Pareto Evolution and Co-evolution in Cognitive Game AI Synthesis

Yi Jack Yau, Jason Teo, and Patricia Anthony

School of Engineering and Information Technology
Universiti Malaysia Sabah, Locked Bag No. 2073,
88999 Kota Kinabalu, Sabah, Malaysia
yijackyau@gmail.com,{jtwteo,panthony}@ums.edu.my

Abstract. The Pareto-based Differential Evolution (PDE) algorithm is one of the current state-of-the-art Multi-objective Evolutionary Algorithms (MOEAs). This paper describes a series of experiments using PDE for evolving artificial neural networks (ANNs) that act as game-playing agents. Three systems are compared: (i) a canonical PDE system, (ii) a co-evolving PDE system (PCDE) with 3 different setups, and (iii) a co-evolving PDE system that uses an archive (PCDE-A) with 3 different setups. The aim of this study is to provide insights on the effects of including co-evolutionary techniques on a well-known MOEA by investigating and comparing these 3 different approaches in evolving intelligent agents as both first and second players in a deterministic zero-sum board game. The results indicate that the canonical PDE system outperformed both co-evolutionary PDE systems as it was able to evolve ANN agents with higher quality game-playing performance as both first and second game players. Hence, this study shows that a canonical MOEA without co-evolution is desirable for the synthesis of cognitive game AI agents.

Keywords: Game AI, Co-Evolution, Evolutionary Artificial Neural Networks, Pareto Differential Evolution, Evolutionary Multi-Objective Optimization.

1 Introduction

Artificial intelligence for games (game AI) represents one of the most useful and practical platforms for studying evolutionary computation systems. Game has well defined rules that make them easier to simulate on a computer and its applications to many real world problems in economics, politics, biology, and countless other areas. Zero-sum board games provide a simple yet interesting testing-bed to study both the machine learning and the optimization aspects of soft computing systems. Firstly, these games have perfectly defined sets of rules that limit the possible behaviors of the players, thereby simplifying the problem at hand. Furthermore, games have a clear objective for the players to reach. In addition, games have enough information to allow a wide range of possible behaviors to emerge as represented by the various strategies of the game players.

S. Obayashi et al. (Eds.): EMO 2007, LNCS 4403, pp. 227–241, 2007.

In spite of the additional complexities of co-evolutionary models, they hold some significant advantages that have been exploited within the context of EAs to support the generation of solutions to a series of complex problems. Co-evolutionary techniques have been successfully applied to a number of games, for instance, Awari [1], Pong [2], Nim [3], and Go [4,5]. Evolutionary Programming (EP) has also been used to create ANNs that are capable of playing Tic-Tac-Toe (TTT) [6,7]. Although many single-objective evolutionary techniques have been successfully applied to many different kinds of games, a large number of research issues and questions still remain for multi-objective evolutionary techniques when applied to games.

In previous work, an enhanced version of a hybrid co-evolutionary implementation using the Pareto Differential Evolution (PDE) algorithm known as the Pareto Co-evolutionary Differential Evolution (PCDE) algorithm [8]. This algorithm was reported to be able to automatically synthesize neural network game-playing agents both as the first and second players with reasonable playing strength through the introduction of Pareto multi-objective evolution [8]. In this study, the main objective is to look into the effects of the introduction of the co-evolution technique and whether it is actually beneficial or otherwise to the Pareto evolutionary optimization process. A comprehensive empirical comparison of performance between the previous system of PCDE, a new archived-based version of PCDE called PCDE-A and the canonical PDE without co-evolution is carried out. All of the above implementations do not require an explicit evaluation function for the purpose of automatically generating the game AI for TTT since the scoring from playing against a rule-based player is used as the objective evaluation method during evolution. Finally, the performance of the respective approaches will be measured according to the playing strength of the evolved ANN game-playing agents pitted against three different levels of players (expert, medium and random players).

1.1 Tic-Tac-Toe

TTT is a standard two-player zero-sum game, in which two players alternately put crosses and circles in one of the compartments of a 3 by 3 board. The objective of the game is to get a row of 3 crosses or 3 circles before the opponent does. Player one is the player that moves first, making a cross, followed by player two, making a circle. If at the end of the game both players cannot meet the objective, it means that a draw is awarded to both players. There are 4 player types in TTT. The novice player makes random moves, the intermediate player will block their opponent from winning, the experienced player knows that playing in certain first squares will lose the game, and the expert player will never lose. When both players are at the expert level, the purpose of a TTT first player is to force a win or a draw; however a second player should force a draw by blocking the winning moves of first player. This is because if the first player starts the game with an optimal first move, it will never lose if no mistake is made for following moves, so the second player can only force a tie. The only