Building Chess Endgame Databases for Positions with many Pieces using A-priori Information

Eiko Bleicher*
Lehrstuhl Mathematische Optimierung
Fakultät Mathematik und Informatik
Friedrich-Schiller-Universität Jena
07740 Jena, Germany

August 26, 2004

Abstract

This note describes a possible solution to some endgames in chess with many pieces, when a complete retrograde database construction will fail because of the size of the problem and the time needed to build it up. The note shows how human knowledge, position-specific limitations and existing databases can be combined to prove a position to be won, drawn or lost, and providing the game theoretical optimal moves. The approach presented is fast enough to give results within minutes to hours when a normal PC is used. The computations with our program "Endgame Freezer" on the sample position Shirov vs. Fishbein in Section 3 took about eight minutes to complete, for example.

1 Introduction

When analyzing endgames in chess one is often in the unfortunate position that there are too many pieces left to look it up in currently available endgame databases. On the other hand the position examined may be too complex to be solved by using ordinary chess playing software. The main idea of the concept presented is to perform a retrograde analysis, but reduce the index space for the resulting database by orders of magnitudes. This reduction leads not only to space savings but to big time savings, making this approach usable as a real time analyzing tool. Former work on the subject of retrograde analysis has either tried to solve a certain material balance in total or at least with hard coded restrictions, i. e. van der Herik, Herschberg, and Naka constucted the database KRP(a2)–KbBP(a3) in 1987 [HHN87] by restricting the pawns to one square each and considering the bishop on black squares only. Later, Nalimov, Wirth, and Haworth analyzed the material constellation KQQ–KQQ which was of interest in the Kasparov – World game [NWH99]. Retrograde analysis dates back to 1970 (Ströhlein) [Str70], and van den Herik, and Herschberg gave an detailed introduction in 1985 [HH85]. The basic retrograde algorithm we use is described by Wu and Beal (2001) [WB01]. The new aspect presented here is to include assumptions on the outcome of the game in some positions if certain conditions hold.

The basic idea of retrograde analysis is to start with the final positions, i. e. checkmate positions. The predecessors of lose positions are wins for the other side,

*Email: bleicher@k4it.de

and the predecessors of win positions are candidates for lose positions etc. The presented approach adds a special behaviour when looking at a certain position, as it is first checked if there is a certain rule defined that evaluates it as won, drawn or lost. That way it isn't necessary to include these positions into the database.

Our current prototype implementation (called "Endgame Freezer" or "Freezer" for short) supports the use of the endgame tablebases published by Eugene Nalimov [NalTB]. The databases generated can be browsed and the browsed moves can be exported in pgn standard. Our implementation allows three types of restriction.

This note is organized as follows. Section 2 describes the three types of restrictions we implemented, followed by some examples in Section 3. The note closes with a conclusion, acknowledgments and some references.

2 Supported Rules

2.1 Restricting Placements of Pieces

The key idea is to restrict pieces to a small set of squares that are relevant for the given position. As the problem space is roughly the product of the number of squares each piece can be placed on, cutting down these numbers yields a major saving in both space and time. There are some ways of using this approach without the need of introducing any knowledge, i. e. one can use the property of the bishop that it is stuck to either white or black squares. Further, a pawn cannot move backwards and can change its file only in case of a capture. These cases are quite obvious, and do not need any more details here.

However there are many more ways to limit the number of squares for pieces. In many cases one wants to check if a certain position is won for one side. If so, the position might also be a win if the pieces of the winning side do not move to certain parts of the board, for example the king might only use central squares, or a knight should not make manoeuvres near the borders. When analyzing for a draw, the following situation may arise: Does the opponent have a win, if I only make moves in this area of the board? A typical situation of this is a fortress in KQP-KRP. The limitation of squares for pieces allows checking if there are some special manoeuvres that win or draw. On the other hand, if the limitation is too strong, one might lose a winning move when looking for a win, or a drawing move when looking for a draw. So this approach gives "only" a lower bound for the possible achievement of the party restricted. Although it is possible to apply restrictions to both sides, one might get inaccurate results when limiting the weaker side and possibly cutting away drawing or winning possibilities.

For each piece/square combination forbidden one has to specify how the arising position should be treated, i. e. win, draw or loss. These positions cannot be left out totally unseen, as they can be legal successors of positions of interest that must be evaluated for the retrograde analysis. If a piece shall not go to a certain square, this combination should be defined as a loss.

2.2 Captures

When a capture of a piece occurs this normally implies the use of the corresponding database with one piece less. If these databases exist they can be used. However, in many cases these files do not exist, or are not available from the analyzer's hard disk. Often only a few capturing cases are of interest, and these cases can be handled manually. In most endgame positions examined a loss of a piece directly implies a loss if there is no recapture. The approach presented allows the definition of rules that evaluate these positions as win, draw, or loss. The rules can be restricted to the type of piece and the square of capture. A possible recapture can be set to give a different result, and a second recapture can get another special treatment. Of course, when applying these rules the possibility for a stalemate can be tested.

2.3 Pawn Promotions

As the promotion of a pawn changes the material balance, existing algorithms can only generate databases with pawns if all of the corresponding databases with pieces instead of pawns are available. However, like the capture of a piece, a pawn promotion often plays a decisive role. Rules can be defined to apply to certain promotion squares, and can handle special treatments if the promoted piece can be captured.

3 Examples

3.1 Defending a Difficult Position

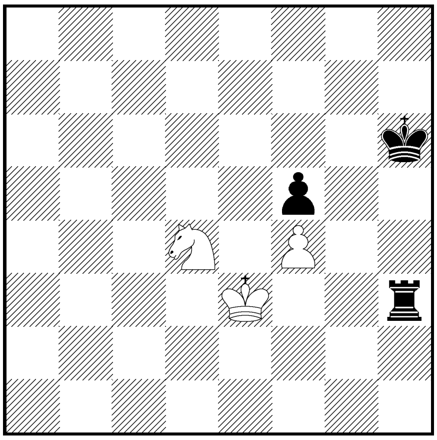

Figure 3.1: *A. Shirov – A. Fishbein, Kerteminde 1991*

This position can be proven in a totally safe way to be a draw . At the time of writing this note (August 26, 2004), the according six men database for this position hasn't yet been published by Nalimov or others. As the pawns are blocking each other, there is no way to promote a pawn without a capture. All sub databases needed are true five men tables which are available. There are three rules that break this endgame down to a problem with four pieces.

1. In case of a capture, look up the correct result in the corresponding Nalimov tablebase.
2. Stick the white pawn to f4 (loss for white if the pawn occupies another square, capture moves are handled by the first rule).
3. Stick the black pawn to f5 (loss for black if the pawn occupies another square, capture moves are handled by the first rule).

3.2 A Fortress

Fortresses seem to be a common thing in matches of humans against the computer, but they happen to arise on the board in games between humans, too. The main idea is that the defending party places its pieces in a manner that the attacker's pieces cannot make progress. A classic example for a fortress is shown below in figure 3.2.

Although this endgame is not difficult to evaluate with the appropriate five men tablebase at hand, the example shows how the right use of conditions can shrink the problem size down by a good amount. The following rules were used to prove the draw. Further refinements are possible, but already the conditions below reduce the index-space down to approximately 80 thousands entries.

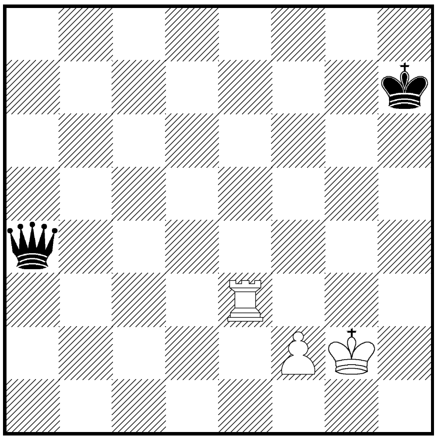

Figure 3.2: *A classic fortress*

1. In case of a capture, look up the correct result in the corresponding Nalimov 4-men tablebase.
2. Stick the white pawn to f2 (loss for white if the pawn occupies another square; captures are handled by the first rule).
3. Stick the white rook to e3 and g3 (loss for white if the rook occupies a square other then e3 or g3, capture moves are handled by the first rule).
4. Stick the white king to the 4x4 subboard with corners e1–e4–h4–h1 (in analogy to the third rule).
5. Stick the black king to the subboard with corners a4–a8–h8–h4 (win for black, if his king can get to a1–h3).

3.3 Mating with Bishop and Knight

When examining the material constellation king, bishop, and knight vs. king all books on endgames known to the author present the mating manoeuvre containing the famous W-rule, the route that the attacking knight should take. However, it was neither clear nor easy to test if the attacking side can win without moving the knight in that manner. With the approach presented there is an easy way to prove that this manoeuvre is not mandatory. As each "W" uses a central square (d4, d5, e4, or e5) only one rule is needed, that regards a position as lost for white if the knight occupies one of the central squares. White still can force mate. The longest optimal sequence with this restriction has 36 moves; without any restrictions the black king can be checkmated within at most 33 moves according to the Nalimov tablebases.

The following position shows an alternative way to mate the defending king without using the "W-manoeuvre".

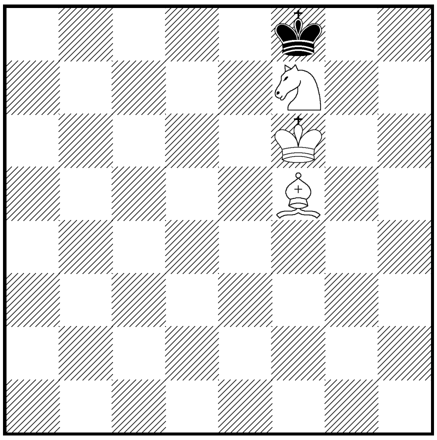

Figure 3.3: *Mating with bishop and knight without the "W"-rule*

1.♘h6 (instead of ♘e5 which would start a "W") ♔e8 2.♘g4 ♔d8 3.♘e3 ♔c7 4.♘c2 ♔d6 5.♗e6 ♔c5 6.♔e7 ♔c6 7.♗b3 ♔c5 8.♔d7 ♔b5 9.♔d6 ♔a5 10.♔c6 ♔a6 11.♘a3 ♔a5 12.♔c5 ♔a6 13.♘b5 ♔b7 14.♗e6 ♔a6 15.♘d6 ♔a5 16.♗d7 ♔a6 17.♗b5+ ♔a7 18.♔c6 ♔b8 19.♔b6 ♔a8 20.♔c7 ♔a7 21.♘c8+ ♔a8 22.♗c6#. Altogether the knight has performed the roundtrip f7–h6–g4–e3–c2–a3–b5–d6–c8.

4 Conclusion

This note shows that in many cases a-priori information can be exploited so that a common retrograde approach can be applied to larger problems that otherwise cannot be addressed. Of course several other rules can be added to the implementation. The crucial point is that there is no automatic validation of the human-defined rules, so constructing difficult rule sets can introduce logical mistakes. Chess grandmaster and endgame expert Karsten Müller has found the Freezer prototype to be useful not only to prove correct outcome assumptions, but also to give a good starting point for conventional human analysis.

Although some positions might need quite a good human insight to get the conditions right, there are a lot of positions that can be handled without any room for mistakes. These are positions for which all of the corresponding sub databases have been generated, for example positions with seven pieces, as quite a good amount of the six men tablebases are done. If in this 7-men-case pawns are left on the board, they should be blocked against each other, so that they cannot advance without a capture. On the other hand, long chains of blocked pawns can also be studied, i. e. whether the attacking king can win a pawn or not. As the pawns cannot move, they do not make the index space any bigger.

Although this method of restrictions has only been applied to Western chess so far, it should be useful for many other games like Chinese chess, checkers, and variants of nine men morris.

5 Acknowledgments

I want to thank Ingo Althöfer for advising my Ph. D. project and providing a wealth of interesting thoughts. Chess grandmaster and endgame expert Karsten Müller deserves thanks for testing the Freezer prototype and choosing interesting test positions. I also thank Eugene Nalimov, Andrew Kadatch, and Robert Hyatt

for generating and compressing tablebases up to 6 pieces and making them publicly available.

References

[HH85] Herik, H. J. van den, and Herschberg, I. S. (1985). *The Construction of an Omniscient Endgame Data Base*. ICCA Journal, Vol. 8, No. 2, pp 66–87.

[HHN87] Herik, H. J., Herschberg, I. S., and Naka, N. (1987). *A Six-Men-Endgame Database: KRP(a2)KbBP(a3)*. ICGA Journal, Vol. 10, No. 4, pp 163–180.

[Mül] Müller, K. *Endgame Corner*. Periodical column on endgames, every two weeks. `http://www.ChessCafe.com`.

[NalTB] Tablebases by E. Nalimov. `ftp://ftp.cic.uab.edu/pub/hyatt/TB/`.

[NWH99] Nalimov, E. V., Wirth, C. and Haworth, G. McC. (1999). *KQQKQQ and the Kasparov–World Game*. ICCA Journal, Vol. 22, No.4, pp 131–138.

[Str70] Ströhlein, T. (1970). *Untersuchungen über kombinatorische Spiele*. Dissertation, Fakultät für Allgemeine Wissenschaften der Technischen Hochschule München.

[WB01] Wu, R., and Beal, D. F. (2001). *Fast, Memory-efficient Retrograde Algorithms*. ICGA Journal, Vol. 24, No. 3, pp 147–159.