (2002)
19. van Wezel, W., Jorna, R., Meystel, A: Planning in Intelligent Systems, pp.23,65,96. John Wiley and Sons, Hoboken, New Jersey, (2006)