Evolving Artificial Intelligences to Compete in Real-Time Strategy Games

Alice Quintanilla
Southwestern University

Introduction

Evolutionary computation is a technique which aims to create novel and intelligent machine behavior by evolving agents which learn how to succeed in their particular domain with as little human direction as possible. Video games are a common application domain for evolutionary computation because they provide a computer-simulated testing environment and a clearly defined set of goals to judge individuals on. The goal of this project is to evolve an artificial neural network which evaluates game states in MicroRTS, a minimalist distillation of the popular real-time strategy video game genre, designed with AI research in mind.

Challenges for AI in RTS Games

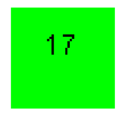
RTS Games have a number of unique qualities that make them especially difficult to evolve artificial intelligences for.

- Players act simultaneously
- Simulating RTS Games is more time and resource intensive than simulating board games and less complicated video games.
- Effective behavior includes both micro-managing individual unis, and controlling groups of units.
- Individual actions have little consequence on the game-state; it's often necessary to perform multiple specific actions in sequence.
- Very large branching factor: the sheer amount of actions available to a player at a given game state is so great that it's impossible to evaluate the favorability of all possible game states.

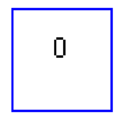
MicroRTS

The goal of MicroRTS is to use your units to attack and destroy all enemy units and buildings while protecting your own. To this end, a player may construct a number of additional units.

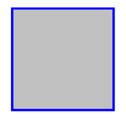
Resource Tile: (not constructible) may be harvested by workers.



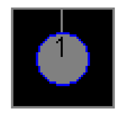
Bases: stockpiles resources and produces workers.



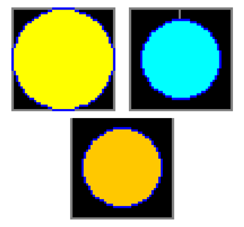
Barracks: may construct various non-worker units.



Worker: Mobile, may harvest resources and attack enemies.



Non-Worker Mobile Units: Mobile, may attack enemies.



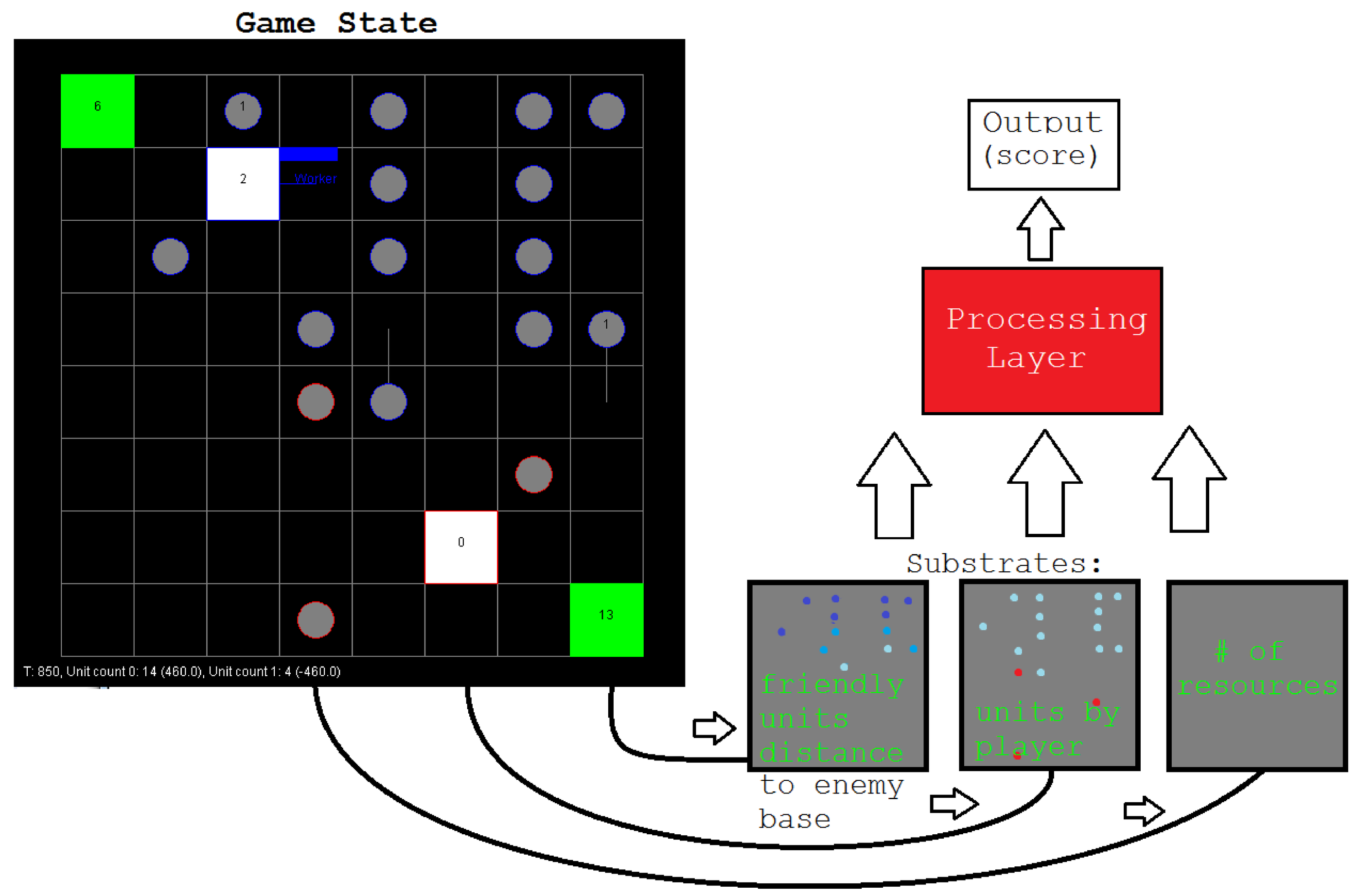
Because MicroRTS was designed specifically as a benchmark for artificial intelligence, there are many advantages to using it over any other RTS:

- It's deterministic, which makes forward-simulation possible
- Doesn't have all the minutia of a real RTS,
- Map is of adjustable size, allowing for scaling difficulty

Evaluating Game States

The AIs evolved in this work are neural networks, which are computational abstractions of human brains. Their job is to evaluate potential future states of the game, and to assign them scores based on how favorable they are to a specific player.

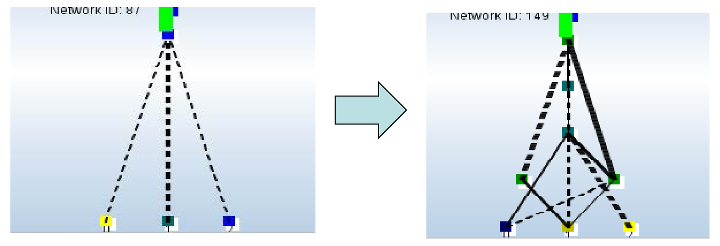
In order for the neural network to be able to make meaningful distinctions between different intermediate game outcomes, each game state must be separated into features that populate neural input substrates, each containing information about different types of specific game elements.



Evolving Neural Networks

In this project, artificial neural networks are generated and evolved using HyperNEAT (Hypercube-based Neuro Evolution of Augmenting Topologies), an approach to solving AI problems inspired by biological evolution:

- The first genotypes are created with Compositional Pattern-Producing Networks (CPPNs)
- A population of 25 neural networks are created from those genotypes



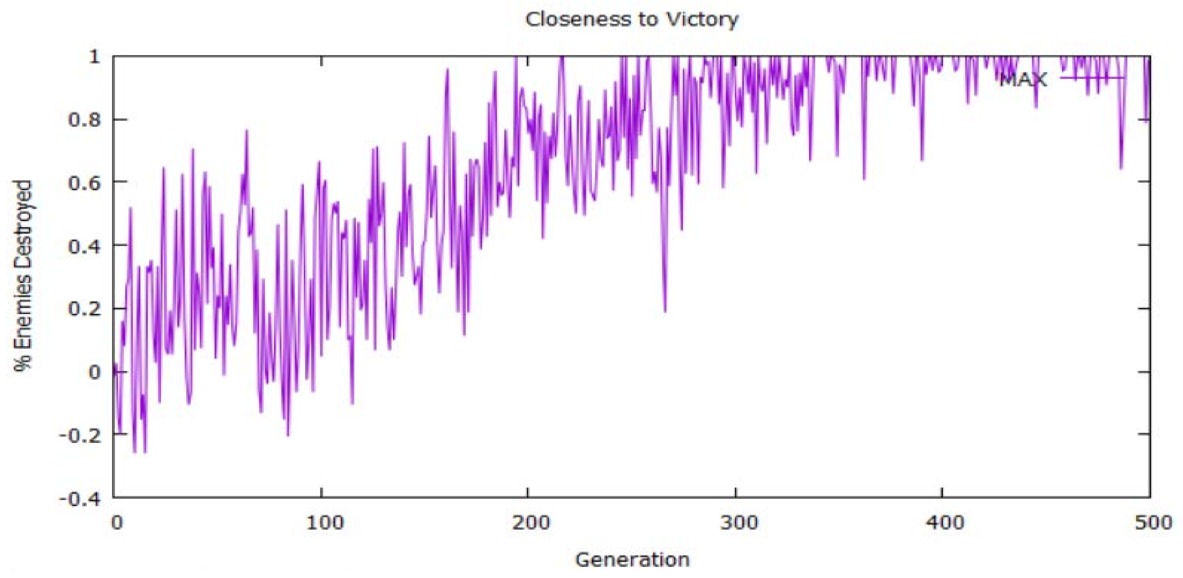
- Each network is tested against a static opponent for 3 games and given average fitness scores.
- After all tests are complete, the genotypes of the most successful networks are bred and mutated to create the next generation, and so on.

Opponent

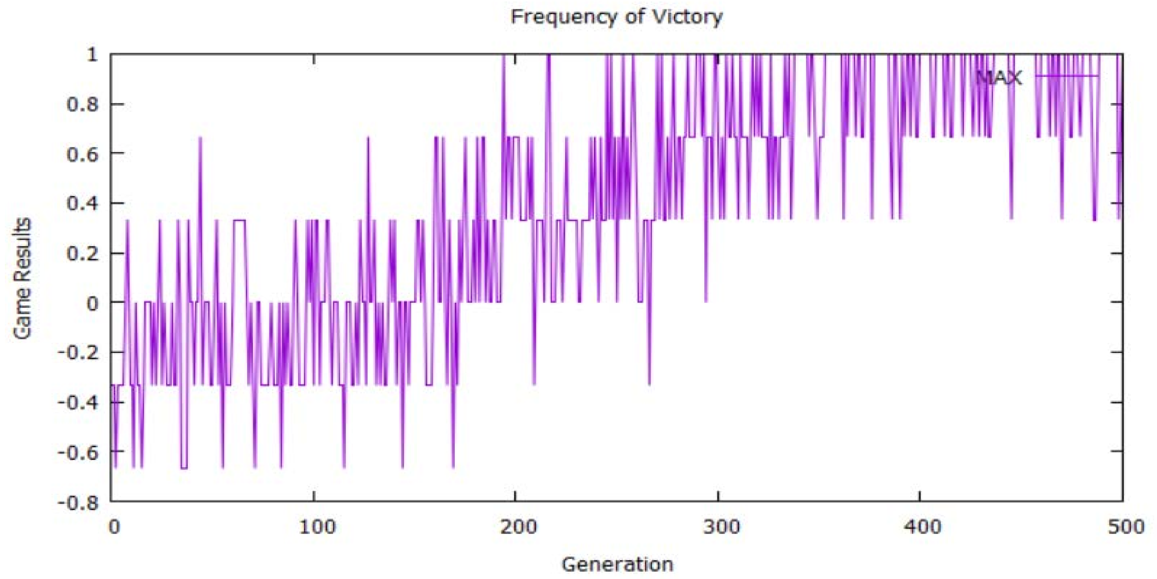
The opponent that the evolving AIs face is static, meaning that it does not evolve or get better over time. As for how this agent makes decisions: it has a human-coded list of actions that it may take, all of which are categorized as either aggressive or passive. Every game state, this agent has a 5/6 chance of acting aggressively, and a 1/6 chance of acting passively. This random component makes it so that the opponent doesn't always perform at the exact same level.

Results

These graphs show the average performance of the highest scoring individuals from each generation.



1 : victory
0 : draw
-1 : loss



References

[1] Cameron Browne et al. 2012. A Survey of Monte Carlo Tree Search Methods IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES 4, 1 (March. 2012),

[2] Gai, Krishnamakari, and Jain. 2010. Learning Multiuser Channel Allocations in Cognitive Radio Networks: A Combinatorial Multi-Armed Bandit Formulation

[3] Santiago Ontañón. 2017. Artificial Intelligence. Retrieved June 13, 2017 from https://github.com/santiontanon/microrts/wiki/Artificial-Intelligence

[4]Santiago Ontañón. 2012. Experiments with Game Tree Search in Real-Time Strategy Games: arXiv:1208.1940 [cs.AI]