Controlled Conspiracy-2 Search*
(Extended Abstract)

Ulf Lorenz
Department of Mathematics and Computer Science
University of Paderborn, Germany

Abstract. When playing board games like chess, checkers, othello etc., computers use game tree search algorithms to evaluate a position. The greatest success of game tree search so far, has been the victory of the chess machine 'Deep Blue' vs. G. Kasparov, the best human chess player in the world.

When a game tree is too large to be examined exhaustively, the standard method for computers to play games is as follows. A partial game tree (envelope) is chosen for examination. This partial game tree may be any subtree of the complete game tree, rooted at the starting position. It is explored by the help of the \( \alpha \beta \)-algorithm, or any of its variants. All \( \alpha \beta \)-variants have in common that a single faulty leaf evaluation may cause a wrong decision at the root.

To overcome this insecurity, we propose Cc2s, a new algorithm, which selects an envelope in a way that the decision at the root is stable against a single faulty evaluation. At the same time, it examines this envelope efficiently. We describe the algorithm and analyze its time behavior and correctness. Moreover, we are presenting some experimental results from the domain of chess.

Cc2s is used in the parallel chess program P.ConNerS, which won the 8th International Paderborn Computer Chess Championship 1999.

Keyword: Algorithms and Datastructures

1 Introduction

Some games have been proven to be PSPACE-complete. As a consequence, we cannot do anything better than to examine a complete game tree when we want to find a perfect decision or if we want to know the value of the starting situation.

For most of the interesting board games we do not know the correct values of all positions. Therefore, we are forced to base our decisions on heuristic or vague knowledge. An approximation is done by the following method.

First of all, a partial game tree is chosen for examination. This subtree may be a full-width, fixed-depth tree, or any other subtree rooted at the starting position. We call this subtree an envelope. Thereafter, a search algorithm assigns

* This work was supported by the DFG research project “Selektive Suchverfahren” under grant Mo 285/12-3.
Fig. 1. Only the envelope is examined by a search algorithm.

heuristic evaluations to the leaves and propagates these numbers up the tree by
the minimax principle. Usually the envelope is examined by the help of the \( \alpha \beta \)-algorithm [5], or the MTD(f)-algorithm [1]. As far as the error frequency at the
root is concerned, there is no difference, whether or not the envelope is examined
by the \( \alpha \beta \)-algorithm or by a pure minimax algorithm. The result is always the
same, only the effort to get the result differs drastically.

The approximation of the real root value by the help of fixed-depth envelopes
leads to good results. Nevertheless, there have been found several enhancements
that form the envelope more individually. Some of these techniques are domain
independent like Singular Extensions [2], Nullmoves [3] [5] or Fail High Reductions [7]. Many others are domain dependent. The form of the envelope strongly
determines the quality of the search result.

We distinguish between two classes of game tree search algorithms. On the
one hand there are those which are built to determine the minimax value of an
envelope. The \( \alpha \beta \)-algorithm, the SCOUT-algorithm [15] [12] or SSS* [19] have
been exhaustively examined in the last 30 years.

A different class is that of the incremental searching algorithms [16] which
‘grow’ the search tree one step a time. At each step a leaf of the current tree is
chosen (selection), and the successors of that leaf are added to the tree (expansion). The new leaves are evaluated and the new heuristic values are updated
bottom up (update). In contrast to e.g. the \( \alpha \beta \)-algorithm, these algorithms need
linear space in the number of searched nodes. The advantage, however, is that
the grown trees need not be of uniform depth and the envelopes need not be
determined before the search is finished. Examples of such iterative techniques
are the Berliner’s B* algorithm [5], Palay’s probability-based method [14], and
Conspiracy Number Search. Conspiracy Number Search has been introduced by
D. McAllester [13]. J. Schaeffer [19] has interpreted the idea and has developed
a search algorithm that behaves well on tactical chess positions. Lorenz et al.
[11] have presented first ideas of how to build an algorithm which is able to do
the same job more efficiently.

The startup point of Conspiracy Number Search (CNS) is the observation
that, in a certain sense, the \( \alpha \beta \)-algorithm computes decisions with low security.

The changing of the value of one single leaf (e.g. because of a fault of the
heuristic evaluation function) can change the decision at the root. Thus, the
\( \alpha \beta \)-algorithm takes decisions with security (i.e. conspiracy) one.