COMP SCI 5401 FS2017 Assignment 2c

Dalton Cole
drcgy5@mst.edu

December 3, 2017

Methodology

For this experiment, a coevolutionary search of the Iterated Prisoners Dilemma was performed. Table 1 shows the configurable parameters for the experiment. For the statistical analysis shown later in this report, Table 2 shows the configuration parameters used.

The following sub-sections will describe some of the underlining structures of the coevolutionary algorithm used.

Initial Starting Population

The fitness for the initial starting population was assigned via the tic-for-tat method. This only influenced in the parent selection portion of first evaluation cycle. The purpose of this is because the underlining goal is to create an evolutionary algorithm which is strong against the tic-for-tat method. Having the initial population be based on that gives the tic-for-tat initial goal.

Move Queue

Every Prisoner initially starts out with the exact same move queue each time it is evaluated. This is done so each strategy is on an even playing field, meaning, no strategy has an easier or harder move sequence to evolve from.

Tree Generation

There is a fifty-fifty percent chance that full method or grow method for ramped half-and-half will be chosen. When grow method is chosen, each node has a $((MaxDepth - CurrentDepth)/MaxDepth) * 100$ percent chance of growing into a branch. Each operator and memory location has an equal chance of being chosen.

Sub-Tree Crossover

For recombination, sub-tree crossover was used. A node of equal depth on each tree was chosen to switch between one another. An equal depth node was chosen as to not go over the depth limit.

Experimental Setup

Table 2 shows the configurations used for this experiment. One hundred was used as the population size so the experiment would run in a timely manner. Fifty children was chosen as a happy balance between exploitation and exploration. Fitness Proportional Selection was selected because it gave a slight speed boost when compared to K-Tournament without Replacement. The Plus Survival Selection Strategy was selected so past generations could more directly influence current generations in the creation of a more optimal solution. A Coevolutionary Fitness Sampling value of 10 was chosen because it lead to more generations than a larger value, but comparing strategies to one another is still seen as important. Parsimony Pressure

Table 1: Configurable Parameters

Parameter	**Options**
Random Seed	Any Integer
Number of Iterations to Play	Any Integer
Number of Iterations to Keep in Memory	Any Integer
Maximum Tree Depth	Any Integer
Number of Runs	Any Integer
Number of Fitness Evaluations	Any Integer
Population Size	Any Integer
Number of Children Per Generation	Any Integer
Parent Selection Strategy	Fitness Proportional Selection Over Selection
Survival Selection	Truncation K-Tournament without Replacement
Parsimony Pressure Penalty Coefficient	Any Float
Termination Convergence Criterion	Any Integer
Survival Selection Strategy	Plus Comma
Coevolutionary Fitness Sampling	Any Integer between 1 and Population Size + Children Count - 1
Detect Cycling	true or false
Deter Cycling	true or false

Table 2: Configurable Parameters

Parameter	**Options**
Random Seed	0, 1, 2
Number of Iterations to Play	30
Number of Iterations to Keep in Memory	5
Maximum Tree Depth	10
Number of Runs	30
Number of Fitness Evaluations	10,000
Population Size	100
Number of Children Per Generation	50
Parent Selection Strategy	Fitness Proportional Selection
Survival Selection	Truncation
Parsimony Pressure Penalty Coefficient	0.0, 0.5, 1.0
Termination Convergence Criterion	10,000
Survival Selection Strategy	Plus
Coevolutionary Fitness Sampling	10
Detect Cycling	false
Deter Cycling	false

Table 3: Averages and Standard Deviations for Each Strategy

	Mean	Standard Deviation
Relative Fitness 0.0	1.7305289	0.314925982
Relative Fitness 0.5	3.120685967	0.413506412
Relative Fitness 1.0	3.280441267	0.353049552
Absolute Fitness 0.0	3.4	0
Absolute Fitness 0.5	3.4	0
Absolute Fitness 1.0	4.2	0

was selected to test how much it effected the dependent variable (absolute fitness against Tic-For-Tat). Parsimony Pressure took on three different values over the course of the experiment: 0.0, 0.5, and 1.0.

Results

The results of the experiment showed that there was no significant difference between using a higher Parsimony Pressure Penalty Coefficient when compared to a lower one or none at all. Figures 7, 8, 9, 10, and 11 all come to this conclusion. It is worth noting, as seen in Table 3, that the mean fitness increases as parsimony pressure goes up, just not at a significant level.

Figures 1, 2, and 3 show the relative fitness plots for the three trials. Figures 4, 5, and 6 show the absolute fitness plots for the three trials.

Discussion

Figures 1, 2, and 3 show the relative progression of fitness value between coevolutionary algorithms. With no parsimony pressure, Figure 1 slowly loses fitness over time. This is likely because each Prisoner is adapting to the other prisoners, thus decreasing the fitness between one another. As can be seen in Figure 4, this has no effect on the absolute fitness. Unlike 0.0 parsimony pressure, Figures 2 and 3 show a general increase in relative fitness as time goes on. However, Figures 5 and 6 show a slow decent in the absolute fitness of the two strategies. This may be because some strategies are getting better at playing with one another, but not at the overall goal of beating tic-for-tat. One possible solution to this would be to re-introduce fitness based on tic-for-tat every x generations to weed out individuals that are solely evolving to fight other evolved Prisoners.

Conclusion

From this experiment, it has been found that although a higher parsimony pressure coefficient leads to a higher mean fitness, there is no significant difference between using no parsimony pressure and using a medium or high coefficient. In the future it is worth looking at what effects on fitness re-introducing assigning fitnesses based on tic-for-tat every x generations will have on the end fitness.

Bonus 1

To detect cycling, a dequeue is used. Each time after survival selection is completed, the best Prisoner is added to the end of the dequeue. If this Prisoner already exists in the dequeue, but is not already at the end, then we know that cycling has occurred, as in, a Prisoner who was previously the best, but was dethroned, is now the best again. An investigation of the effects of cycling is talked about in Section *Bonus 2*.

Figure 1: Relative Fitness For Parsimony Pressure Penalty Coefficient = 0.0

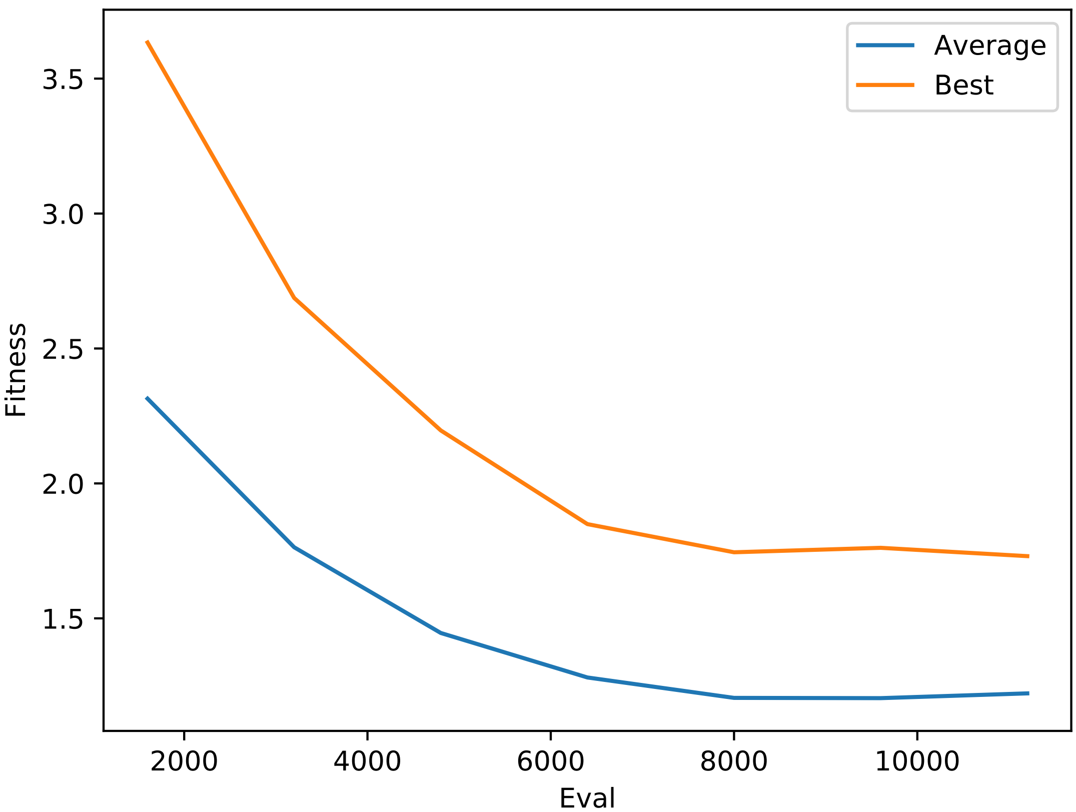

Figure 2: Relative Fitness For Parsimony Pressure Penalty Coefficient = 0.5

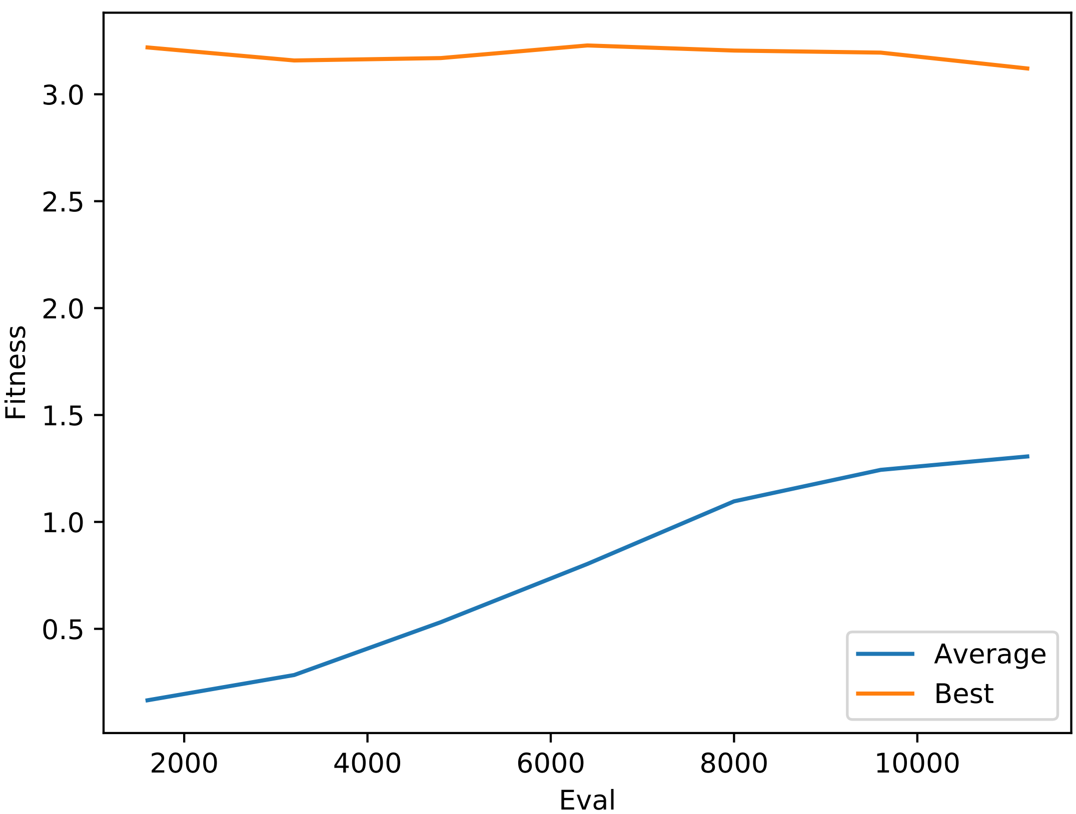

Figure 3: Relative Fitness For Parsimony Pressure Penalty Coefficient = 1.0

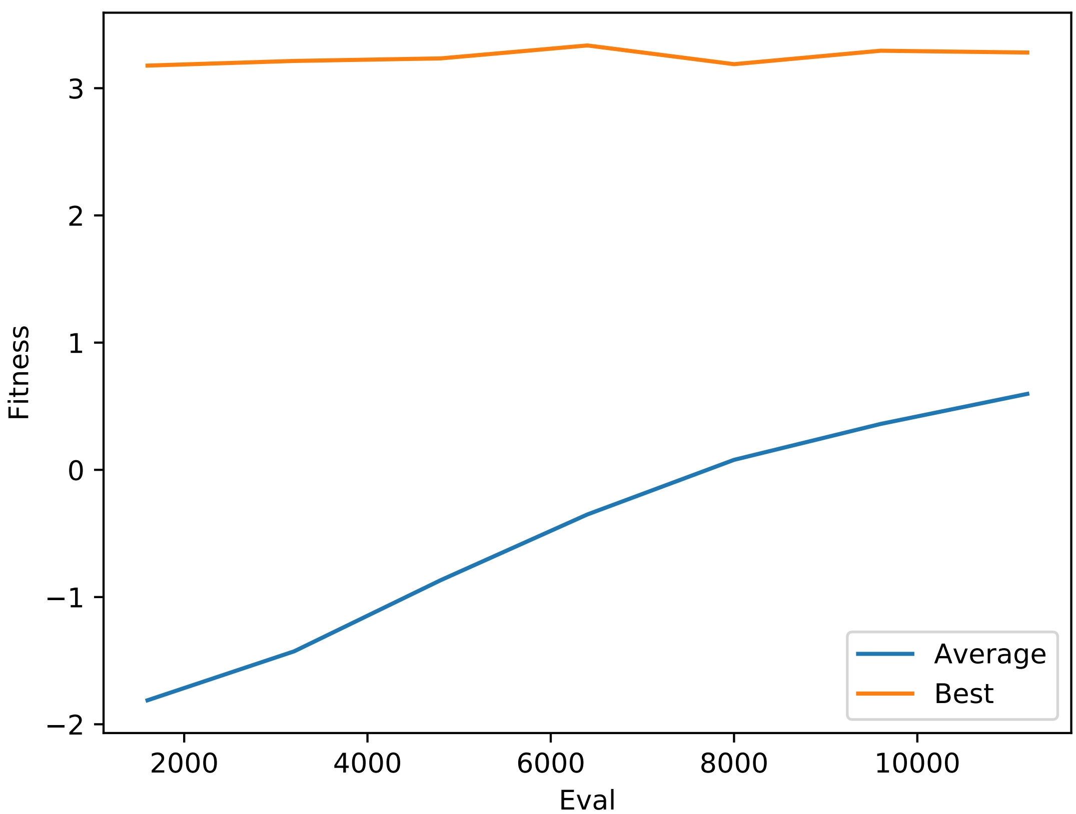

Figure 4: Absolute Fitness For Parsimony Pressure Penalty Coefficient = 0.0

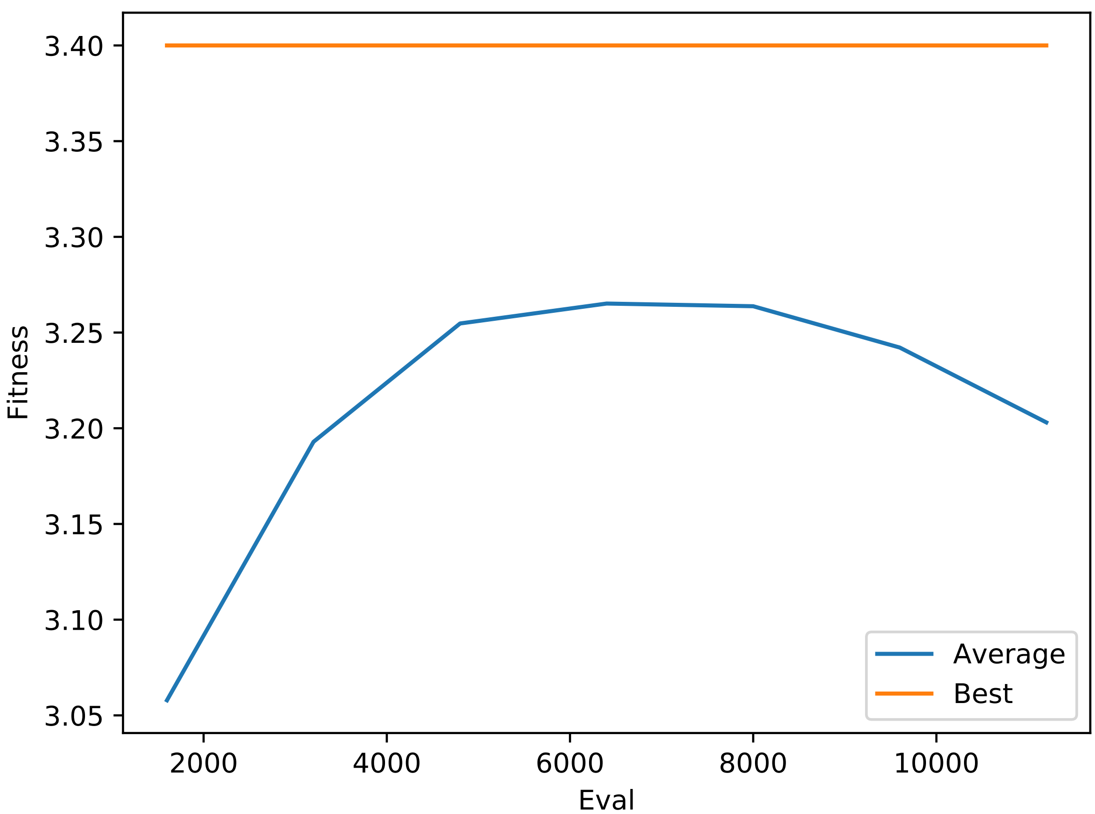

Figure 5: Absolute Fitness For Parsimony Pressure Penalty Coefficient = 0.5

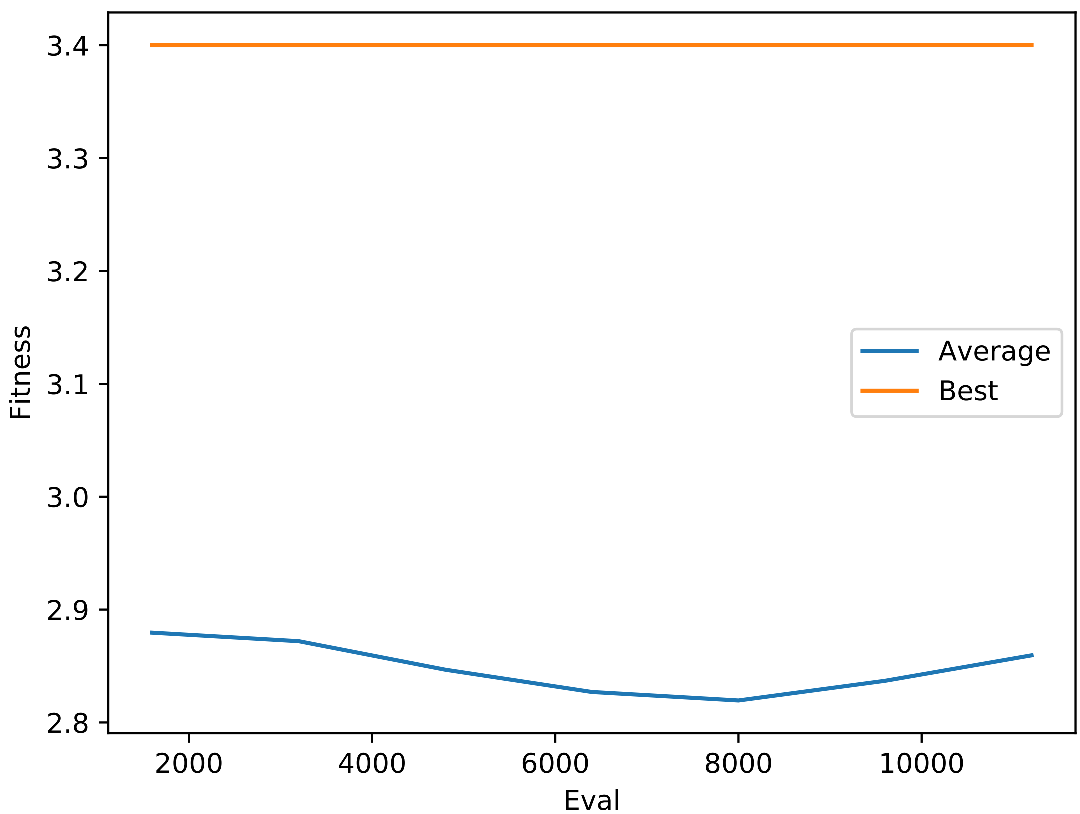

Figure 6: Absolute Fitness For Parsimony Pressure Penalty Coefficient = 1.0

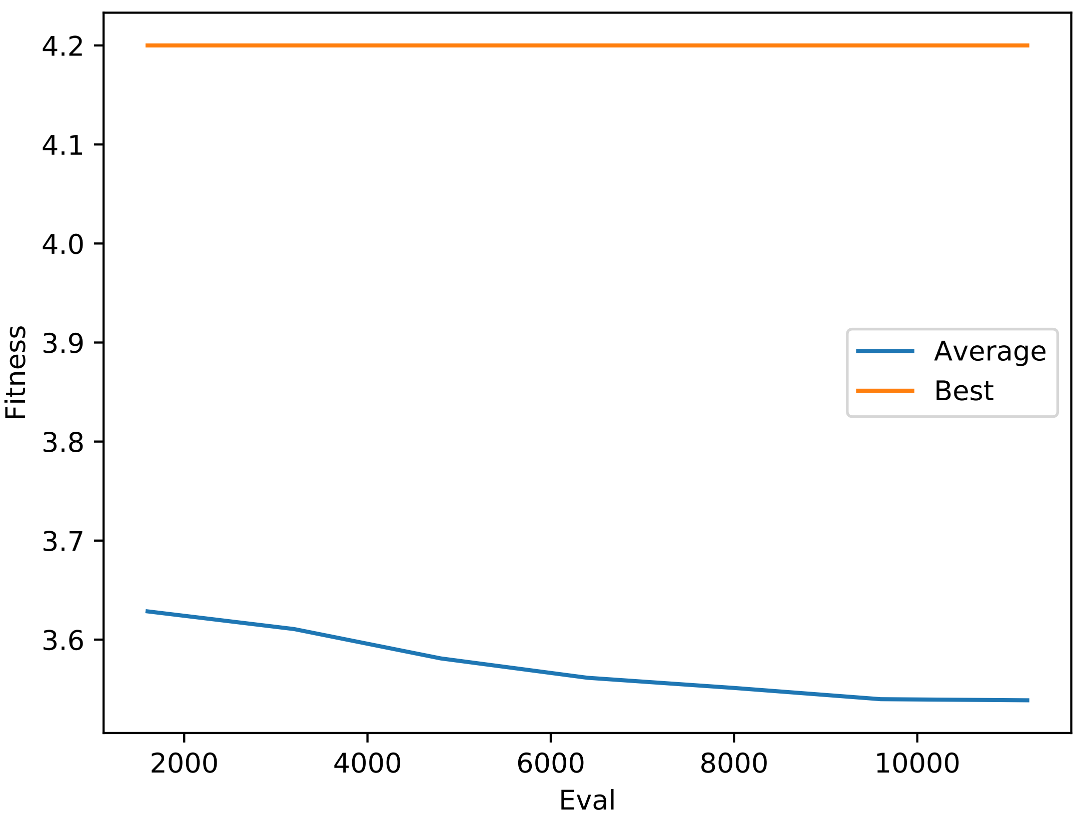

Figure 7: Statistical Analysis for Relative Parsimony Pressure Penalty Coefficient 0.0 VS 0.5

Parsimony_Pressure_Penalty_Coefficent	0	0.5
F-Test Two-Sample for Variances		

	Variable 1	*Variable 2*
Mean	1.730529	3.120686
Variance	0.099178	0.170988
Observations	30	30
df	29	29
F	0.580033	
P(F<=f) one-tail	0.074198	
F Critical one-tail	0.5374	

M(1) < M(2) ^ F > F-Critical => Equal Variances

t-Test: Two-Sample Assuming Equal Variances

	Variable 1	*Variable 2*
Mean	1.730529	3.120686
Variance	0.099178	0.170988
Observations	30	30
Pooled Variance	0.135083	
Hypothesized Mean Difference	0	
df	58	
t Stat	-14.649	
P(T<=t) one-tail	1.89E-21	
t Critical one-tail	1.671553	
P(T<=t) two-tail	3.78E-21	
t Critical two-tail	2.001717	

t stat < t Critical => Same

Figure 8: Statistical Analysis for Relative Parsimony Pressure Penalty Coefficient 0.0 VS 1.0

Parsimony_Pressure_Penalty_Coefficent	0	1

F-Test Two-Sample for Variances

	Variable 1	*Variable 2*
Mean	1.730529	3.280441
Variance	0.099178	0.124644
Observations	30	30
df	29	29
F	0.795693	
P(F<=f) one-tail	0.27112	
F Critical one-tail	0.5374	

M(1) < M(2) ^ F > F-Critical => Equal Variances

t-Test: Two-Sample Assuming Equal Variances

	Variable 1	*Variable 2*
Mean	1.730529	3.280441
Variance	0.099178	0.124644
Observations	30	30
Pooled Variance	0.111911	
Hypothesized Mean Difference	0	
df	58	
t Stat	-17.9439	
P(T<=t) one-tail	1.2E-25	
t Critical one-tail	1.671553	
P(T<=t) two-tail	2.39E-25	
t Critical two-tail	2.001717	

t stat < t Critical => Same

Figure 9: Statistical Analysis for Relative Parsimony Pressure Penalty Coefficient 0.5 VS 1.0

Parsimony_Pressure_Penalty_Coefficent	0.5	1

F-Test Two-Sample for Variances

	Variable 1	*Variable 2*
Mean	3.120686	3.280441
Variance	0.170988	0.124644
Observations	30	30
df	29	29
F	1.371807	
P(F<=f) one-tail	0.199847	
F Critical one-tail	1.860811	

M(1) < M(2) ^ F < F-Critical => Unequal Variances

t-Test: Two-Sample Assuming Equal Variances

t-Test: Two-Sample Assuming Unequal Variances

	Variable 1	*Variable 2*
Mean	3.120686	3.280441
Variance	0.170988	0.124644
Observations	30	30
Hypothesized Mean Difference	0	
df	57	
t Stat	-1.60931	
P(T<=t) one-tail	0.056537	
t Critical one-tail	1.672029	
P(T<=t) two-tail	0.113073	
t Critical two-tail	2.002465	

t stat < t Critical => Same

Figure 10: Statistical Analysis for Absolute Parsimony Pressure Penalty Coefficient 0.0 VS 0.5 (NOTE: 0.0 and 0.5 have the same absolute fitness)

Parsimony_Pressure_Penalty_Coefficent 0 0.5

F-Test Two-Sample for Variances

	Variable 1	*Variable 2*
Mean	3.4	3.4
Variance	1.84E-30	1.84E-30
Observations	30	30
df	29	29
F	1	
P(F<=f) one-tail	0.5	
F Critical one-tail	0.5374	

M(1) = M(2) ^ F > F-Critical =>Same

Figure 11: Statistical Analysis for Absolute Parsimony Pressure Penalty Coefficient 0.0 and 0.5 VS 1.0 (NOTE: 0.0 and 0.5 have the same absolute fitness)

Parsimony_Pressure_Penalty_Coefficent	0	1

F-Test Two-Sample for Variances

	Variable 1	*Variable 2*
Mean	3.4	4.2
Variance	1.84E-30	3.26E-30
Observations	30	30
df	29	29
F	0.5625	
P(F<=f) one-tail	0.063553	
F Critical one-tail	0.5374	

M(1) < M(2) ^ F > F-Critical => Equal Variance

t-Test: Two-Sample Assuming Equal Variances

	Variable 1	*Variable 2*
Mean	3.4	4.2
Variance	1.84E-30	3.26E-30
Observations	30	30
Pooled Variance	2.55E-30	
Hypothesized Mean Difference	0	
df	58	
t Stat	-1.9E+15	
P(T<=t) one-tail	0	
t Critical one-tail	1.671553	
P(T<=t) two-tail	0	
t Critical two-tail	2.001717	

t stat < t Critical => Same

Figure 12: Relative Fitness For Bonus 1 and 2

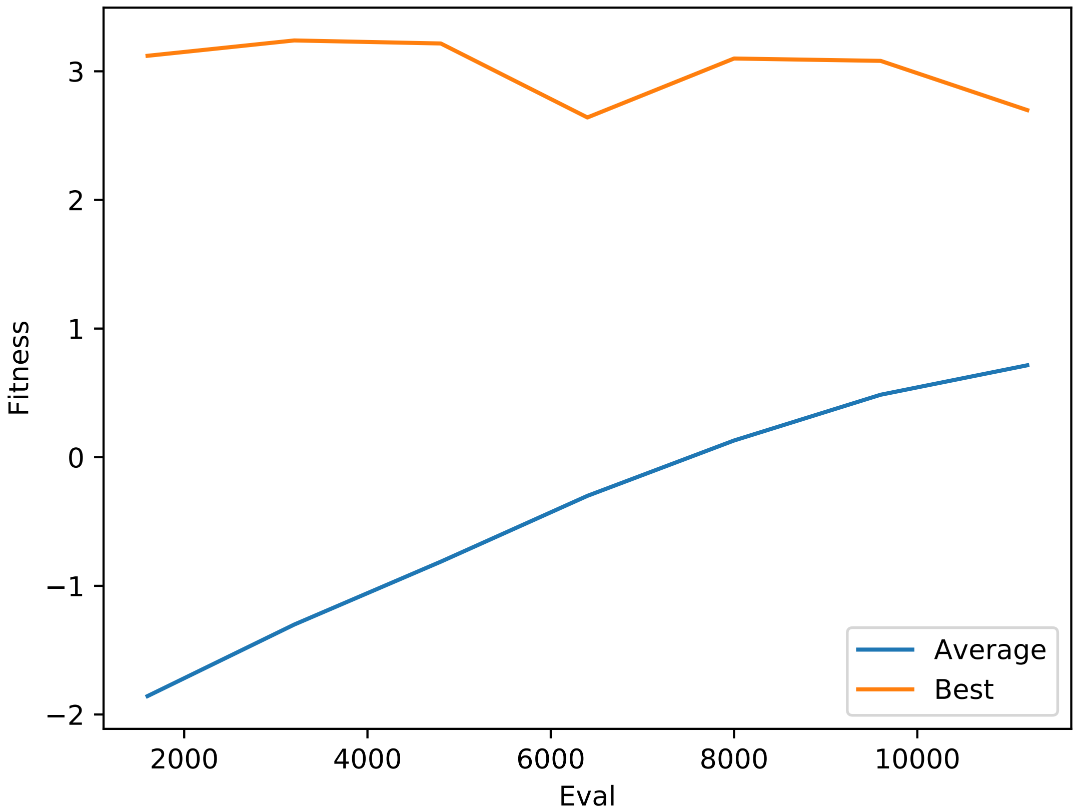

Table 4: Averages and Standard Deviations for Bonus

	Mean	Standard Deviation
Relative Fitness 1.0	2.698237067	1.271516005
Absolute Fitness 1.0	5.0	0

Bonus 2

To deter cycling, when a cycle is found, the best element is removed and replaced by a randomly generated Prisoner. This is to add diversity into the population which should prevent cycling in the future.

In 30 runs of detecting and deterring cycling, 23 cycles were formed and deterred. As can be seen in Figure 12, average fitness slowly increases, however, best fitness slowly decreases. As with the main experiment, this might be due to the Prisoners getting better at playing with one another. Figure 13 shows that the absolute fitness is maximized initially and stays maximized for the rest of the run, average fitness slowly decreases however.

Figure 14 shows the relative statistics comparing trial 3 (parsimony coefficient = 1.0) to the bonus trial. It is shown that the normal trial without cycling detection is significantly better. This is surprising since with deterring cycling, the strategy can have a perfect run against tic-for-tat. Figure 15 shows that there is no significant difference between the two trials. Table 4 shows the mean and standard deviation for the bonus trial.

Figure 13: Absolute Fitness For Bonus 1 and 2

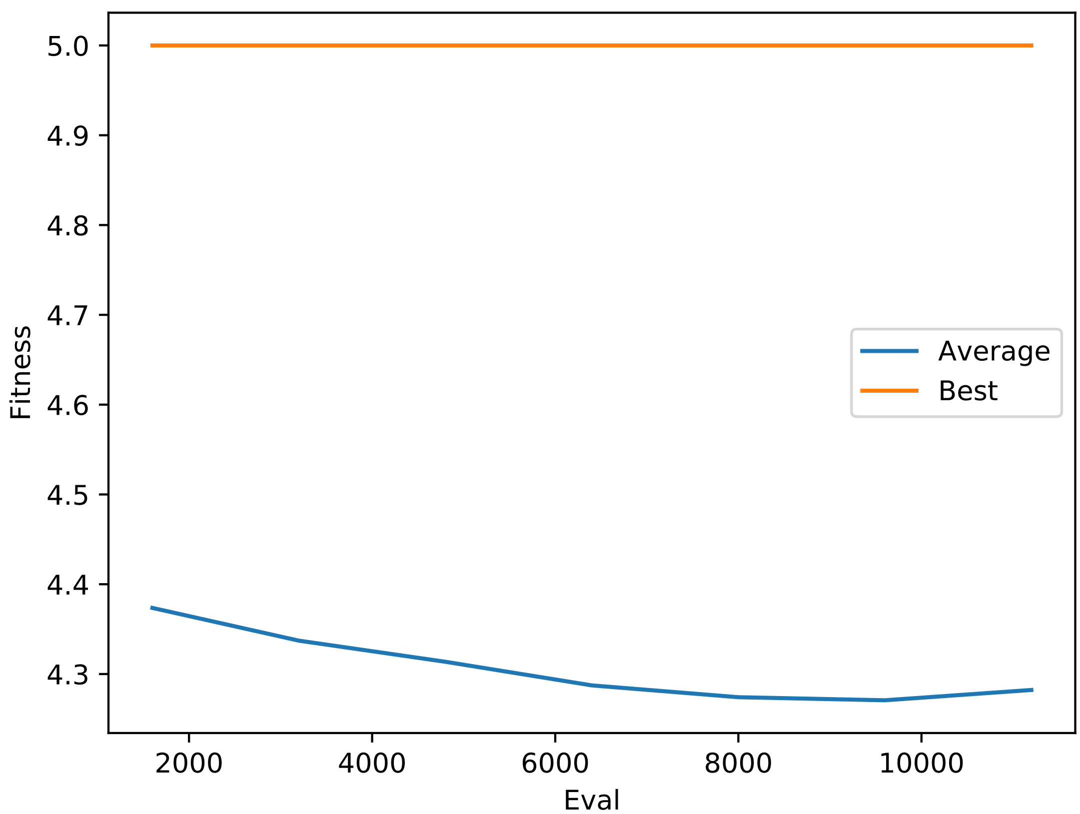

Figure 14: Statistical Analysis Comparing Base Problem to Bonus 1 and 2 using Relative Fitness

	Normal	Deter Cycling

F-Test Two-Sample for Variances

	Variable 1	*Variable 2*
Mean	3.280441	2.698237067
Variance	0.124644	1.616752952
Observations	30	30
df	29	29
F	0.077095	
P(F<=f) one-tail	3.84E-10	
F Critical one-tail	0.5374	

M(1) > M(2) ^ F < F-Critical => Equal Variance

t-Test: Two-Sample Assuming Equal Variances

	Variable 1	*Variable 2*
Mean	3.280441	2.698237067
Variance	0.124644	1.616752952
Observations	30	30
Pooled Variance	0.870698	
Hypothesized Mean Difference	0	
df	58	
t Stat	2.416502	
P(T<=t) one-tail	0.009419	
t Critical one-tail	1.296319	
P(T<=t) two-tail	0.018838	
t Critical two-tail	1.671553	

t stat > t Critical => Normal Better

Figure 15: Statistical Analysis Comparing Base Problem to Bonus 1 and 2 using Absolute Fitness

	Normal	Deter Cycling

F-Test Two-Sample for Variances

	Variable 1	*Variable 2*
Mean	4.2	4.838709677
Variance	3.26E-30	0.806451613
Observations	30	31
df	29	30
F	4.05E-30	
P(F<=f) one-tail	0	
F Critical one-tail	0.539289	

M(1) < M(2) ^ F < F-Critical => Unequal Variance

t-Test: Two-Sample Assuming Unequal Variances

	Variable 1	*Variable 2*
Mean	4.2	4.838709677
Variance	3.26E-30	0.806451613
Observations	30	31
Hypothesized Mean Difference	0	
df	30	
t Stat	-3.96	
P(T<=t) one-tail	0.000213	
t Critical one-tail	1.697261	
P(T<=t) two-tail	0.000426	
t Critical two-tail	2.042272	

t stat < t Critical => Same