8.2 Goal-Directed Search in Chess End Games

Jaap van den Herick

The game of chess has been in use as a testing ground for artificial intelligence research for over thirty years. Recent research into computer chess has focused on endgames. Programming the end game has proved to be surprisingly difficult; even programming correct play in the most elementary end games has turned out to be recalcitrant, if ‘correct’ is taken to mean: winning, within the rules of the game, whenever possible.

This paper first discusses chess thinking and pattern-based representation of knowledge and subsequently examines a model in which pattern-knowledge is combined with conventional depth-first searching. The aim of the model is to investigate whether the use of pattern-knowledge is as important for human chess skill as it is for the chess skill of the machine. To equal the strength of a chess grandmaster with an ELO rating of 2650, a chess program should at least commit no mistake in the elementary end games.

The balance between implemented knowledge and possible searching is determined by the degree in which the playing strategy is exhibited in textbooks, subject to the requirement that the balance struck should be clear to human players. The model is instantiated by an algorithm for the end game King, Knight and Pawn vs. King (KNPK). Examples are given and a comparison between machine play and the thinking of some chess grandmasters is made. Finally, an extension of the model to end games more complex than the one mentioned is made; this additional complexity is felt by human players as well.

1. Introduction

Initially, the aim of research in the field of computer chess as a testing ground for artificial intelligence research was to create a chess program playing a game indiscernable from that of a human being. Taking chess as a micro-world, this means that the aim was to construct a chess program that would pass the Turing test (Turing 1950). The idea behind it was that one might thus gain insights into human thinking.

However, chess programmers have reached the Turing milestone without simulating human chess thinking. They have developed techniques which enable the program to play an acceptable game, judging the latter by the moves produced by the program.

One of the best indications of having passed the Turing test has been the participation of computer programs as opponents in simultaneous exhibitions of grandmasters. For example, international grandmaster Pfleger did not identify a chess-playing program as one of his opponents during a simultaneous performance (Hamburg, Aug. 1980); after the exhibition he could hardly believe this to have been the case.

In the current state of chess-playing programs two important questions emerge. Firstly, on the practical side: why do programs not play well in the end game? Secondly, on the theoretical side: what has been learnt about human chess thinking from the chess program?; or, complementarily: can we describe the essence of human chess thinking and its relation to chess programs better than before?

2. The End Game

The game of chess can be divided into three parts: the opening, the middle game and the end game. Chess programs will (most of the time) survive opening problems by referring to a standard library containing the usual book variations. In the
middle game, too, they are playing quite well by now. But unless they have reached an important material advantage before the end game, accidents may mar their end play. Of course, if a computer program plays against another computer program, the other one is subject to the same accidents, hence it is not predictable which program will win.

In end games, the issue is not computing as such, but handling and trying to reach favourable positions. The highest goal in chess is to mate; however, in a normal ending, before reaching that goal, many intermediate goals (subgoals) have to be attained. It is hard to make these subgoals conceivable to a program if the latter is based on tree searching and uses a uniform evaluation function (as already pointed out by Shannon in the section "Another Type of Strategy" of the first well-known article about computer chess Shannon (1950) and more recently by Berliner (1974) and again by Bratko, Kopec and Michie (1978).

In most endings there are many exceptions, which enormously complicate the strategy the program should follow. Hence, chess programmers have divided the end game into various types of endings according to the ideas of end-game publicists such as Averbach (1979), Chéron (1970-73), Euwe (1940-41), Fine (1941) and others. Programmers have tried to find special strategies for certain configurations.

The end games can be denoted by canonical names, introduced by Michie (Berliner, 1978). In this system the abbreviations of the chess pieces are King (K), Queen (Q), Rook (R), Bishop (B), Knight (N) and Pawn (P). The King and Rook vs King ending is then denoted by KRR. More generally, the white pieces on the board are noted, starting with the King and then the black ones, also starting with the King.

Many studies of (elementary) end games have been reported from 1968 onwards. At first, they were mostly based on tree searching: Huberman (1968: KKK, KBBK, KBKK), Tan (1972: KPK; 1974: KBKP; 1977: PAWNONLY POSITIONS).

Afterwards, there was a tendency to base them on a representation of knowledge without searching: Bramer (1975: KKK; 1977: KPK), Beal (1977: KPK), Beal et al. (1980: KPK).

Another approach has emerged over the last few years, viz. the construction of data bases: Clarke (1977: KPK, KKK), Thompson (1977/79: KQKR and several other four-piece endings), Michie (1977), Arlazarov and Futer (1977/79: KRPKR, KQPK). Recent research has concentrated on the design of a computer language ALI (Advice Language 1) to simplify existing methods for handling patterns and pattern-oriented strategies (Bratko, Kopec and Michie, 1978). By using advice tables, these authors have constructed programs for the KRR and the KRRN endings (Bratko et al., 1980; Kopec et al., 1980).

More complete overviews of studies of several (elementary) endings can be found in other publications (Clarke, 1980; Frey, 1977; Van den Herik, 1980a).

In fact, no chess grandmaster has any problem at all with elementary end games such as KRR, KBBK and KBNK. He learned how to handle these endings in his youth from a chess book or by seeing them played. However, to the beginner the KBNK ending is hard to acquire, and most general chess programs do not now succeed in mating from an arbitrary position (except if they have a special routine to do so (Kent et al., 1980).

Most of the chess grandmasters are in the same position as the beginner when they have to play the KQPK ending or the KQKR ending (Fenner, 1979; Michie, 1977), though, mostly, they intuite whether the position is a winning one (the pattern is recognized as such), but the (actual) way to win is not known and still has to be discovered (searching according to patterns known).

3. The thinking Grandmaster vs. the 'thinking' machine

After reaching the milestone of programming a computer to play a game of chess indiscernable from that played by a human being, chess programmers raised their sights. The new aim was to bring their programs up to the level of grandmaster or even to that of the world champion.

The problems arising out of this new aim are: firstly, although the output of a chess machine looks like the output of a chess player, what do we know about their respective thinking processes? This is important because we want to explain, if possible, the differences in playing strength between an average chess player and a grandmaster in