

# COMP SCI 5401 FS2017 Assignment 2c

Stuart Miller  
[sm67c@mst.edu](mailto:sm67c@mst.edu)

December 2, 2017

## 1 Overview

Assignment 2c provides an improvement to the Iterative Prisoner's Dilemma (IPD) problem by employing a coevolutionary Genetic Programming Search across single agents evaluated by a tit-for-tat strategy search.

## 2 Methodology

As before, the algorithm works by generating a population of random trees (50% full depth trees, 50% grown tress), then iteratively generating children based on tree crossover from parents. Children are occasionally mutated, then simulated by playing tit-for-tat to obtain a fitness rating and added to the population. To obtain this fitness rating, each agents is plays a set number of rounds on a tit-for-tat strategy (the agent's opponent chooses what the agent itself did last). An option is also present to score each agent's fitness across multiple runs of IPD, with multiple randomized initial memories. Doing so allows for a more accurate fitness rating. After each generation, the population is thinned by removing poor agents chosen in a k-tournament.

The main feature of this solution involves computing not only an absolute fitness for each agent, but also a composite coevolutionary fitness. This fitness is calculated after each generation is established (i.e. after all children have had their fitness independently evaluated). This composite fitness (referred to as "comp\_fitness" or variables prefixed with "comp" in code) is obtained by choosing a random sample of opponents from the population. Opponents are chosen without replacement. The absolute fitnesses of all chosen opponents and the agent itself are averaged and assigned to the agent in question as its composite fitness.

## 3 Experimental Setup

The following default configuration values were chosen (Figure 1). Some algorithm parameters were dictated in the assignment (d, k, l, seedType, and evals). Termination is set to merely number of evals since this assignment dictated a 10,000 eval minimum. A mu and lambda of 100 and 50 were chosen. This allowed for rather sizable populations, but still a large number of generations before 10,000 evals. A low mutation rate was chosen as GP typically relies more heavily on recombination. Survival selection was set to k-Tournaments with a k of 40. This allowed for sufficient exploration. As noted in Section 2, there is an option to re-randomize an agents memory across multiple runs of IPD for more well-rounded fitness. This has been set to 5 here. Finally, there is a small parsimony coefficient and plus survival strategy. These were simply inherited from the last assignment, since they worked well there.

coevSamp	= 10
d	= 10
evals	= 10000
k	= 5
l	= 30
lambda	= 50
logfile	= ./log/default.txt
mu	= 100
mutation	= 0.05
p	= 0.01
parentSelection	= 1
rerandmem	= 5
runs	= 3
seed	= 689318354
seedType	= 1
solutionfile	= ./sln/default.txt
survivalK	= 40
survivalSelection	= 1
survivalStrat	= 1
termTest	= 0
unchanged	= 200

Figure 1: Configuration Values

The variation chosen for this assignment was the mutation operator. I thought it interesting that Koza states that small mutation sizes are preferred for GP, so an examination of this seems appropriate. The three values chosen were 5%, 0%, and 80%. 5% was the chosen default and seemed like an adequate approximation of Koza's recommended minimal value. 80% was a sufficiently high number, and 0% was chosen to see the effect of no mutation at all.

## 4 Results

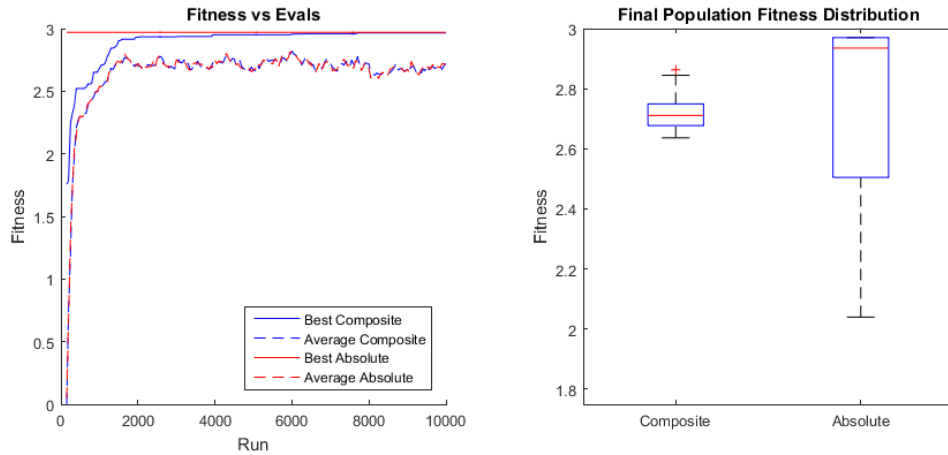


Figure 2: Default Mutation (5%)

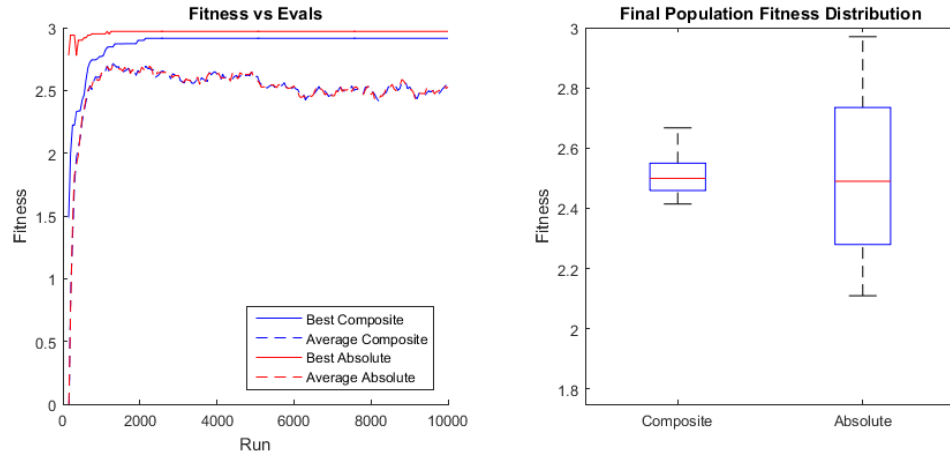


Figure 3: High Mutation (80%)

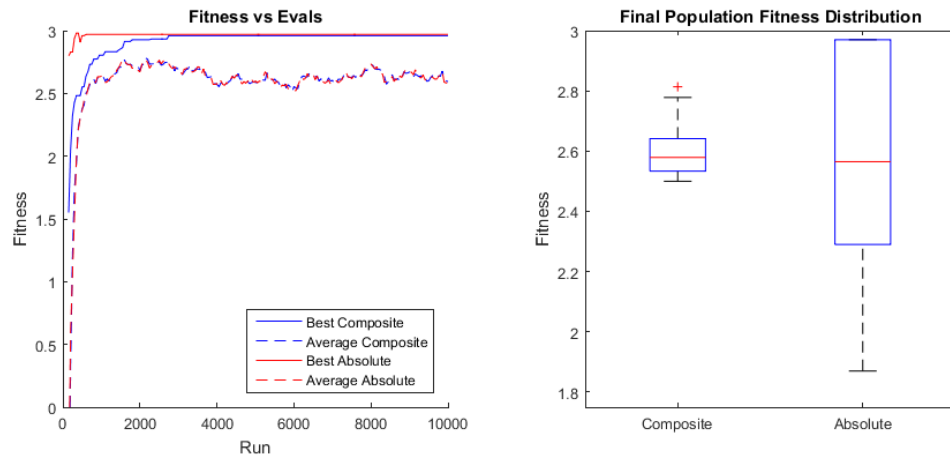


Figure 4: No Mutation (0%)

## 5 Discussion

## 6 Conclusion