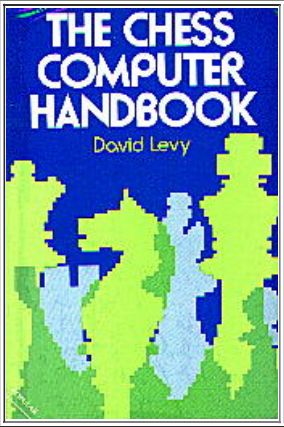
Chess Bibliography: The Chess Computer Handbook

Last update: 14/10/97

The Chess computer handbook



Title: The Chess computer handbook
Author: David Levy
Editorial: Batsford
ISBN: 0-7134-4220-4, 129 pages
Copyright: D.N.L. Levy, 1984

Abstract: This interesting little book gives a thorough introduction to the basics of a chess program, detailing with examples the fundamental techniques. Besides, it gives advice on how to play against a machine, and shows how it can help you to develop your skills. A chapter on what to look for in a chess computer proves most useful.

Review:

This little book packs between its covers one of the most clear, useful guides you could find to introduce you to the art of computer chess programming. In fact, it was the first book I read on the topic, a number of years ago, and its detailed, clear explanations, together with its many sample positions and diagrams made it a real pleasure to read, and also prompted me to write a simple, but perfectly workable chess program with the utmost ease.

The book itself is structured in a number of different parts: first, and introduction to the magic and mysteries of chess programming, then a selection of some of the best computer games, followed by a somewhat speculative chapter on how strong can chess programs become. The second part shows you how to play against a machine for maximum results. The author knows this subject first hand, as the famous *Levy bet* put all the pressure on him to play against a computer for blood. And win, he did. Finally, some very good advice on what to look for in a computer chess program or dedicated machine, which are still most useful even today.

After this brief, and somewhat enthusiastic introduction, a detailed review chaper by chapter:

The very first chapter, **Position representation and Move generation** begin with the most basic concepts: how do you represent a chess position in the memory of a computer, and how can you generate all legal moves in a position. Although apparently very basic, the topic is discussed at length, including such finesses as move generation by square offset, table driven move generation, and incremental move generation.

Once we have the position and the moves, we need some device to distinguish a good move among the rest. Enter Chapter 2, **Position Evaluation**, where the all-important evaluation function is introduced in theory, then actually created, step by step. We are told about what it should measure and how: material, mobility, development and castling, center control, king attack, king safety, pawn structure, piece attack and defence, and endgame features. All of them (did I mentioned it before ?) generously accompanied by practical sample positions, either specially composed or taken from real games (Fischer-Spassky, for instance), and diagrams, with all necessary numbers and calculations worked out in glorious detail. Final advice is given in how to adjust the weightings of the different components of the evaluation function.

Now we have the position, the moves, and some way to discriminate among them. But we have to search those move before we can evaluate them, and Chapter 3, **Tree searching** introduces the necessary techniques: mini-max, alpha-beta pruning, the killer heuristic, the alpha-beta windows, and iterative deepening. Here you'll finds diagrams and even line drawings to make all examples as clear and easily understandable as possible.

With all these preliminary steps understood and dominated, Chapter 4, **Search strategies** tells us about Shannon-A and B strategies, techniques to implement forwad pruning (the null move heuristic), and the extremely important concept of transposition tables and endgame databases. To be honest, all of them are treated with much less detail than the former concepts, mostly they are briefly introduced for you to know about their existence, but this time without examples or diagrams, just a ligth presentation.

Chapter 5, **Best computer games** features eight full-length games of quite high quality between programs and quite strong players (IMs and GMs), and between one program and another. All games are fully commented and analyzed.

Now, I've alway found Chapter 6, **How strong can computers become**, as one of the most interesting chapters of this book or of any other computer chess book for that matter. It develops in detail the puzzling concepts of zugzwang (the obligation to move proves

fatal), the horizon effect, the *never* concept, and the problem with extremely similar positions with totally different outcomes. In particular, the 'never' concept impressed me most, and certainly, even after 15 years have passed by since the publication of this book, most modern programs, if not all, fail miserably whenever the *never* concept appears in a position. See some of my Test positions that show off clearly this problem.

Chpater 7, **How to play against a chess program** is more relaxed, and tells you what you need to know to survive the opening, the middlegame and the endgame against a chess program. The author did survive long enough to win the *Levy bet*. though of course, neither him nor you or me would have many hopes of surviving against Deep Blue, no matter how good the advices. About the only way I can imagine to defeat a Deep Blue would be to develop on the board a position where the *never* concept applies, and the human does recognize its *presence* !

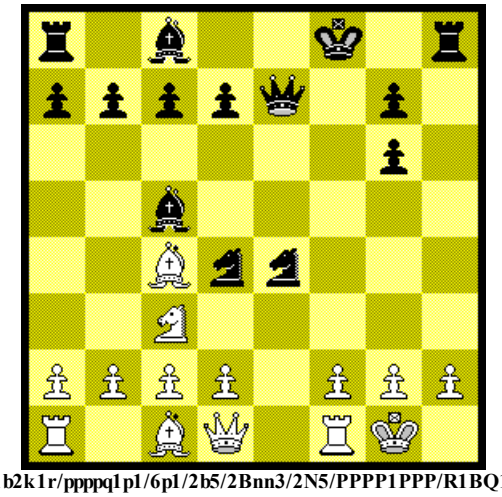
Chapter 8, **What to look for in a chess computer** is not as dated as you might expect. More or less advanced, all dedicated machines and programs still guide by the same basic principles, though I don't think you need to worry about a modern program or machine making or accepting illegal moves. But the rest, mostly applies even today.

The book ends after 129 glorious pages, with a Glossary and a Bibliography, as it should.

Here is a sample position from this book: quoted from pag. 71, **Best computer games**:

... Black is a piece for a pawn up, and could reasonably be expected to win, but with the black king stranded in an unsafe area of the board, in a game between two computer programs, the result is not quite so clear. However, there exists in this position the most beautiful combination ever played by a computer program ...

Diagram 40: Blitz 6.5 vs BELLE



FEN: r1b2k1r/ppppq1p1/6p1/2b5/2Bnn3/2N5/PPPP1PPP/R1BQ1RK1/ b

Black to play and win: 1. ... Rxh2 !!

Reviewer Notes:

When I first saw this position, I thought it was a mate in 5, and BELLE just happened to find it midgame, due to its hardware-assisted speed of calculation.

But things are not so simple, and this position *is not* a mate in 5. Which is more, it's also not so crystal clear that *1. ... Rxh2 !!* is the best move, however spectacular.

For instance, **Chess Genius 1.0**, running on a P100 and using a hash table of just 320 Kb, needs more than 4h 26m to search up to 10 full plies + 12 extended plies, to find that *d7-d5* is the best move, with a value of +7.78. This can be interpreted this way: the program is finding *delaying tactics* (absurd piece sacrifices) to avoid a mate in 5. As there are delaying sacrifices, actually *it's not a mate in 5*.

Chess Genius only considers that *Rh8xh2* is best when looking at 3/15 plies, evaluating it at +3.03. After that, all plies from 4/16 to 10/22 select *d7-d5* with values ranging from +3.83 to +7.78.

Crafty 12.7, running on the same hardware, but with a 12 Mb transpositions hash table and a 5 Mb pawn structures hash table, searches 12 full plies in 4h 17m, and it also thinks that *d7-d5* is the best move, with a value of +6.812.

NEW Addendum:

Kai Luebke tried this position on some of the chess programs he has at hand, and sent me these results:

KAI

Brand-new **Chess Master 5500** could not find the mate either, and it changed its mind a number of times, as it was deepening more and more into the position. For instance, when searching to 5 plies in 3 seconds, it found BELLE's move, *1. ... Rxh2*, evluated at +3.83. Depths of 6,7,8, and 9 ply preferred instead *1. ... Qe5*, and finally, a 10-ply seach in 48 min. 22 sec., produced *1. ... Qh4*, evaluated this time at +12.96.

MChess Pro 5.0 found BELLE's move *1. ... Rxh2* with a 4-ply search in 7 seconds, evaluated at +4.71. The 6-ply search in 50 seconds produced *1. ... Qh4*, evaluated at +7.49, and finally an 8-ply search in 5 min. 40 sec. found *1. ... d5*, with a value of +8.09.

Finally, **Kai** also tried **Crafty 12.06** on his quite powerful hardware, and to his amazement, he discovered that Crafty played BELLE's move *1. ... Rxh2* inmediately, ***from book !***, though with an 8-ply search in 33 sec., it evaluated it at +3.181. The 9-ply search preferred *1. ... Qh4*, evaluated at +3.594 in 48 sec., then the 10-ply search found once again *1. ... Rhx2* in 1 min. 54 sec, with a value of +4.427, and finally, the 11-ply search discovered in 17 min. 31 sec. *1. ... Qe5*, with a value of +5.247.

PANEK

Another gentle contributor, **Ed Panek**, tried this position on **Chess Genius 5.0**, running on a PII/266 Mhz with a 16 Mb hashtable. He found that CG5.0 preferred *1. ... d7-d5* at low depths, but then changed to *1. ... Qe7-h4* when looking at 6,7,8,9, and 10 plies in 00:35, 02:00, 04:34, 17:26, and 01:01:22, respectively, with values from +4.72 to +11.54. He let it run until it reached two hours, but it didn't change its mind after that, nor did it find any mate.

Why is it so difficult to find a best move for this position ? Because *there are many ways of winning* it. Black has many good moves, all of them winning quickly, so the number of refutations is lower than expected. If there are many best moves with very similar values, the programs have to take extra care to distinguish the *absolute best* among them.

In a real game, finding the absolute best is irrelevant as long as the chosen move is enough to win easily, so spending extra effort to discriminate between two very good moves is not justified. However, for the purposes of a test or a problem, it makes sense trying to single out the ultimate winning move.

HYATT

Looking for further opinions, I sent an e-mail to **Robert Hyatt**, which is the author of **Blitz 6.5**, in which I commented my lack of success with this position:

VA: *" ... a position of an ACM game between **BLITZ 6.5** and **BELLE**, which I have not yet succeeded in solving with **Crafty**, even with a 200-hours run !! (here *solved* means to determine in how many moves is it a mate, and what is the first beginning move. The actually move played by **BELLE**, **Rxh2**, is *not* a mate in 5 or 6, and isn't sure at all it's the best move, either ..."*

RH: *"In that game, we were running on a machine that started swapping like crazy (Washington DC, 1978) and we were getting about 1 second of CPU time every 5 minutes, so we fell into that disaster ... **BELLE** played that move thinking it won a pawn. We didn't see the "trick" with 1 second and took the rook, which led to getting killed ..."*