Genetic Algorithm Learning in Game Playing with Multiple Coaches

Chuen-Tsai Sun Ying-Hong Liao Jing-Yi Lu Fu-May Zheng

Department of Computer and Information Science
National Chiao Tung University
Hsinchu, Taiwan 30050
E-mail: ctsun@cis.nctu.edu.tw

Abstract— **We explore the concept of diversified selection by employing multiple coaches in a game-playing program with a genetic algorithm (GA) based learning module. Although the importance of diversity in choosing offspring in a gene pool has been addressed in the past, few authors have discussed how to maintain diversity in real-world applications. Most existing suggestions are based on a balanced distribution of candidates, but this is not a realistic assumption for search problems in a multidimensional space. We show in this paper that when more than one coach is used in a game-playing environment, the collective learning result is better than other learning curves in which only a single coach is involved, no matter whether the coach is the best one or the worst one. We also use expanded chromosomes for measuring position scores in a static evaluation function to achieve improved learnability. Our work can be classified under the Evolutionary Strategy paradigm mentioned in [3].**

I. Introduction

To achieve the goal of continual evolution, it is important to keep the gene pot boiling. In other words, a successful evolutionary strategy should force chromosomes to exhibit diversity so that the evolutionary process will not suffer from early saturation brought on by uniformity. Although this issue has long been discussed by researchers, very few satisfactory strategies have been suggested because of the difficulty of achieving a balance between fitness and diversity.

The simplest way to achieve diversity is by increasing mutation rates or by introducing more radical mutation operators, such as the *partial complement operator* discussed in [2]. It has been found, however, that this approach usually results in poor performance. Increasing the population size to compensate for this side effect is not a good idea because it results in poorer learning efficiency. Another method of enhancing diversity is to consider the distribution of individuals as an evaluation factor in the selection process, for example, in [?]. This idea was also employed in the *rank-space method* suggested by Patrick Winston in his AI textbook [9]. However, it is generally difficult to define distance adequately in a multidimensional search space. Recently, a systematic discussion of *disruptive selection* has been presented that provides us with a better understanding of the behavior of a nonmonotonic fitness function [4]. This method successfully solves problems such as optimizing a *needle-in-a-haystack* function, which is traditionally GA-hard. Nevertheless, in general, we cannot establish a balance between fitness and diversity merely by selecting some of the most unfit members. For this reason, in this paper we explore the possibility of employing multiple standards of success to maintain a diversified population. We take game-playing as the field of experimentation.

In the next section, we review the basics of game-playing and describe the reasons behind our decision to use multiple-coached GA mechanisms for learning. Simulation results are discussed in Section III. The paper closes with a brief summary and suggestions for future work.

II. Using GAs in Game Playing

Most successful game-playing programs employ a heuristic search that uses a *static evaluation function* to guide the direction of the search. A typical linear evaluation function has the following form:

$$h = w_1 f_1 + w_2 f_2 + \cdots + w_n f_n,$$

where the function value h is called the *static evaluation* of a game board configuration, the f_i's are *features* that play important roles in game-playing strategies, and the w_i's are *weights* that indicate the relative importance of the features. With such an evaluation function, we can apply the well-known *minimax search* algorithm as well as *alpha-beta pruning* techniques. The power of a game-playing program is thus determined by two factors: how the discriminating features are selected and how the weights are assigned. These two factors have been the focus of a great deal of research since Arthur Samuel published his seminal work on machine learning [8]. In this paper we concentrate on the second factor using genetic algorithms.

Genetic algorithms (GAs) avoid local minima or suboptimal results. Consequently, GAs are ideal for searching the weight space of the heuristic function used in game-tree algorithms. In the design of the GA learning algorithms, we take into consideration many important factors, e.g., those summarized in [3]. We treat a chromosome string as

0-7803-1899-4/94 $4.00 ©1994 IEEE

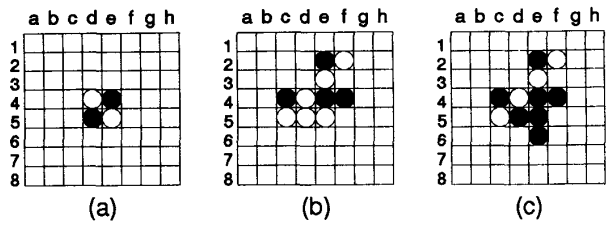

Figure 1: *Game of Othello*. (a) shows the initial set-up; after Black plays to e6 on (b), the board configuration changes to (c).

a vector of real-valued parameters which represent the coefficients of the game-playing heuristic function. Although it has been shown how GAs can be used for multiobjective optimization [1], selecting the most fit members at all times will not guarantee continuing evolution. As is well known, to keep the evolutionary process going, we have to take *diversity* into consideration. While some researchers have suggested considering population distribution or employing quality scores, we provide an approach that uses multiple coaches to guarantee diversity.

We shall test our idea in the domain of *Othello*, which is a challenging game for human players because of the difficulty of envisioning the drastic board changes that result from moves. On the other hand, this game is of reasonable complexity for the task of learning because of its moderate branching factor. World class Othello programs have been developed by Rosenbloom [7] and Lee and Mahajan [6]. They applied techniques such as iterative deepening, move ordering, pattern classification, and Bayesian learning in their work. Their approach required a great deal of human expertise for both game-playing strategies and learning mechanisms. Our goals in this paper include using GAs to identify the weights of importance automatically and exploring the possibility of improving learning effectiveness/efficiency by using multiple coaches.

Before explaining how GAs can be applied to game-playing strategy adaptation, we shall briefly describe the domain of Othello. The game is played by two players, Black and White, on an 8 by 8 board, which is initially set up as shown in Figure 1(a). Black starts the game by placing a black piece on any empty square on the board adjacent to one or more of White's pieces. By this move, Black captures the adjoining white pieces, which are then flipped over to show their black side. Figures 1 (b) and (c) show an example. The players take turns placing pieces on the board until neither player can make another move. The player with the most pieces on the board is then declared the winner. Refer to [5] for further details.

III. Simulation Results

To play Othello games, our five coach programs and GA learning programs employ features such as position, piece advantage, mobility, and stability. Before beginning this project we organized a local computer Othello tournament with about 60 competitors. The top five players in the tournament were selected as the five coach programs. A full-width minimax search with alpha-beta pruning was commonly used in the coach programs. To deal with time constraints, some of the programs employed the textbook iterative deepening strategy, i.e., they performed a full N-ply search before attempting an $N + 1$-ply search. We will call the coaches $Coach_1$, $Coach_2$, $Coach_3$, $Coach_4$, and $Coach_5$, respectively, in order of increasing power. In other words, $Coach_5$ was the program that won the tournament. Note that the primary purpose of this paper is not to develop a world-class Othello program. Instead, our focus is on studying the behavior of a GA program when it learns from multiple coaches. Our approach is based on the realistic assumption that fitness is usually defined by more than one autonomous agent in the environment.

All the programs were coded in C++. The population size was set to 200. Each member of the population took about 30 seconds on a 486/DX-33 PC to finish a game. Evolutionary behavior was visible even with the relatively small population size. We employed fitness-proportional selection in this project. Each member of a certain generation played two games with one coach. Each pair of opponents took turns starting the two games. The fitness score was defined as the sum of the piece advantages of the two games. In terms of reproduction, we found that recombination, especially crossover, was very productive in yielding good offspring.

To test the validity of the proposed model, we applied it in several learning situations. In our first try, we used a game-playing heuristic function with six features: board position measure, piece advantage, current mobility (defined as the difference in the number of possible moves between the GA player and the coach), stability (number of unflippable pieces), potential mobility 1 (number of opponent's pieces adjacent to empty squares), and potential mobility 2 (number of empty squares adjacent to opponent's pieces).

For comparison, we initially let the GA player learn from a single coach. Learning curves demonstrating the evolutionary processes against $Coach_1$ and $Coach_5$ are shown in Figure 2 and Figure 3, respectively. As expected, the best individual in each case learned to beat the corresponding coach because it successfully found a good weight combination for its heuristic function. The GA player won by a larger margin against $Coach_1$ than against $Coach_5$, since $Coach_5$ was the tougher of the two coaches.

Now we come to the interesting question: If the GA player has an opportunity to learn from all five coaches, will the result be better than that in the cases where only a single coach is involved, regardless of whether the coach is the champion or an ordinary player? This question is not trivial, because we know that learning with multiple goals may result in useless interpolation. The only way to answer this question is by doing experiments.

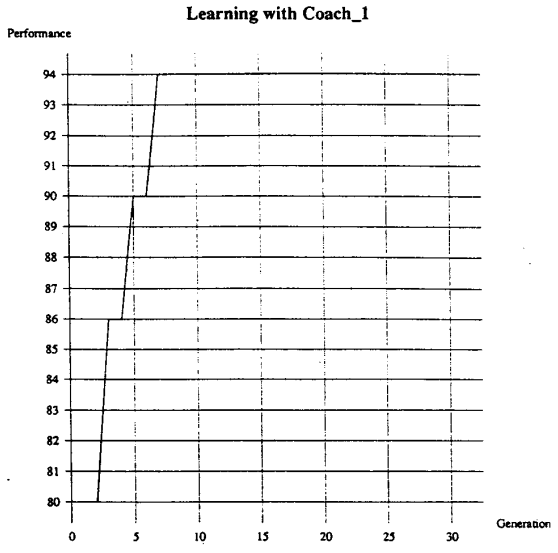

Figure 2: *Evolutionary Curve:* Learning with $Coach_1$.

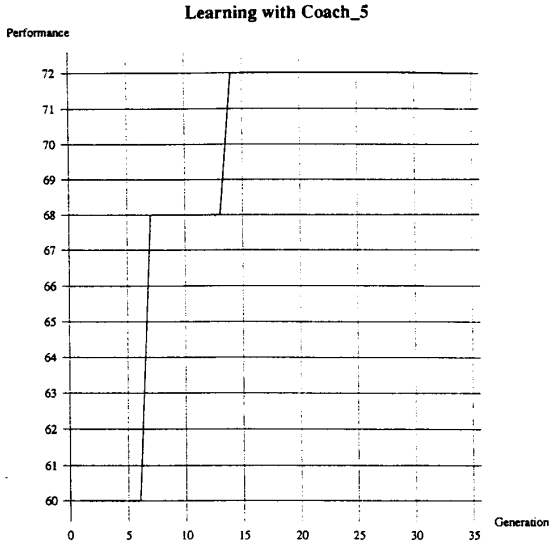

Figure 3: *Evolutionary Curve:* Learning with $Coach_5$.

In the initial population for training with multiple coaches, each individual was scheduled to play two games with $Coach_1$. In any later generation, only those individuals who had beaten $Coach_i$ and were not changed by the reproduction operators would be scheduled to play against $Coach_{i+1}$. The evolutionary learning results are shown in Figure 4.

From the learning curve we can observe that every time an individual developed a better weight combination, the result was likely to ripple out to affect its performance against other coaches. Moreover, we observe that the best

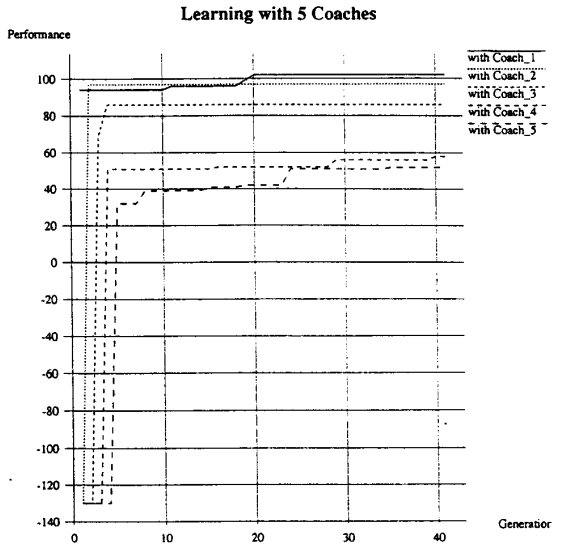

Figure 4: *Evolutionary Curves:* Learning with all five coaches. Each coach is denoted by a number.

players against different coaches were in general different individuals, thus the ripple effect was caused by crossover. For $Coach_1$, obviously, the learning effect was better than that with single-coached training. The best player in the final population against $Coach_1$ achieved a total score of more than 100, which was better than the result shown in Figure 2. In this case, keeping chromosomes with greater variety helped to produce a performance breakthrough. However, the best player against $Coach_5$ could not achieve the final score in Figure 3. A possible explanation for this is that since every GA player against $Coach_5$ had to beat the other coaches first, it might have over-committed itself to some features.

After we found that the GA produced satisfactory results, we tried to enhance its potential ability to identify lower-level features. The current board position measurement plays a pivotal role in Othello. In previous designs the importance of each square was determined by human experience. To avoid this kind of high-level human involvement in learning, we broke the configuration feature of the whole board down into individual position features. In the expanded chromosome encoding. we represented the board position measure as 10 values, taking advantage of the symmetry of the Othello board. Furthermore, we found that a single feature of relative mobility was not enough to reflect the importance of the total number of legal moves in different game-playing stages, such as opening moves and closing moves. Hence, we broke that feature into two subfeatures: the number of possible moves for the player and the number of possible moves for the coach. Thus we had a total of 17 features in the revised game-playing heuristic function.

The learning curve with this expanded chromosome for-

mat is shown in Figure 5. As expected, the learning speed was reduced because it was difficult to find the correct setting of importance for each position. However, interaction between coach curves occurred much more frequently than before.

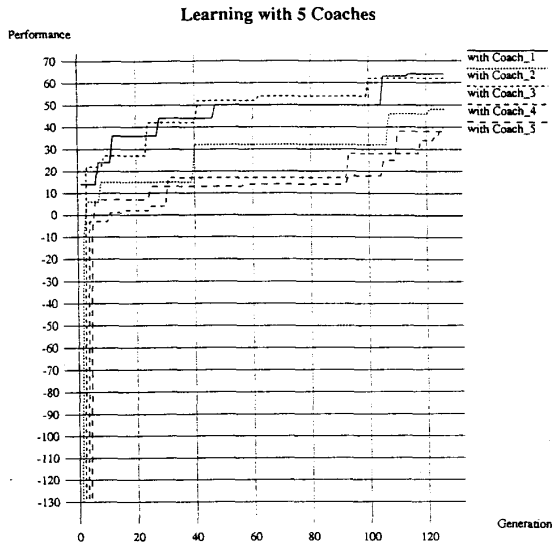

Figure 5: *Evolutionary Curves:* Learning with all five coaches, with expanded position features.

Another interesting point is that the order of the coaches in terms of power, from the GA player's point of view, was not quite the same as the ranking found in the tournament. For example, the GA player beat $Coach_3$ by a larger margin than $Coach_2$, whereas the former beat the latter in the tournament. Moreover, as pointed out in [10], an interesting relationship could develop between average human players (Hs), expertise-based game-playing programs (Es), and GA-based learning programs (Ls). This relationship would be cyclic, just like that between the scissors, paper, and stone in the well-known children's game: although Es outperform Hs, Ls learn to beat Es, and Hs are, in most cases, able to beat Ls easily. The apparent reason for this relationship is that Ls over-commit themselves to the weak points of Es and develop a narrow strategy that is not good for generalization.

Since it is interesting to see whether training with multiple coaches enables Ls to avoid this drawback, we had the programs play against the commercial Othello program *Reversi for Windows*^{TM}. There are four levels in Reversi: *Beginner*, *Novice*, *Expert*, and *Master*. We chose the *Master* level for our program players, including all the coaches and the best GA players against each coach. Since we had two versions of GA encoding, we use the superscripts 6 and 17 to denote the number of genes used in the chromosome format. For example, $Best_2^{17}$ denotes the best player against $Coach_2$ using 17 features. We gave our programs 0.3 seconds for each move; this was in general less than the time consumed by *Reversi*, which typically used 2 to 3 seconds for its mid-game search. The results are summarized in Table 1.

Table 1: *Play against Master.* Two games were played. The table shows the sum of the final scores of our program players against *Reversi Master*.

$Coach_1$	+38	$Best_1^6$	+28	$Best_1^{17}$	-38
$Coach_2$	+6	$Best_2^6$	-38	$Best_2^{17}$	-6
$Coach_3$	+8	$Best_3^6$	+6	$Best_3^{17}$	-60
$Coach_4$	-10	$Best_4^6$	-12	$Best_4^{17}$	+2
$Coach_5$	+20	$Best_5^6$	+11	$Best_5^{17}$	-18

If we consider the $Best_i^{17}$'s, the cyclic relationship is quite clear. $Best_i^{17}$ beat $Coach_i$, $Coach_i$ beat *Master*, and *Master* beat $Best_i^{17}$, with $Best_4^{17}$ the only exception. However, if we consider the $Best_i^6$'s, which employed more human knowledge in terms of position evaluation than the $Best_i^{17}$'s, the outcome was quite different. In this case, we had three GA players, $Best_1^6$, $Best_3^6$, and $Best_5^6$, who beat both their coaches and the Reversi Master program. Although the results enhanced our belief in the learning ability of GAs, they also left us many questions to answer in the future.

IV. Concluding Remarks

The maintenance of diversity is an importance issue worthy of further investigation. In addition to the traditional approach of trying to construct a balanced criterion between chromosome diversity and evolutionary convergence, we can explore the promising alternative of using multiple standards of survival or success. Game-playing provides a fertile ground for experiments in this area. In this paper we have focused primarily on two aspects of game-playing: learning behavior with multiple coaches and the effects on learning of using different levels of features.

Our future work will include experiments on cooperation among elite individuals at the end of the evolutionary process. The motivation for this future work is that at the end of the experiments reported here, the best players against different coaches were generally different; no single player was able to beat all five coaches. Obviously, the GA failed to put all pieces of important information together in a single chromosome. This suggests that a type of higher-level mechanism should be incorporated to combine all the good qualities.

We also want to examine the possibility of introducing Lamarckian properties into the paradigm used here. After a fast learning algorithm is applied to individuals in a generation, the resulting weight vector could be encoded back into the chromosomes so that the adapted behavior of the parents could be passed on to their children.

A final remark is that we also tried pure self-adaptation in the early stages of this project, i.e., we let randomized

initial individuals play games among themselves. The result, as expected, was poor. Without suitable challenges from the environment, evolution is impossible because of the lack of learning direction.

References

[1] Carlos M. Fonseca and Peter J. Fleming. Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, 1993.

[2] David E. Goldberg. *Genetic Algorithms in Search, Optimization & Machine Learning.* Addison Wesley, 1989.

[3] Kenneth De Jong and William Spears. On the state of evolutionary computation. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 618–623, 1993.

[4] Ting Kuo and Shu-Yuen Hwang. A genetic algorithm with disruptive selection. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, 1993.

[5] Kai-Fu Lee and Sanjoy Mahajan. A pattern classification approach to evaluation function learning. *Artificial Intelligence*, (36):1–25, 1988.

[6] Kai-Fu Lee and Sanjoy Mahajan. The development of a world class Othello program. *Artificial Intelligence*, (43):21–36, 1990.

[7] Paul S. Rosenbloom. A world-championship-level Othello program. *Artificial Intelligence*, (19):279–320, 1982.

[8] Arthur L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229, 1959.

[9] Patrick Henry Winston. *Artificial Intelligence.* Addison-Wesley, 1992.

[10] Edge C. Yeh, G.K. Ruan, and Y.Y. Hsu. Development of game strategy by genetic algorithm for "five pieces in a line". In *Proceedings of the First National Symposium on Fuzzy Set Theory and Applications*, pages 543–550, 1993. in Chinese.