RESEARCH RE: SEARCH & RE-SEARCH

Aske Plaat

plaat@theory.lcs.mit.edu

RESEARCH
RE: SEARCH &
RE-SEARCH

This book is number 117 of the Tinbergen Institute Research Series. This series is established through cooperation between Thesis Publishers and the Tinbergen Institute. A list of books which already appeared in the series can be found in the back.

RESEARCH RE: SEARCH & RE-SEARCH

PROEFSCHRIFT

TER VERKRIJGING VAN DE GRAAD VAN DOCTOR AAN DE ERASMUS UNIVERSITEIT ROTTERDAM OP GEZAG VAN DE RECTOR MAGNIFICUS

PROF.DR. P.W.C. AKKERMANS M.A.

EN VOLGENS BESLUIT VAN HET COLLEGE VOOR PROMOTIES.

DE OPENBARE VERDEDIGING ZAL PLAATSVINDEN OP DONDERDAG 20 JUNI 1996 OM 13.30 UUR

DOOR

ASKE PLAAT
GEBOREN TE ODOORN.

Promotiecommissie

Promotores:	prof.dr. A. de Bruin prof.dr. J. Schaeffer
Overige leden:	dr. J.C. Bioch prof.dr.ir. R. Dekker prof.dr. H.J. van den Herik dr. W.H.L.M. Pijls (tevens co-promotor)

RESEARCH RE:
SEARCH & RE-SEARCH

Aske Plaat

ABSTRACT

Search algorithms are often categorized by their node expansion strategy. One option is the depth-first strategy, a simple backtracking strategy that traverses the search space in the order in which successor nodes are generated. An alternative is the best-first strategy, which was designed to make it possible to use domain-specific heuristic information. By exploring promising parts of the search space first, best-first algorithms are usually more efficient than depth-first algorithms.

In programs that play minimax games such as chess and checkers, the efficiency of the search is of crucial importance. Given the success of best-first algorithms in other domains, one would expect them to be used for minimax games too. However, all high-performance game-playing programs are based on a depth-first algorithm.

This study takes a closer look at a depth-first algorithm, Alpha-Beta, and a best-first algorithm, SSS*. The prevailing opinion on these algorithms is that SSS* offers the potential for a more efficient search, but that its complicated formulation and exponential memory requirements render it impractical. The theoretical part of this work shows that there is a surprisingly straightforward link between the two algorithms—for all practical purposes, SSS* is a special case of Alpha-Beta. Subsequent empirical evidence proves the prevailing opinion on SSS* to be wrong: it is not a complicated algorithm, it does not need too much memory, and it is also *not* more efficient than depth-first search.

Over the years, research on Alpha-Beta has yielded many enhancements, such as transposition tables and minimal windows with re-searches, that are responsible for the success of depth-first minimax search. The enhancements have made it possible to use a depth-first procedure to expand nodes in a best-first sequence. Based on these insights, a new algorithm is presented, MTD(f), which out-performs both SSS* and NegaScout, the Alpha-Beta variant of choice by practitioners.

In addition to best-first search, other ways for improvement of minimax search algorithms are investigated. The tree searched in Alpha-Beta's best case is usually considered to be equal to the minimal tree that has to be searched by any algorithm in order to find and prove the minimax value. We show that in practice this assumption is not valid. For non-uniform trees, the real minimal tree—or rather, graph—that proves the minimax value is shown to be significantly smaller than Alpha-Beta's best case. Thus, there is more room for improvement of full-width minimax search than is generally assumed.

Preface

I would like to thank my advisors Arie de Bruin, Wim Pijls, and Jonathan Schaeffer, for being just as eager as I to understand minimax trees. We have spent countless hours discussing trees and algorithms, trying to find the hidden essence. Their experience with minimax search, as well as their different backgrounds, were of great value. They know how much this research owes to their willingness to spend so much time discussing ideas, combining theory and practice, trying to understand trees. It has been a great time, for which I thank you three.

This work started in May 1993 in Rotterdam, the Netherlands, with Arie and Wim, at the department of Computer Science of the Erasmus University. I thank my colleagues then at the department of Computer Science for creating an interesting environment to work in. I thank Jan van den Berg for helping me experience how enjoyable research can be, which made me decide to pursue a PhD. It has been good talking to Roel van der Goot on subjects ranging from parsing to politics. Maarten van Steen answered many questions on parallel computing. Harry Trienekens and Gerard Kindervater know much about parallel branch and bound as well as operating systems. Reino de Boer knows LaTeX inside out.

From October 1994 to October 1995 I spent most of my time at the University of Alberta in Edmonton, Canada, with Jonathan Schaeffer. Working together with someone with so much experience and energy, and so many ideas, has been an extraordinary experience. Jonathan, I thank you for giving me the opportunity to do research with you and to learn from you; but most of all, I thank you for good times. In Edmonton many people did their best to make my stay as pleasant as possible. Jonathan and Stephanie Schaeffer made sure that I felt at home, helping me wherever and whenever they could. I thank the people of the department of Computing Science of the University of Alberta for creating a good working environment. I have had many discussions on minimax search with Mark Brockington, author of Keyano, one of the top Othello programs. Yngvi Björnsson and Andreas Junghanns, who started building a new chess program, were always keen to discuss new ideas too. Rob Lake answered my troff questions. I thank all of the members of the Games research group for making both the scheduled and the informal meetings interesting and enjoyable.

A number of people took the time to comment on or discuss various parts of this research. I thank Victor Allis, Don Beal, Hans Berliner, Murray Campbell, Rainer Feldmann, Jaap van den Herik, Hermann Kaindl, Laveen Kanal, Rich Korf, Patrick

van der Laag, Tony Marsland, Chris McConnell, Alexander Reinefeld, and especially George Stockman, the inventor of SSS*, for their helpful comments.

I thank Victor Allis for teaching a class at the Free University in Amsterdam, in September 1993, in which he explained the concept of the minimal tree in a way that (to me) was strangely reminiscent to solution trees, planting a seed of the SSS*–Alpha-Beta relation.

In addition, research meetings with Victor Allis, Henri Bal, Dennis Breuker, Arie de Bruin, Dick Grune, Jaap van den Herik, Gerard Kindervater, Wim Pijls, John Romein, Harry Trienekens, and Jos Uiterwijk on search, combinatorial optimization and parallelism provided a valuable broader perspective for this work.

I thank Mark Brockington, Arie de Bruin, Jaap van den Herik, Andreas Junghanns, Wim Pijls, Jonathan Schaeffer, and Manuela Schöne for their careful and detailed reading of drafts of this thesis. Their many comments and suggestions have greatly enhanced its quality. Furthermore, Peter van Beek and Rommert Dekker made valuable suggestions for improvement of parts of this work. Thanks to Gerard van Ewijk for reporting an especially nasty bug.

Through all of this, Saskia Kaaks has been a great friend. I thank you for everything, with love.

Rotterdam,
March 31, 1996.

Contents

List of Figures