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

Title	An Analysis of Voting Algorithm in Games [課題研究報告書]
Author(s)	Sato, Yuichiro
Citation	
Issue Date	2013-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/11345
Rights	
Description	Supervisor:Iida Hiroyuki, 情報科学研究科, 修士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

An Analysis of Voting Algorithm in Games

Yuichiro Sato (1110030)

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

February 6, 2013

Keywords: 3-Hirn, consultation algorithm, multiple choice system, voting algorithm, decision making.

In game engine research, improving performance by multiple choice systems are researched. In 1985 Althöfer started 3-Hirn with a seminal experiment in the game of chess. In chess, the 3-Hirn consists of two different strong chess engines and one human weak chess engine. After Althöfer's delightful success, Obata et al. reported consultation algorithm where many game engines choose one move by simple majority rule, which improves the performance on Shogi game. To make many engines, they apply noises on an evaluate function which BONANZA, a strong engine has. Also, they reported consultation of three strong Shogi programs: YSS, GPS, and BONANZA plays better games than any of the three individual programs. Even though the advantage of multiple choice systems is clear in practice, the reason why these systems work well is not clear. In multiple choice systems, contribution of weak engines improves total output. This is a paradox. In this thesis, we report a reasonable explanation of this paradox.

First, we introduce a mathematical representation of consultation algorithm. This representation is enough powerful, therefore it is able to make a biggest framework for the analysis of consultation. To analyze consultation, the most important point of view is that an engine is a program which chooses a move in a position in a deterministic way, except if a random algorithm is adopted. In other words, if the same position appears

Copyright © 2013 by Yuichiro Sato

many times during the game, the engine suggests the same move every time. Therefore, we can represent an engine as a mapping from positions to moves. To evaluate each move that is chosen by engine, let us suppose that there exists a perfect player, and it has a perfect evaluation function. An evaluation function is a mapping from Cartesian products of positions and moves to real numbers. This perfect player perfectly evaluates all the moves in a position. To analyze the consultation algorithm, we need to make a voting matrix. The voting matrix is a matrix which is made of voting results by game engines in the consultation. Then, count up the number of same moves which engines suggest and choose the majority as the consultation result. Also, let us define as the consultation algorithm works well if and only if the consultation algorithm averagely chooses better moves. We do not expect the consultation algorithm always chooses a best move, but it averagely chooses a better move. Therefore, this definition is reasonable. Under these definitions, we continue to analyze the consultation algorithm in this thesis.

Obata et al. reported experiments of consultation algorithm with random numbers. In these experiments, they prepare many engines that contribute to consultation which are generated with random numbers in the evaluation function of the original strong Shogi engine, BONANZA. To treat this situation, let us introduce noise functions. Noise function is a function which is generated by a fixed list of random numbers, once it is generated, it is fixed. Noise functions describe the difference between the original engine and noisy engines. A noise function depends on positions and the original engine, because some positions or engines could be sensitive for noises, some could be not. According to an analysis, you can see the voting matrix of the consultation algorithm with random numbers depends only on noise functions. Therefore, one tends to conclude that the consultation algorithm is a random algorithm.

To keep analysis going, stochastic algorithm is needed to be considered. We need a partition of moves which is made according to a classification such that whether the move is better, equivalent or worse than the specific original move in a position. What we expect for the consultation algorithm is that the majority of noisy engines choose a better candidate than the original move by chance. Suppose there exist exact probabilities of the

original engine changing its move from the original move to another move. Then we can derive the condition of the consultation algorithm works well. The consultation algorithm works well if expected improvement of consultation is greater than expected reduction of consultation where expected improvement/reduction is an expectation value of improvement/reduction which is gain if the original engine is replaced by another engine. Also, from experimental condition, the expected improvement of each noisy engine needs to be less than the expected reduction of each noisy engine. This is the condition which the consultation algorithm works well. Also, we can calculate probabilities of consultation algorithm choosing a better or worse move in a position. The lower bound of the consultation algorithm choosing a better move and the upper bound of the consultation algorithm choosing a worse move is derived. These probabilities are useful to calculate the lower bound of expected improvement of the consultation algorithm and the upper bound of expected reduction of the consultation algorithm. If the lower bound of expected improvement is greater than the upper bound of expected reduction, the consultation algorithm works well in any case. Therefore, we can predict whether the consultation algorithm works well or not in a situation. These results have no contradiction with the result which is reported in previous works. If we specify legal moves as a better or worse move, then our result derives an equation which is reported in the previous works. Therefore we conclude we get the exact solution for the consultation algorithm.

3-Hirn is a system which is made of two engines and a human. In this system, the human chooses a better candidate from candidates which are suggested from engines.The human is weaker than each engine in the game, but this system is stronger than each engine. The weak player's contribution improves the final outcome. It looks unreasonable that this system becomes stronger than any individual engines. However, there is a reasonable explanation of this puzzle in almost the same context as consultation algorithm. An engine has an evaluation function to improve its search speed. This function tends to be rough compared to the one a human has. Even though the engine has a rough evaluation function, the computer is really powerful, so it is able to cover the weakness by making a vast search tree. For human, the situation is opposite. A strong human player has a

really good evaluation function and cuts the search tree efficiently. Therefore, if the human is good at evaluation but not for search and engines are oppositely good at search but not for evaluation, 3-Hirn works well.

In this thesis, we clearly explained the origin of an advantage of consultation algorithm with random numbers, and 3-Hirn. In this thesis a new definition is derived of the necessary and sufficient condition for consultation algorithm working well. This new definition is an improvement of the existing one because it considers all choices. Our analysis has a possibility to be applicable to human activities not only engine.