COMP SCI 5401 FS2017 Assignment 2c

Stuart Miller
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.

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 (this will bee experimentally evaluated later in Section 4. 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

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

parsimony coefficient and plus survival strategy. These were simply inherited from the last assignment, since they worked well there.

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.
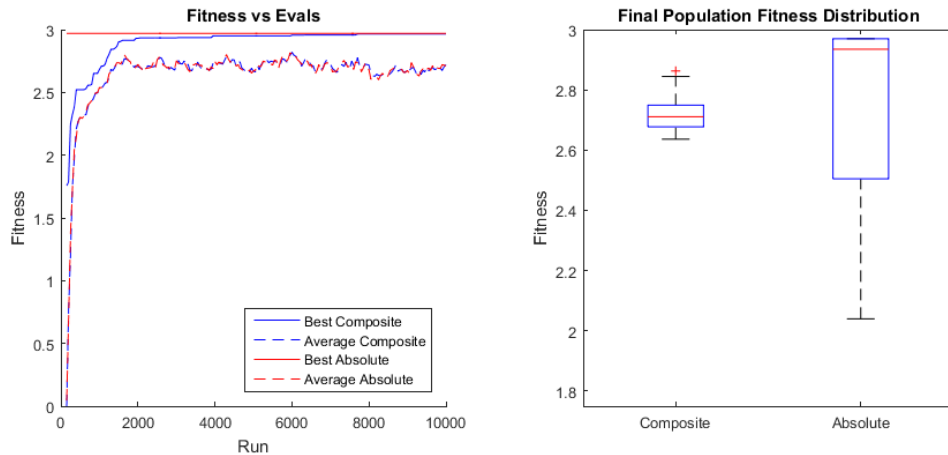
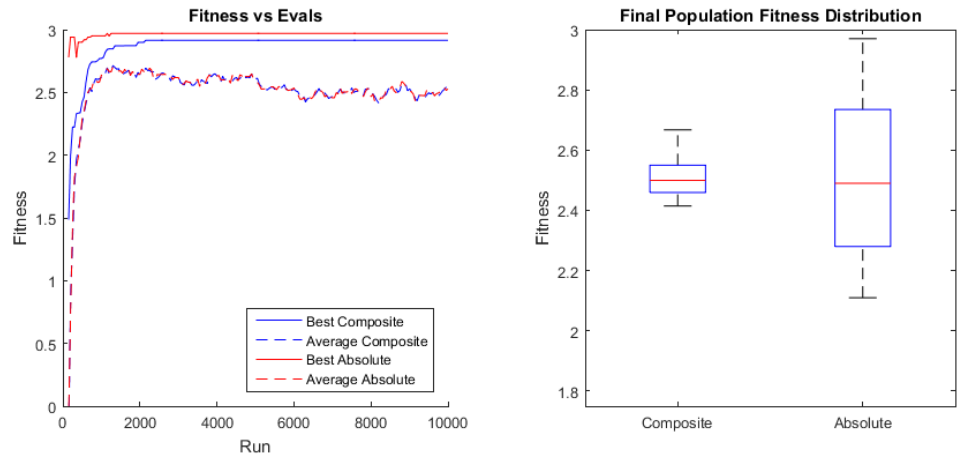## 4   Results

Figure 2: Default Mutation (5%)
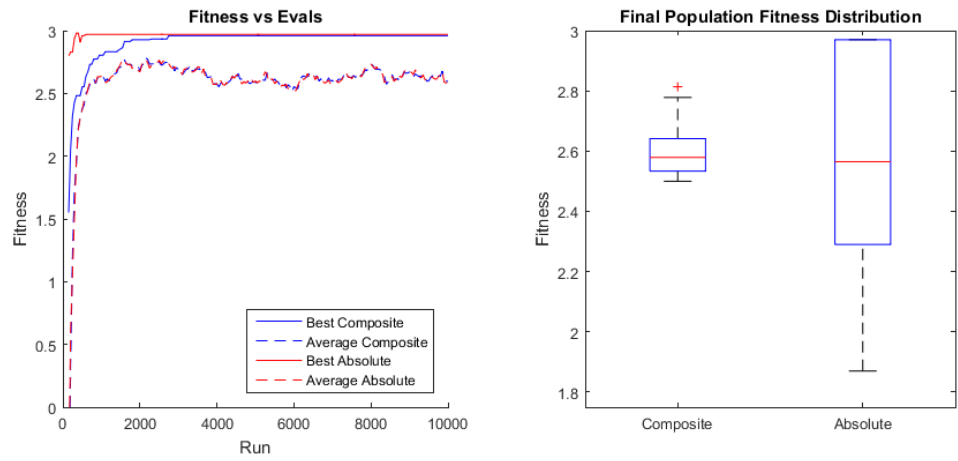
Figure 3: High Mutation (80%)

Figure 4: No Mutation (0%)

	Default		No Mutation		High Mutation	
	Composite	Absolute	Composite	Absolute	Composite	Absolute
Median	2.711	2.935	2.579	2.565	2.500	2.490
Average	2.720	2.710	2.592	2.590	2.505	2.528
Std. Deviation	0.055	0.330	0.068	0.342	0.059	0.270

Figure 5: Table of Averages

Default vs. High Mutation Composite

F-Test Two-Sample for Variances

	Variable 1	Variable 2
Mean	2.7197732	2.5054545
Variance	0.003034692	0.003569289
Observations	100	100
df	99	99
F	0.850223024	
P(F<=f) one-tail	0.210484683	
F Critical one-tail	0.717328593	

t-Test: Two-Sample Assuming Unequal Variances

	Variable 1	Variable 2
Mean	2.7197732	2.5054545
Variance	0.003034692	0.003569289
Observations	100	100
Hypothesized Mean Difference	0	
df	197	
t Stat	26.37285678	
P(T<=t) one-tail	7.50507E-67	
t Critical one-tail	1.652625219	
P(T<=t) two-tail	1.50101E-66	
t Critical two-tail	1.972079034	

Figure 6: Default vs. High Mutation Composite Fitness

Default vs. High Mutation Absolute

F-Test Two-Sample for Variances

	Variable 1	Variable 2
Mean	2.7096	2.5281
Variance	0.109777616	0.073793323
Observations	100	100
df	99	99
F	1.487636162	
P(F<=f) one-tail	0.024730335	
F Critical one-tail	1.394061257	

t-Test: Two-Sample Assuming Unequal Variances

	Variable 1	Variable 2
Mean	2.7096	2.5281
Variance	0.109777616	0.073793323
Observations	100	100
Hypothesized Mean Difference	0	
df	191	
t Stat	4.236182524	
P(T<=t) one-tail	1.76653E-05	
t Critical one-tail	1.652870547	
P(T<=t) two-tail	3.53306E-05	
t Critical two-tail	1.97246199	

Figure 7: Default vs. High Mutation Absolute Fitness

Default vs. No Mutation Composite

F-Test Two-Sample for Variances

	Variable 1	Variable 2
Mean	2.7197732	2.5920365
Variance	0.003034692	0.004688387
Observations	100	100
df	99	99
F	0.647278393	
P(F<=f) one-tail	0.0157722	
F Critical one-tail	0.717328593	

t-Test: Two-Sample Assuming Equal Variances

	Variable 1	Variable 2
Mean	2.7197732	2.5920365
Variance	0.003034692	0.004688387
Observations	100	100
Pooled Variance	0.003861539	
Hypothesized Mean Difference	0	
df	198	
t Stat	14.53518173	
P(T<=t) one-tail	2.36764E-33	
t Critical one-tail	1.652585784	
P(T<=t) two-tail	4.73529E-33	
t Critical two-tail	1.972017478	

Figure 8: Default vs. No Mutation Composite Fitness

Default vs. No Mutation Absolute

F-Test Two-Sample for Variances

	Variable 1	Variable 2
Mean	2.7096	2.5896
Variance	0.109777616	0.118476606
Observations	100	100
df	99	99
F	0.926576307	
P(F<=f) one-tail	0.352573944	
F Critical one-tail	0.717328593	

t-Test: Two-Sample Assuming Unequal Variances

	Variable 1	Variable 2
Mean	2.7096	2.5896
Variance	0.109777616	0.118476606
Observations	100	100
Hypothesized Mean Difference	0	
df	198	
t Stat	2.511723542	
P(T<=t) one-tail	0.006406627	
t Critical one-tail	1.652585784	
P(T<=t) two-tail	0.012813254	
t Critical two-tail	1.972017478	

Figure 9: Default vs. No Mutation Absolute Fitness

High Mutation vs. No Mutation Composite

F-Test Two-Sample for Variances

	Variable 1	Variable 2
Mean	2.5054545	2.5920365
Variance	0.003569289	0.004688387
Observations	100	100
df	99	99
F	0.76130424	
P(F<=f) one-tail	0.088304317	
F Critical one-tail	0.717328593	

t-Test: Two-Sample Assuming Equal Variances

	Variable 1	Variable 2
Mean	2.5054545	2.5920365
Variance	0.003569289	0.004688387
Observations	100	100
Pooled Variance	0.004128838	
Hypothesized Mean Difference	0	
df	198	
t Stat	-9.527933075	
P(T<=t) one-tail	2.9661E-18	
t Critical one-tail	1.652585784	
P(T<=t) two-tail	5.9322E-18	
t Critical two-tail	1.972017478	

Figure 10: High Mutation vs. No Mutation Composite Fitness

High Mutation vs. No Mutation Absolute

F-Test Two-Sample for Variances

	Variable 1	Variable 2
Mean	2.5281	2.5896
Variance	0.073793323	0.118476606
Observations	100	100
df	99	99
F	0.622851428	
P(F<=f) one-tail	0.009677494	
F Critical one-tail	0.717328593	

t-Test: Two-Sample Assuming Unequal Variances

	Variable 1	Variable 2
Mean	2.5281	2.5896
Variance	0.073793323	0.118476606
Observations	100	100
Hypothesized Mean Difference	0	
df	188	
t Stat	-1.402553488	
P(T<=t) one-tail	0.081199597	
t Critical one-tail	1.652999113	
P(T<=t) two-tail	0.162399194	
t Critical two-tail	1.972662692	

Figure 11: High Mutation vs. No Mutation Absolute Fitness

5   Discussion

5.1   Graphical Analysis

The graphs in Figures 2 - 4 show the best fitnesses in the population over time, split up by composite and absolute fitnesses. It is interesting to note that the average for both fitnesses stays mostly the same whereas the absolute best is always higher. This makes sense as the composite consists of score that is always adjusted lower for its fellow population members.

As for the value of mutation, notice high mutation graph in particular. Its average value actually goes down over time, showing that the excessive exploration is actually hurting it in this instance. Alternatively the fitness vs. evals graphs for no mutation and default mutation are nearly identical.

The really drastic differences become evident in the box plot of the final population. Notice that for default mutation, the red median line is right up near the top, whereas for the high and no mutation, it is in the middle. This is true for both composite and absolute fitnesses, although it is much more pronounced for absolute. (See the table in Figure 5 for the exact numerical values.) In all cases, the default mutation value scored the best fitness (comparing the median and average values of the composite and absolute fitness). This confirms that the default value of a small 5% mutation rate is preferable.

5.2   Statistical Analysis

The statistical analysis can be seen in Figures 6 - 11. In all cases except for absolute fitness High Mutation vs. No Mutation, the null hypothesis can be safely rejected, concluding that there is not a statistically proven advantage to using high mutation over no mutation (or vice versa). This is acceptable though, as the main point that the default mutation of 5% is better still stands true.

6   Conclusion

It is clear from the analysis in Section 5.1 that the default mutation rate of 5% is most beneficial. This follows with the advice given by Koza and stated in class lectures. It is interesting to note, however, that using no mutation at all is not necessarily better or worse than using too much mutation; each suffer from roughly equivalent decreased performance.

Additionally, none of the parameters ever allowed the algorithm to attain anything above a 3 payoff. It was relatively easy for the algorithm to settle on a solution where both agents cooperate, but impossible to find an appropriate time to defect. Perhaps this is the most beneficial solution, as, in real-life prisoners dilemna situation, it is much preferable to take the mutually beneficial payout rather than to go for the extreme scenario and risk complete loss.