Learning Control for an Xpilot-AI Agent Playing Capture the Flag

Gary Parker, Department of Computer Science, Connecticut College, New London, CT, USA
Sarah Penrose, Department of Computer Science, Connecticut College, New London, CT, USA

ABSTRACT

Xpilot-AI is an environment where researchers can test autonomous agent control and learning programs. It is based on Xpilot, which is an open source, 2d space combat game. Xpilot-AI allows a programmer to write scripts that control an agent playing a game of Xpilot. In previous work, a wide range of techniques have been used to develop controllers that are focused on the combat skills for an Xpilot agent. In this research, a Genetic Algorithm (GA) was used to evolve the parameters for an expert agent solving the more challenging problem of capture the flag.

Keywords: Autonomous Agent Learning, Evolutionary Computation, Genetic Algorithm, Video Games, Xpilot-AI

INTRODUCTION

Video games provide a good environment for research in autonomous agent learning. This research is motivated by two goals. One is to increase the entertainment value of the game by creating non-player agents, both cooperative and competitive, that increase the challenge of the game for a human player. The other is to test learning methods to create control programs that can be used in other agents such as autonomous robots. In this research, we use a learning method to develop a control program for a video game agent. Creating competitive non-player agents can be difficult, especially if a diversity of controller or agents that can adapt are desired. One method of learning that can create distinct competitors and has the potential for adapting to different players is Evolutionary Computation (EC). Video games can make viable test beds for researching different methods for evolving control programs for autonomous agents including actual robots. The video game used for this research, Xpilot, is an excellent environment due to its low computational requirements and the Xpilot-AI add-on that allows a user to create an agent to control ships during the game (Parker, 2010).

Many researchers choose to use video games to test various artificial intelligence techniques because they offer a large dynamic environment. These environments make the
learning that is required for an evolved agent to compete seriously in these games much more challenging. The resultant evolved solutions for video game agents are also often more versatile, because they are not hard-coded, and may develop strategies a human programmer would never consider.

Research with learning control for video game agents has a long history. In the past, researchers have evolved agents for the video game Quake3, a 3-dimensional multiplayer game (Priesterjahn, 2006). They used genetic algorithms to evolve an agent that performed better than the bot provided with the game, and then went on to co-evolve opponents to use instead of the provided bot. In other work, the game Pac-Man was used to consider a predator-prey scenario (Yannakakis, 2004). In this case, the connection weights for a neural network were evolved by a GA to serve as the control for the predators' team behavior. Another example is the development of an agent to play a real-time strategy game known as DEF-CON through a combination of artificial intelligence techniques such as simulated annealing, decision tree learning, and case-based reasoning (Baumgarten, 2009).

The goal of the research reported in this paper is to create a competitive agent for Xpilot, a 2D combat-based video game. Many experiments to evolve agents for the Xpilot environment have been completed in the past, although all have concentrated on the development of agents for a combat role. The parameters for a combat expert agent were successfully evolved (Parker, 2007a) using a standard GA. Other artificial intelligence strategies, such as multi-layer neural networks (Parker, 2007b) and reinforcement learning (Allen, 2010), have also been used. These approaches have all been successful strategies to make a better Xpilot combat agent. In most of these works, the agent developed was far superior to those provided by the game.

The research reported in this paper differs from previous work in Xpilot because it explores the capture the flag game mode rather than the traditional combat scenario. The typical combat style of game play in the Xpilot environment has agents that may be controlled by either a computer script or a human player. In the capture the flag mode, which has yet to be researched, agents are still controlled by scripts or humans in a combat environment. However, the goal is no longer to destroy their opponents, but instead to retrieve the opposing team’s flag, a large ball, and drag it to a goal to receive points. The control for this type of play is more difficult, as humans playing the game can easily attest. The ball is massive and difficult to drag by a ship, which has much less mass.

In addition to similarities with learning for other video game agents, this research also has similarities with control and learning systems for RoboCup robots and their simulation. Much research has been done in this area. Li, Wang, and Zell (2007) used a reference point and detailed kinematics information to develop a controller for the soccer playing agents to dribble the ball. Pratomo, et al. (2011) used predetermined scenarios to develop control programs for all modes of play. Research has also been done in the use of evolutionary techniques to learn the control programs for RoboCup agents. Ciesielski and Lai (2001) evolved the weights for a two-layer (a single hidden layer) neural network to control dribble and shoot behavior. They had limited success, but noted that more learning time and altering the fitness function may produce better agents. Luke (1998) used genetic programming to develop a team for the IJCAI / RoboCup97 softbot competition. It was the only team entered that did not use a human-crafted cooperative decision-making control program. Although their team did not win the competition, it played well and beat some of the human-crafted teams. There is a loose correlation between our capture the flag task in Xpilot and robot soccer. The similarity has to do with the robot soccer player going to the ball, dribbling, and shooting. However, our problem is vastly different because of the huge effect the ball has on the Xpilot agent’s movements once it’s attached, which is much of the focal point of this research.
Related Content

Established and Recently Proposed Variants of Particle Swarm Optimization
E. Parsopoulos Konstantinos and N. Vrahatis Michael (2010). *Particle Swarm Optimization and Intelligence: Advances and Applications* (pp. 88-132).
[www.igi-global.com/chapter/established-recently-proposed-variants-particle/40631?camid=4v1a](www.igi-global.com/chapter/established-recently-proposed-variants-particle/40631?camid=4v1a)

Optimized Base Station Sleeping and Renewable Energy Procurement Scheme Using PSO
[www.igi-global.com/article/optimized-base-station-sleeping-and-renewable-energy-procurement-scheme-using-pso/169888?camid=4v1a](www.igi-global.com/article/optimized-base-station-sleeping-and-renewable-energy-procurement-scheme-using-pso/169888?camid=4v1a)
Design Automation, Modeling, Optimization, and Testing of Analog/RF Circuits and Systems by Particle Swarm Optimization
www.igi-global.com/chapter/design-automation-modeling-optimization-testing/72823?camid=4v1a

An Agile Architecture for a Legacy Enterprise IT System
www.igi-global.com/article/an-agile-architecture-for-a-legacy-enterprise-it-system/162709?camid=4v1a