JAIST Repository
https://dspace.jaist.ac.jp/

Title	コンピュータ将棋における高精度キラーヒューリスティックの研究
Author(s)	橋本，隼一
Citation	
Issue Date	2007-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/3583
Rights	
Description	Supervisor:飯田　弘之，情報科学研究科，修士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

High precision killer heuristics in the domain of computer Shogi

Hashimoto Junichi (510078)

School of Information Science,
Japan Advanced Institute of Science and Technology

February, 2007

Keywords: Game tree search, Computer Shogi, Killer Heuristics.

This study aims to realize a program which defeats the human champion at Shogi. Since computers were born, to design a program which plays a game has been one of the main theme of artificial intelligence. This is because people think the ability of playing game is a symbol of human intelligence. So in fact in 1997, it became big news that the Deep Blue, which was a chess computer defeated Kasparov who was a world champion of those days. Now, 10 years later from that news, Shogi programs achieved the same strength as top amateur players but are still less than expert players unlike chess because Shogi is much more complicated than chess.

The zero sum two-person finite games with perfect information, such as chess or Shogi, are often modeled as a tree, in which the root node corresponds to a start position and leaf nodes to winning, drawing or loosing positions. A basic algorithm, which is known as Minimax, searches the tree in order to find the best move. Imagine a game when the height is constant d and there are b successors in each position. In this case, the algorithm has to visit all nodes and evaluate b^d positions to find the best move. $\alpha\beta$ method is a famous algorithm which can find the same result by evaluating $O(b^{d/2})$ positions in the best case. In other words, $\alpha\beta$ method may search a game tree with double height in the same time.

The efficiency of $\alpha\beta$ method depends on the order of successors. It works most efficiently if the successors are sorted in the *best* order. Of course, the

---

Copyright © 2007 by Hashimoto Junichi

best move is not found before the search finishes. So, in order to improve the search efficiency, it needs to judge if a move is good or not without any search routine.

Generally, domain dependent heuristics can answer the questions like what kind of moves are good or bad. Check and capture moves in chess and Shogi are good instance of this heuristics. There are domain independent heuristics, too. Killer heuristics, which is the main theme of this study, is one of them.

Killer heuristics is a technique to use the best move which was found in similar node with the current searched node. This best move is called a killer move. Killer heuristic is important because the killer moves are generally searched first in a node for searching efficiency. This heuristic is known as good and used in many programs today. However, detailed study about killer heuristic was little.

At first, we make clear up the characteristics of classical killer moves; the best move of sibling node (Brotherbest), the best move of sibling node which last move is pass (Passbest), and the best move which last move is same to the last move of current node (Premovebest). *Efficiency* is defined as quantitative measure and the characteristics are compared by this measure from some point of view, such as game progress dependency, search depth dependency and move category distribution.

Next, we tried to improve the efficiency of these killer moves based on the analysis and propose a new method named “Context Killer Heuristic”. The heuristics use the best move of a node which last n-moves are same to the move sequence in a current node, We got this idea from Premovebest, so the propsed method is may be said as a extention of it. The moves stored by the heuristics have a relation to the sequence of moves unlike classical killer moves. The implemented program ($n = 2$) defeated an old program and can answer the next-move questions more. The test results show that the implemented program became stronger than the old one in a statistical sense.

Basically, killer heuristics affect to only search speed. The test results are caused by the use of killer move in selective search. Near the terminal nodes, this method searches only a simple move such as capture or recapture. While proposed killer heuristic stores more complicated move, so it can complements week point of selective search.

Today it seems almost all basic algorithms and ideas for game-tree search are already on the table. We think future works will lay emphasis on how to use *on-line* and *off-line* information well and how to improve the basic algorithm. This study is a first step in this direction.