three catagories of possible errors, each with its own cause and each with a different level of solvability. Shapiro and Michie describe a self-commenting facility whereby expert-supplied text fragments attached to individual attributes are recognized to form run-time explanations of decisions made by rules previously induced. A theoretical paper is given by Schrüfer. He analyses the apparent paradox that experience shows that a deeper tree-search yields better play, whereas mathematical investigation of the problem predicts less reliable results for deeper searches.

The majority of the articles is very original, and of high quality. The amount of background knowledge needed to comprehend the papers ranges from the elementary (Levy, Kopec, Hartson, and Lindner) to the sophisticated (Schrüfer).

It is a superb book for its contents as well as for its design. Like its three predecessors, it is an excellent snapshot of the state of the art in Computer-Chess research, a must for anyone seriously interested in the field.

In April 1987, the fifth conference in this series will be held in Noordwijkerhout, the Netherlands. It is hoped that this will result in another such splendid book.

LITERATURE RECEIVED

EXPERIMENTS IN DISTRIBUTED GAME-TREE SEARCHING

Jonathan Schaeffer

Technical Report TR 87-2
Computing Science Department, University of Alberta,
Edmonton, Alberta, Canada T6G 2H1

We quote the abstract of this report:

Conventional parallelizations of the alpha-beta algorithm have met with limited success. Implementations suffer primarily from the synchronization and search overheads of parallelization. This paper describes a parallel α - β searching program that achieves high performance through the use of four different types of processes: Controllers, Searchers, Table Managers and Scouts. Synchronization is reduced by having all *Searchers* apply the *PVSplit* algorithm on the subtrees they search and having a *Controller* process re-assigning idle processes to help out busy ones. Processor idle time is reduced but at the expense of increased search effort. Search overhead is reduced by having two types of parallel table management: global *Table Managers* and the periodic merging and re-distribution of local tables. Experiments show that 9 processors can achieve 5.67-fold speedups but with 20, additional processors provide little benefit. Speedups are shown to be strongly tied to the efficiency of the search and the size of tree. Given that additional resources are of little benefit in improving parallel α - β, speculative computing is introduced as a means of extending the effective number of processors that can be utilized. *Scout* processes speculatively search ahead in the tree, looking for interesting features and communicates this information back to the α - β program. In this way, the effective search depth is extended. These ideas have been tested experimentally and empirically as part of the chess program *ParaPhoenix.*