NOTES

IMPROVING THE PERFORMANCE OF ENDGAME DATABASES

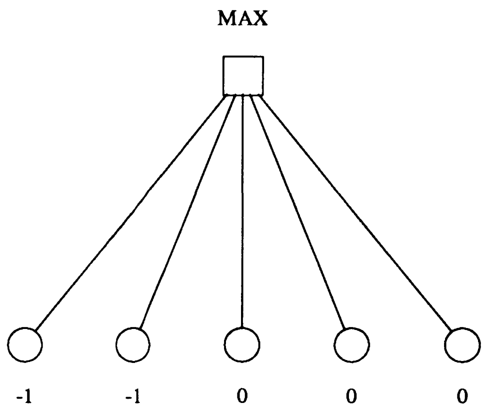
David Levy

Intelligent Chess Software Ltd.,
London, England.

Computer databases for chess endgames currently operate on a simple principle which assumes perfect play by the opponent. A program performs a 1-ply search, assigning terminal-node scores corresponding to a win, draw or loss, in accordance with information found in the database. Maximizing permits the program to select a move which leads to the optimal result. If a position is a theoretical draw, the program will be able to choose from the drawing moves, and it normally does so at random.

By admitting the possibility of imperfect play by the opponent, the program could improve its performance in practical play, such as when playing against a human opponent rather than against another database.

Consider the situation shown below.



It is MAX's turn to move and he can choose from five moves, three of which draw (for a score of 0) and two of which lose (-1). Obviously he will choose a drawing move and the game-theoretical value for MAX is 0.

By contrast, let us assume that there is a probability P of each of the losing moves being chosen by an imperfect opponent, and a probability $1/3 \times (1 - 2P)$ of each of the drawing moves being chosen. In the game-theoretical case P = 0, but in the real world P could take another value, depending on the ability of the opponent and his difficulty of discriminating between the losing and drawing moves. If in practice P = 0.01, then the backed-up value for MAX will be $1/5 \times (-1 \times 0.01 \times 2 + 0 \times 0.99) = -0.004$.

It is not difficult to see how this concept might be applied to the use of a chess database. If the program is pursuing to win a theoretically-drawn position, it performs a full-width search to an even depth, say 8-ply, assigning to the terminal nodes scores of +1 or 0 as appropriate. To the 7-ply nodes the program assigns scores of $(1 \times 0.01 \times L) / (L + D)$, where L is the number of losing moves for the opponent from the 7-ply node, and D is the number of drawing moves. Provided that at least one node in the tree has a score of -1, even if due only to the opponent's blunder, a search of the resulting 7-ply tree will provide a greater practical chance of success against an imperfect opponent than will selecting a drawing root move at random.

Various refinements to this concept suggest themselves, but are beyond the scope of this note. Although the application of this concept to chess-endgame databases is new, so far as I am aware, the idea of tree search assuming imperfect play by the opponent has been described earlier (e.g., Michie, 1981). In other words, when desperate, it is good tactics to mislead one's opponent. The opponent, if not omniscient, i.e., if a human being rather than a computer database, is therefore classified as being, in some sense, contemptible. The notion of a contempt factor has been broached in literature by Slate and Atkin (1977) for full games. Without any risk (as opposed to its use in full games) it may now be applied in endgames tabulated in a database.

References

Michie, D. (1981). A theory of evaluative comments in chess with a note on minimaxing. *The Computer Journal*, Vol. 24, No. 3.

Slate, D.J. and Atkin, L.R. (1977). CHESS 4.5 - The Northwestern University Chess Program. *Chess Skill in Man and Machine* (Ed. P.W. Frey), pp. 82 - 118. Springer Verlag, New York.

CORRECTING GRANDMASTERS' ANALYSES IN ELEMENTARY ENDGAMES

L. Rasmussen

Tranbjerg, Denmark

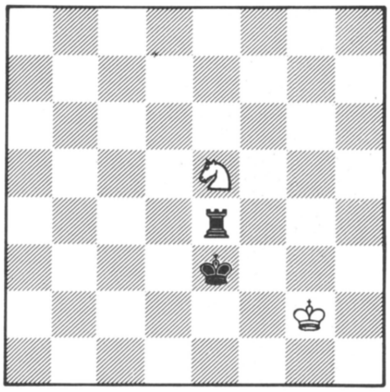
Editor's Introduction

It may be challenging to our readers to analyse this note by Lars Rasmussen dealing with the KRKN endgame, the very same subject that has been treated earlier in this issue by Denis Verhoef and Jacco Wesselius. It will be remarked that the two authors' approaches are as distinct as they could be; nevertheless they are complementary. Whereas Verhoef and Wesselius give an a priori*, classificatory treatment, Rasmussen below provides an* a posteriori*, database analysis. It is not the least of Rasmussen's merits that a home computer sufficed to hold the database and the program to consult it.*

Recently, I have undertaken a definitive analysis of some technical endgames by computer, the most interesting of which are Queen versus Rook, and Rook versus Knight. I have compared my computer's analysis with the existing endgame theory and have found some interesting results as the following KRKN examples will show.

The first instance

The position of Diagram 1 arose in the game Gosh - Gipslis (Calcutta 1979). White played 1. Nf7 and lost. Gipslis means according to ECE (rook endings II, pos. 393) that White could make a draw by 1. Nd7! Kf4 2. Nc5! This is not true, because after **1. Nd7 Kf4 2. Nc5** will follow **2. ... Re2!** Now two variations are possible (A) 3. Kf1 and (B) 3. Kh3.



White: Kg2 Ne5
Black: Ke3 Re4
Black to move.

DIAGRAM 1