# 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<br>Over Selection |
| Survival Selection | Truncation<br>K-Tournament without Replacement |
| Parsimony Pressure Penalty Coefficient | Any Float |
| Termination Convergence Criterion | Any Integer |
| Survival Selection Strategy | Plus<br>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 | 1.815103267 | 0.311954147 |
| Relative Fitness 0.5 | 3.166669567 | 0.336432956 |
| Relative Fitness 1.0 | 3.335390367 | 0.308596783 |
| Absolute Fitness 0.0 | 2.781111167 | 0.432641637 |
| Absolute Fitness 0.5 | 3 | 0 |
| Absolute Fitness 1.0 | 3 | 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, and 10 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 shows the relative progression of fitness value between coevolutionary algorithms. All three of these figures show a decrease in best fitness over time. This is likely because Prisoners are becoming better at playing against one another. Figure 1 has a general decrease in average fitness, while the figures with a non-zero parsimony pressure have a slow increase in average fitness. This may be because fitness is initially decreased due to the parsimony pressure, and increases as large trees become sparse. None of th

Figures 4, 5, and 6 shows the absolute progression of fitness value between coevolutionary algorithms. All the algorithms have a near perfect fitness against tic-for-tat, if not perfect. Although not significantly different, having a parsimony pressure does result in an optimal fitness vs not having parsimony pressure. The average absolute fitness for each three algorithms seem to be rather stable.

Figures 7, 8, 9, 10 show no significant difference between any of the parsimony pressure values. Figure 3 does however show that the mean value generally increases as parsimony pressure increases.

# 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 discussed in Section *Bonus 2*.

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

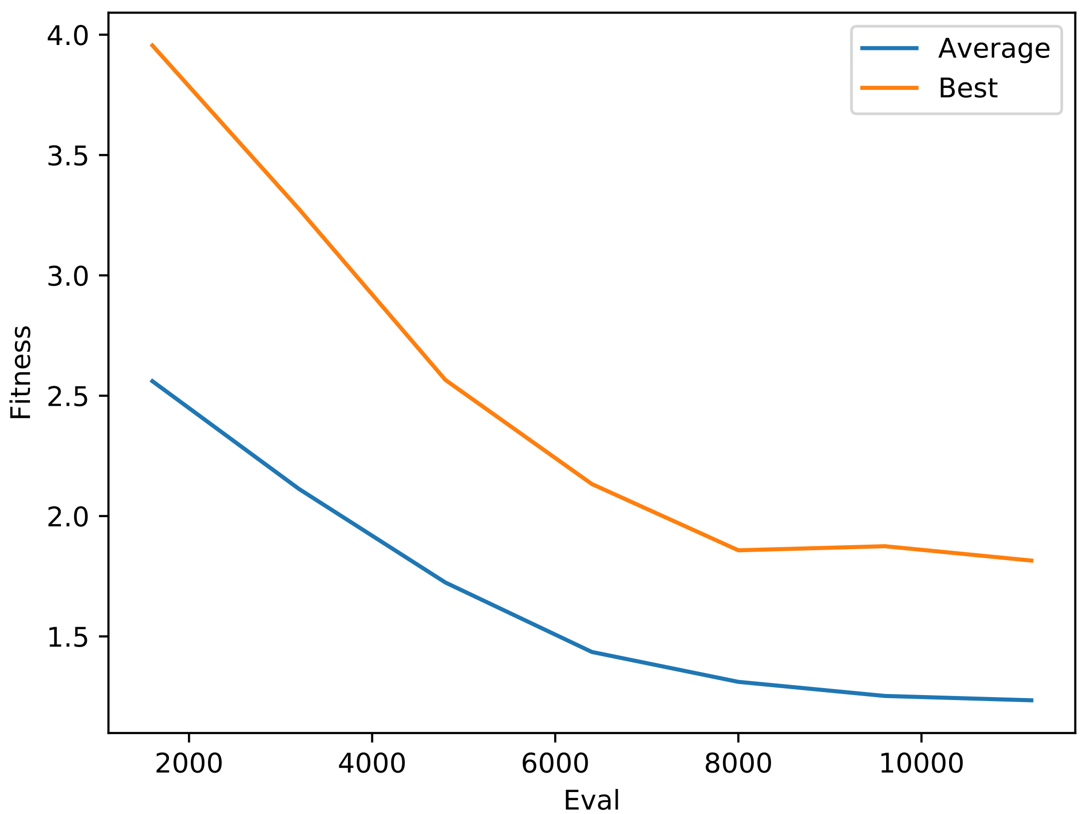

Figure 2: Relative Fitness For Parsimony Pressure Penalty Coefficient = 0.5

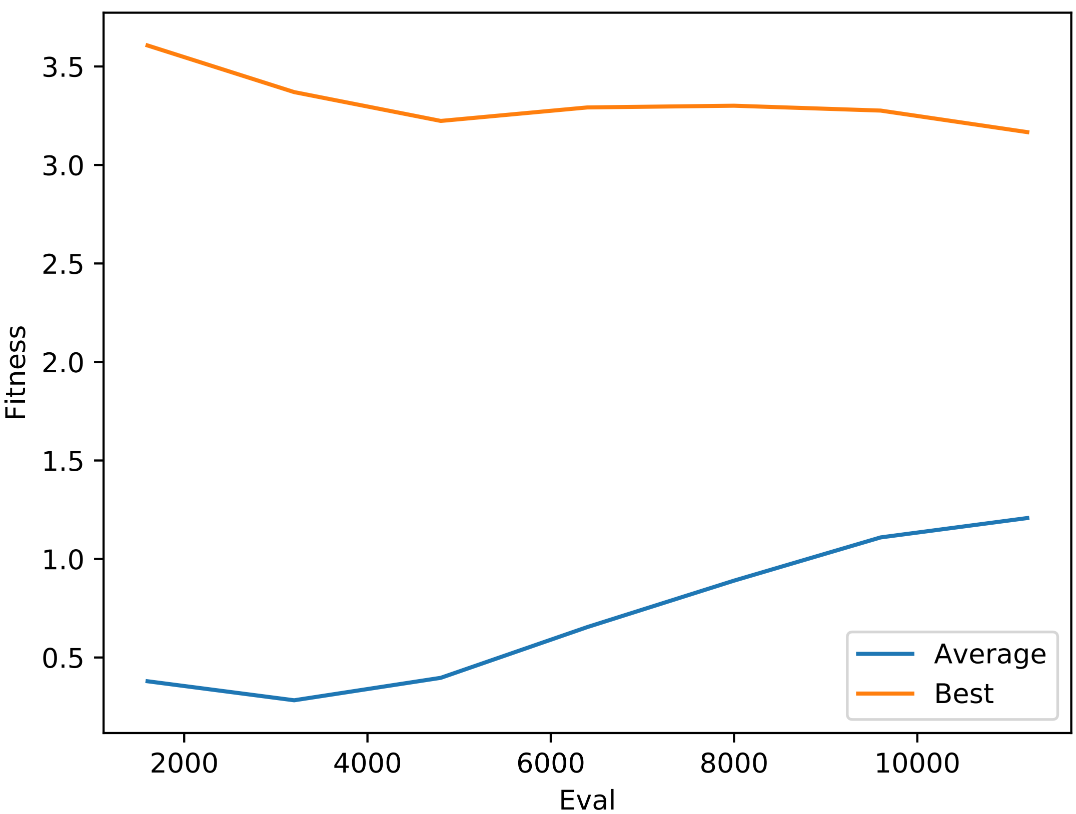

Figure 3: Relative Fitness For Parsimony Pressure Penalty Coefficient = 1.0

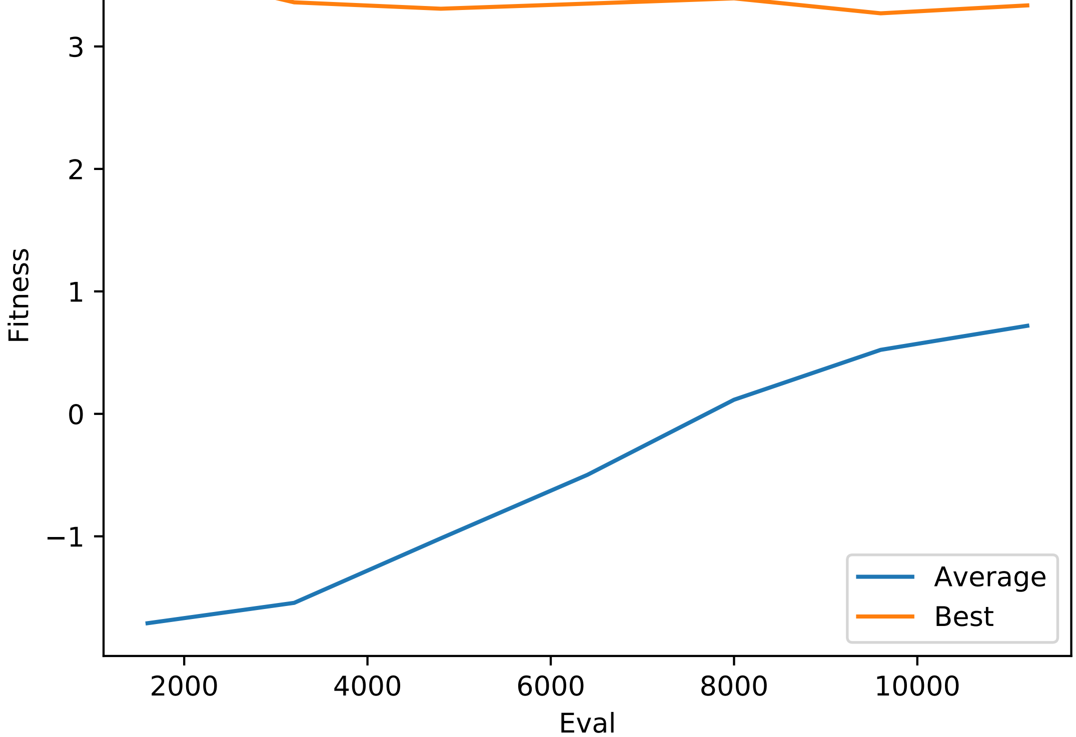

Figure 4: Absolute Fitness For Parsimony Pressure Penalty Coefficient = 0.0

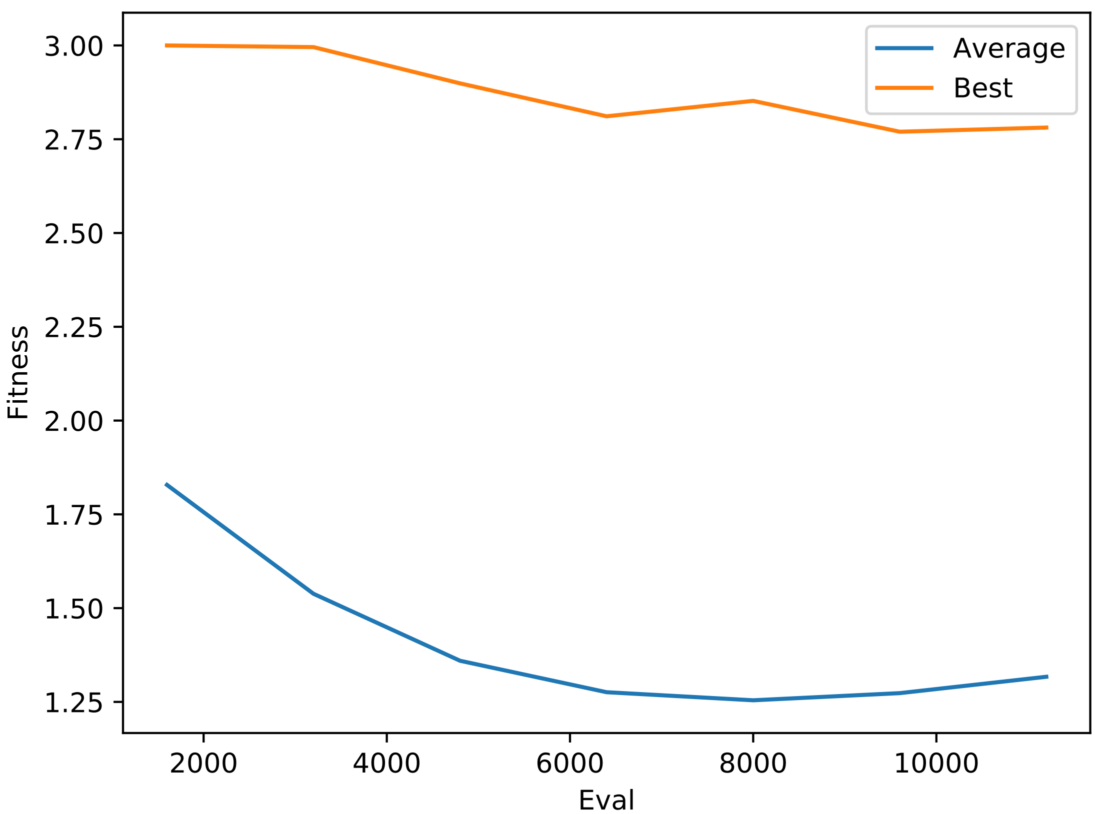

Figure 5: Absolute Fitness For Parsimony Pressure Penalty Coefficient = 0.5

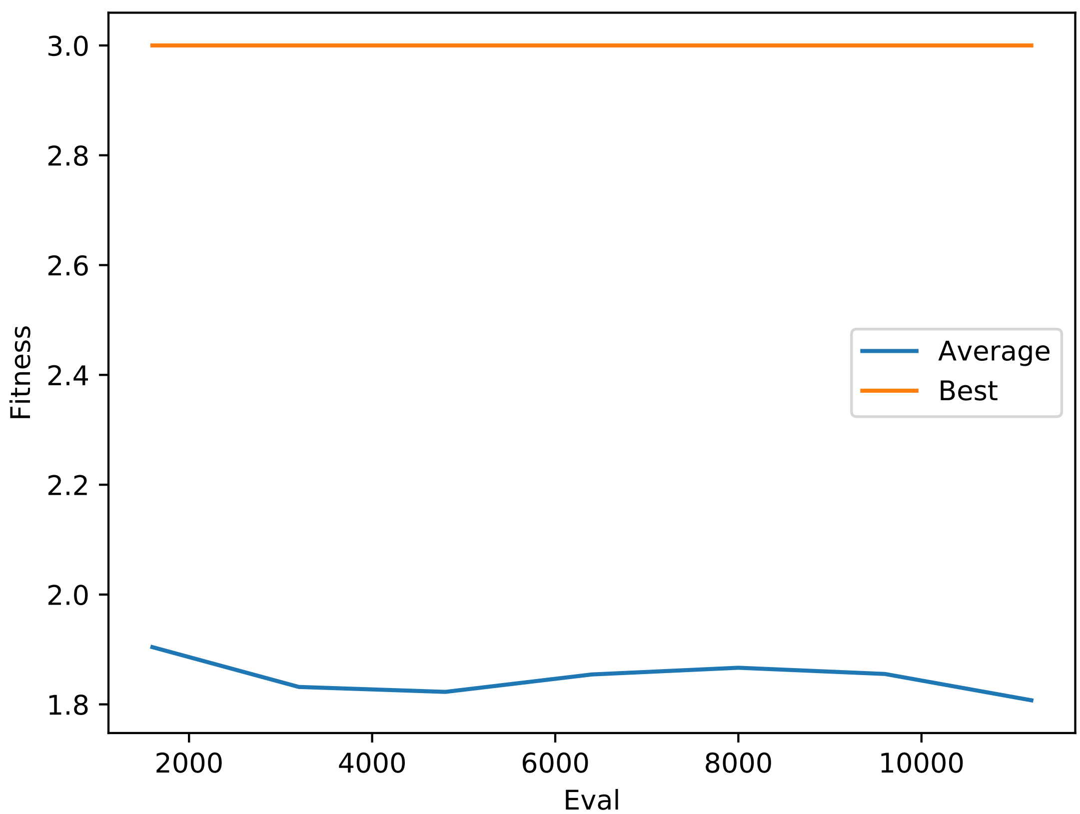

Figure 6: Absolute Fitness For Parsimony Pressure Penalty Coefficient = 1.0

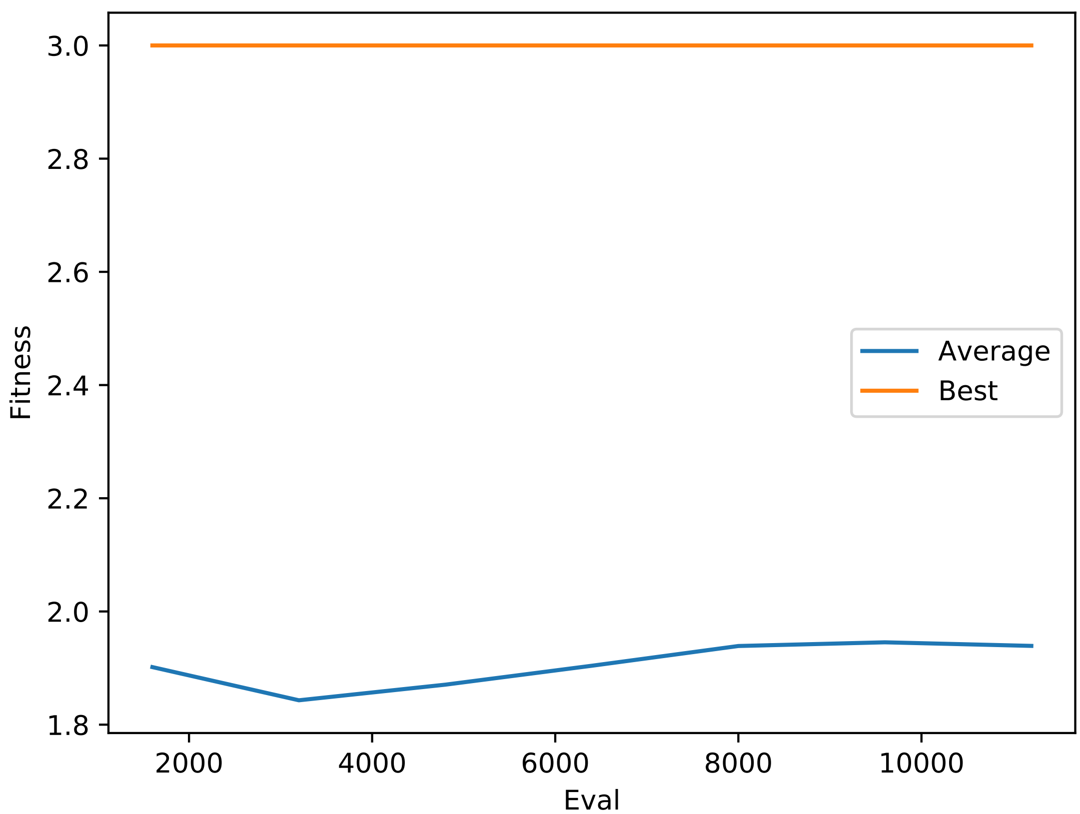

Figure 7: Statistical Analysis for Relative Parsimony Pressure Penalty Coefficient 0.0 VS 0.5

| | 0 | 0.5 |
|---|---|---|

F-Test Two-Sample for Variances

| | *Variable 1* | *Variable 2* |
|---|---|---|
| Mean | 1.815103 | 3.16667 |
| Variance | 0.097315 | 0.113187 |
| Observations | 30 | 30 |
| df | 29 | 29 |
| F | 0.859774 | |
| P(F<=f) one-tail | 0.343427 | |
| 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 | 1.815103 | 3.16667 |
| Variance | 0.097315 | 0.113187 |
| Observations | 30 | 30 |
| Pooled Variance | 0.105251 | |
| Hypothesized Mean Difference | 0 | |
| df | 58 | |
| t Stat | -16.135 | |
| P(T<=t) one-tail | 2.07E-23 | |
| t Critical one-tail | 1.671553 | |
| P(T<=t) two-tail | 4.13E-23 | |
| 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

| | 0 | 1 |
|---|---|---|

F-Test Two-Sample for Variances

| | *Variable 1* | *Variable 2* |
|---|---|---|
| Mean | 1.815103 | 3.33539 |
| Variance | 0.097315 | 0.095232 |
| Observations | 30 | 30 |
| df | 29 | 29 |
| F | 1.021877 | |
| P(F<=f) one-tail | 0.476966 | |
| F Critical one-tail | 1.860811 | |

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

t-Test: Two-Sample Assuming Unequal Variances

| | *Variable 1* | *Variable 2* |
|---|---|---|
| Mean | 1.815103 | 3.33539 |
| Variance | 0.097315 | 0.095232 |
| Observations | 30 | 30 |
| Hypothesized Mean Difference | 0 | |
| df | 58 | |
| t Stat | -18.9766 | |
| P(T<=t) one-tail | 7.42E-27 | |
| t Critical one-tail | 1.671553 | |
| P(T<=t) two-tail | 1.48E-26 | |
| 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

| | 0.5 | 1 |
|---|---|---|

F-Test Two-Sample for Variances

| | *Variable 1* | *Variable 2* |
|---|---|---|
| Mean | 3.16667 | 3.33539 |
| Variance | 0.113187 | 0.095232 |
| Observations | 30 | 30 |
| df | 29 | 29 |
| F | 1.188541 | |
| P(F<=f) one-tail | 0.322469 | |
| F Critical one-tail | 1.860811 | |

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

t-Test: Two-Sample Assuming Unequal Variances

| | *Variable 1* | *Variable 2* |
|---|---|---|
| Mean | 3.16667 | 3.33539 |
| Variance | 0.113187 | 0.095232 |
| Observations | 30 | 30 |
| Hypothesized Mean Difference | 0 | |
| df | 58 | |
| t Stat | -2.02423 | |
| P(T<=t) one-tail | 0.02378 | |
| t Critical one-tail | 1.671553 | |
| P(T<=t) two-tail | 0.04756 | |
| t Critical two-tail | 2.001717 | |

t stat < t Critical => Same

Figure 10: Statistical Analysis for Absolute Parsimony Pressure Penalty Coefficient 0.0 VS 0.5 and 1.0 (NOTE: 0.5 and 1.0 have the same absolute fitness)

0 0.5, 1

F-Test Two-Sample for Variances

| | Variable 1 | Variable 2 |
|---|---|---|
| Mean | 2.781111 | 3 |
| Variance | 0.187179 | 0 |
| Observations | 30 | 30 |
| df | 29 | 29 |
| F | 65535 | |
| P(F<=f) one-tail | #DIV/0! | |
| F Critical one-tail | 1.860811 | |

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

t-Test: Two-Sample Assuming Equal Variances

| | Variable 1 | Variable 2 |
|---|---|---|
| Mean | 2.781111 | 3 |
| Variance | 0.187179 | 0 |
| Observations | 30 | 30 |
| Pooled Variance | 0.093589 | |
| Hypothesized Mean Difference | 0 | |
| df | 58 | |
| t Stat | -2.77112 | |
| P(T<=t) one-tail | 0.003747 | |
| t Critical one-tail | 1.671553 | |
| P(T<=t) two-tail | 0.007495 | |
| t Critical two-tail | 2.001717 | |

t stat < t Critical => Same

Figure 11: Relative Fitness For Bonus 1 and 2

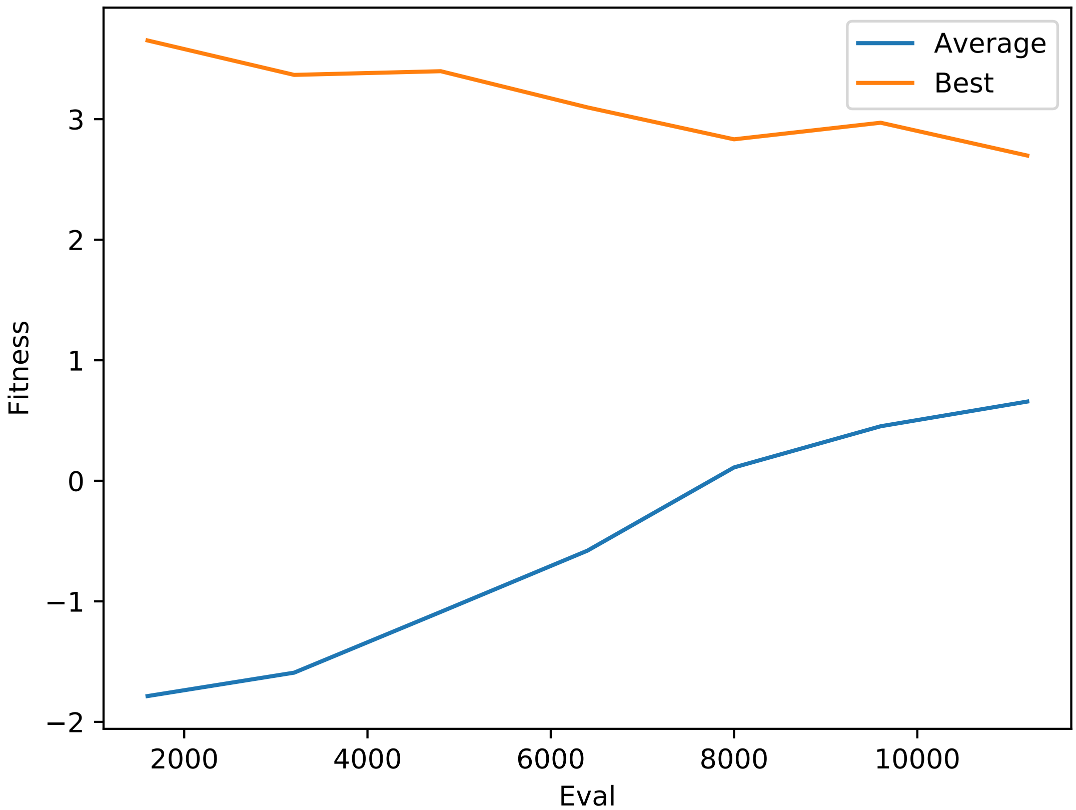


Table 4: Averages and Standard Deviations for Bonus

| | Mean | Standard Deviation |
|---|---|---|
| Relative Fitness 1.0 | 2.697945567 | 1.27891655 |
| Absolute Fitness 1.0 | 3 | 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 11, average fitness slowly increases while best fitness slowly decreases, similar to its none deterring counterpart 3. As with the main experiment, this might be due to the Prisoners getting better at playing with one another. Figure 12 shows that the absolute fitness is maximized initially and stays maximized for the rest of the run, average fitness remains mostly stagnant.

Figure 13 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 in the relative fitness case. This is probably because the best Prisoner is thrown away when a cycle is detected. However, in the absolute fitness case, both algorithms perform equally well, both performing optimally against tic-for-tat. Table 4 shows the mean and standard deviation for the bonus trial.

Figure 12: Absolute Fitness For Bonus 1 and 2

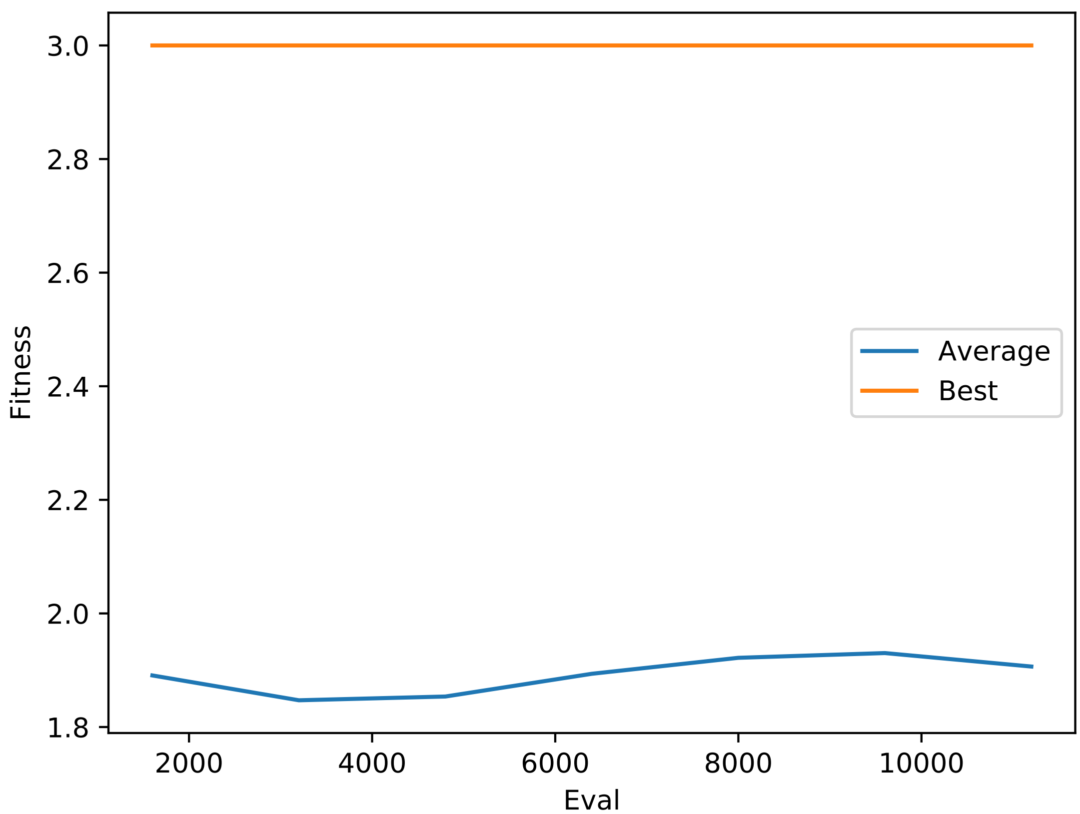

Figure 13: Statistical Analysis Comparing Base Problem to Bonus 1 and 2 using Relative Fitness

| | Normal | Deter Cycling |
|---|---|---|

F-Test Two-Sample for Variances

| | *Variable 1* | *Variable 2* |
|---|---|---|
| Mean | 3.33539 | 2.697945567 |
| Variance | 0.095232 | 1.635627541 |
| Observations | 30 | 30 |
| df | 29 | 29 |
| F | 0.058224 | |
| P(F<=f) one-tail | 1.06E-11 | |
| F Critical one-tail | 0.5374 | |

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

t-Test: Two-Sample Assuming Unequal Variances

| | *Variable 1* | *Variable 2* |
|---|---|---|
| Mean | 3.33539 | 2.697945567 |
| Variance | 0.095232 | 1.635627541 |
| Observations | 30 | 30 |
| Hypothesized Mean Difference | 0 | |
| df | 32 | |
| t Stat | 2.653825 | |
| P(T<=t) one-tail | 0.006145 | |
| t Critical one-tail | 1.693889 | |
| P(T<=t) two-tail | 0.01229 | |
| t Critical two-tail | 2.036933 | |

t stat > t Critical => Normal is Better