Improving Game Board Evaluator with Genetic Algorithms

Ping–Chung Chi[1] and Dana S. Nau[2]
University of Maryland
Collge Park, Md 20742

I. Introduction

Evaluation functions used in game tree search usually consist of parameterized functions of various board features. To create a good evaluation function, it is necessary to find good values for the parameters. In the past, this has usually been done by a combination of human intuition and a trial-and-error approach.

We have developed an algorithm which will automatically find near-optimal parameter values for evaluation functions. This algorithm, called AGAPE, is a simulated evolution algorithm similar to a genetic algorithm [Holland, 75]. Analytical studies of AGAPE suggest that it converges faster than an ordinary genetic algorithm would, and empirical studies of AGAPE show that it produces good evaluation functions in the game of kalah.

II. Outline of Approach

We know of three previous pieces of work on automated construction of evaluation functions. Samuel [67] automated the construction of the evaluation function for his checker playing program, by attempting to make the function conform to the choice of moves made by human experts. Pearl [84] suggested constructing evaluation functions automatically by using statistical sampling, with the goal of making the function return the probability that the board being evaluated was a forced win. Abramson [87] developed a method for constructing evaluation functions to return the "expected outcome" value of a board.

In each of the above pieces of work, the approach was to formulate a guideline for what kind of the value the function should compute, and find a way to construct evaluation functions conforming to this guideline. With each of these approaches, a significant difficulty is that it is not clear how well the guideline corresponds to what the evaluation function really needs to do: enable a game tree search to select the best move for winning the game.

An alternate approach would be to dispense with the guideline altogether, and select the evaluation function based directly on how well it does against other evaluation functions. For example, suppose we have a set of board features $\{x_1,x_2,\ldots,x_n\}$, and suppose we consider the set of all second-degree polynomials in $\{x_1,x_2,\ldots,x_n\}$. Without considering the meaning of any one of these mappings, a "best" one of them can be selected according to how well each of them performs.

A simple-minded way of performing this kind of optimization would require an astronomical amount of computation time: the number of possible mappings is huge, and for each mapping a large number of games would need to be tested to get an accurate estimate of the performance. However, the amount of computation necessary can be drastically reduced by using Monte Carlo techniques based on simulated evolution.

Fogel, Owens, and Walsh [66] pointed out the versatility of their simulated evolution technique. However, they only dealt with problems which can be (approximately) solved with finite state automata. Holland [75] pointed out ways to handle solution domains with string or numerical representations in general with genetic algorithms, which can be thought of as a kind of simulated evolution technique. However, most work in genetic algorithms has dealt with deterministic domains. By extending the idea of genetic algorithms to handle data which is probabilistic in nature, we have developed an algorithm called AGAPE (Approximate Genetic Algorithm with Proportion Estimations), which is described in detail in the complete paper. AGAPE can select a near-optimal evaluation function from among a wide range of possible candidate functions, based on the performance of these functions against a common opponent. We have demonstrated both analytically and empirically that AGAPE can be efficient in terms of the number of games needed to be tested.

III. Empirical Results

The effectiveness of AGAPE was tested using the game of kalah with three bottom holes. In previous studies of kalah, the evaluation function which has been

[1] This work has been supported in part by a Systems Research Center fellowship.

[2] This work has been supported in part by the following sources: an NSF Presidential Young Investigator Award to Dana Nau, NSF NSFD CDR-85-00108 to the University of Maryland Systems Research Center, IBM Research, and General Motors Research Laboratories.

used has usually been the "kalah advantage" (i.e. the number of stones ahead). However, the kalah advantage has some deficiencies as an evaluation function for kalah. For example, the number of stones left (in the playing holes) plays an important role in the strength of a board. By analyzing some of the characteristics of kalah, we produced (by hand) a new evaluation function which performs significantly better than the kalah advantage. When we used AGAPE to produce an evaluation function, the evaluation function produced by AGAPE outperformed our hand-produced evaluation function as well as the "kalah advantage" evaluation function.

IV. Summary

We have developed an algorithm, AGAPE, for use in constructing evaluation functions for games. AGAPE uses simulated evolution techniques to select an approximately optimal evaluation function out of a large class of possible evaluation functions. We have produced mathematical results establishing the efficiency of AGAPE, and have demonstrated the effectiveness of AGAPE in the game of three-hole kalah. These results should be viewed as preliminary, for several reasons.

(1) Although AGAPE seems to perform well on three-hole kalah, it has not been tested on any other game.

(2) AGAPE ranks evaluation functions according to how well they do against a common opponent--but it is not entirely clear how this relates to how well they would do against each other. In fact, it can be shown that for for some games, how well the evaluation functions do against each other is not even a well-defined concept.

Despite these limitations, our results suggest that AGAPE might be a useful technique for the construction of evaluation functions--but to test this hypothesis, more extensive research should be done on other games.

References

[Abramson, 1987] Abramson, B., "The Expected-Outcome Model of Two-Player Games," Ph. D. Dissertation, Columbia University (1987).

[Fogel, Owens, & Walsh, 1967] Fogel, L. J., A. J. Owens, and M. J. Walsh, *Artificial Intelligence through Simulated Evolution*, John Wiley & Sons (1966).

[Holland, 1975] Holland, J. H., *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, (1975)

[Pearl, 1984] Pearl, J., *Heuristics*, Addison-Wesley, Reading, MA (1984).

[Samuel, 1967] Samuel, A. L., "Some Studies in Machine Learning Using the Game of Checkers. II--Recent Progress," *IBM Journal of Research and Development*, Vol 2, pp. 601-617, (1967).