Zachary Bahr
zjb998@mst.edu

CS5401 FS2018 Assignment 1d

I. Methodology

The Evolutionary Algorithm employed in this assignment is a competitive coevolutionary genetic program in which populations of Ms. Pac-Man controllers and populations of ghost controllers are evolved alongside one another. Controllers are in the form of trees where nodes take the form of a function or a terminal (number), and the evaluation of the tree is simply the mathematical expression represented by these function and terminal nodes.

To evaluate the fitness of a controller, a game of Ms Pac-Man is simulated. A controller from each population is randomly selected and paired with one another. Then, before Ms. Pac-Man or the ghosts make a move in the simulation, the potential states of the game are generated and each controller is evaluated with the information from each state. The state that has the maximum value from the controller evaluation returns the move that results in that state. Following fitness evaluation, controllers are evolved via parent selection, mating, and survival selection for a specified number of games. The function and terminal sets, along with parent selection strategy, mating strategy, and survival selection strategy, are described below.

Parent Selection Strategies

- Fitness proportional: individuals are selected in proportion to their fitness
- Overselection: the population are split into the top 20% and lower 80%, and individuals are selected fitness proportionately from the top 20 percent 80 percent of the time, and individuals are selected fitness proportionately from the lower 80 percent 20 percent of the time.

Mating Strategy

- Sub-tree crossover: a subset of nodes and edges from one controller tree is swapped with a subset of nodes and edges from another controller tree.

Survival Strategies

- Truncation: the population is truncated to contain *n* controllers with the highest fitness values, where *n* is the size of the original population.

Function Set

- The function sets for both controllers contain the following functions: addition, subtraction, multiplication, division, and rand. Protected division is employed when one of the operands is 0. The rand function generates a random double between its two operands.

Terminal Sets

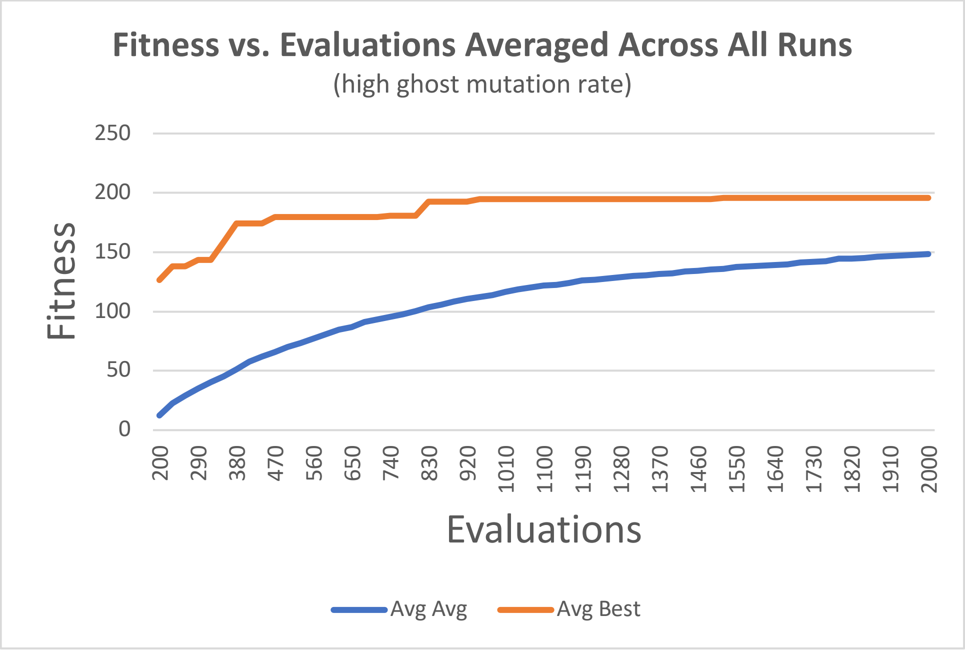
- The terminal set for the Ms. Pac-Man controller contains floating point constants and the following sensory functions: the Manhattan distance between Ms. Pac-Man and the nearest pill, the Manhattan distance between Ms. Pac-Man and the nearest ghost, the Manhattan distance between Ms. Pac-Man and the nearest fruit, and the number of walls adjacent to Ms. Pac-Man.
- The terminal set for the ghost controller contains floating point constants and the following sensory functions: the Manhattan distance between the nearest two ghosts and the Manhattan distance between the nearest ghost to Ms. Pac-Man.

II. Experimental Setup

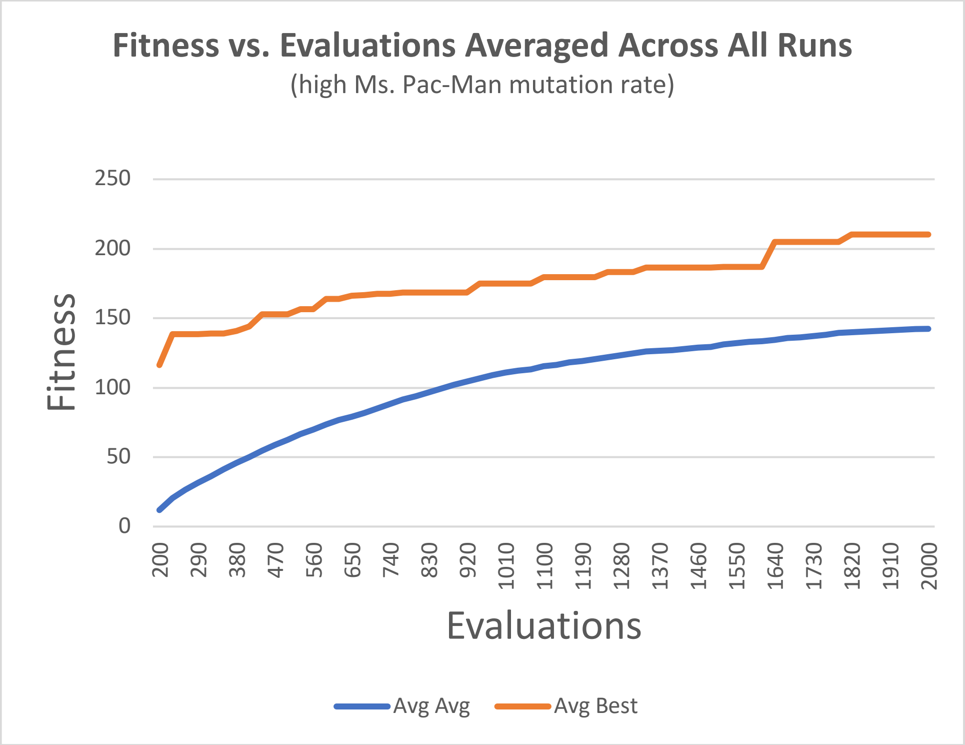
For the experiments in this assignment, population and offspring size were set to 200 and 30 for both controllers, respectively, and overselection and truncation were selected as the parent and survival selection strategies, respectively. This is due to the high-fitness controllers that were generated following multiple runs by the experimenter. It is hypothesized that these strategies performed well due to their elitist nature. This, of course, may have led to premature convergence; however, the overall quality of the solutions did not point to this.

The parameter that the experimenter chose to vary is the mutation rate for the controllers. Mutation is the key to genetic diversity in any population, so adjusting the mutation rate for the ghosts and Ms. Pac-Man should have an impact on the quality of solutions. The experimenter ran three experiments: one with the mutation rate high for Ms. Pac-Man and low for the ghosts, one with mutation rate low for Ms. Pac-Man and high for the ghosts, and one with the mutation rates equal to serve as a control.

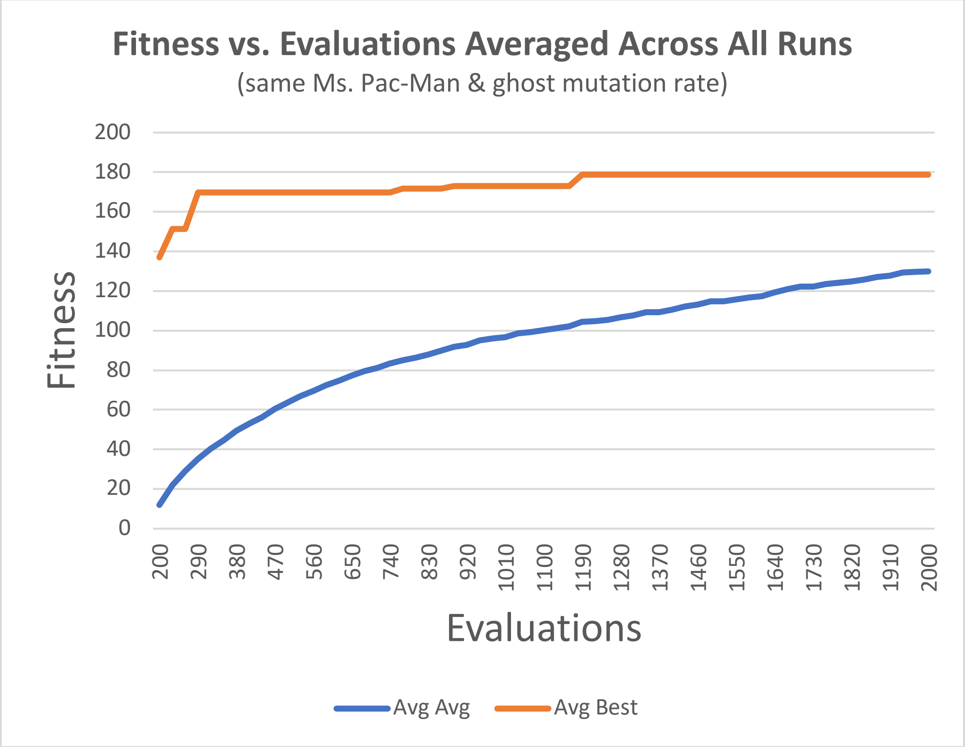
III. Plots



Plot 1. Fitness of Ms. Pac-Man controllers vs. evaluations when ghost mutation rate is high



Plot 2. Fitness of Ms. Pac-Man controllers vs. evaluations when Ms. Pac-Man mutation rate is high



Plot 3. Fitness of Ms. Pac-Man controllers vs. evaluations when Ms. Pac-Man and ghost mutation rate is high

IV. Statistical Analyses

Figure 1. F-Test Two-Sample for Variances

	High Ms. Pac Mutation	*Same Mutation Rate*
Mean	210.25	178.75
Variance	827.5833333	910.25
Observations	4	4
df	3	3
F	0.909182459	
P(F<=f) one-tail	0.469728179	
F Critical one-tail	0.107797789	

Figure 2. t-Test: Two-Sample Assuming Unequal Variances

	High Ms. Pac Mutation	*Same Mutation Rate*
Mean	210.25	178.75
Variance	827.5833333	910.25
Observations	4	4
Hypothesized Mean Difference	0	
df	6	
t Stat	1.511250606	
P(T<=t) one-tail	0.090737868	
t Critical one-tail	1.943180281	
P(T<=t) two-tail	0.181475735	
t Critical two-tail	2.446911851	

As Figure 1 indicates, the two variables are assumed to have unequal variances because |mean(v1) > mean(v2)| and F > F Critical. Thus, a two- sample T-Test assuming unequal variances is employed with the null hypothesis being that the mean difference between the two variables is zero. Because |t Stat| > | t Critical two-tail - Tail|, we reject the null hypothesis and can conclude that the algorithm that produced Variable 1 is statistically better than the algorithm that produced Variable 2. (see Table 1 in Appendix for data)

Figure 3. F-Test Two-Sample for Variances

	High Ghost Mutation	*High Ms. Pac Mutation*
Mean	195.75	210.25
Variance	400.9166667	827.5833333
Observations	4	4
df	3	3
F	0.484442654	
P(F<=f) one-tail	0.283426572	
F Critical one-tail	0.107797789	

Figure 4. t-Test: Two-Sample Assuming Equal Variances

	High Ghost Mutation	*High Ms. Pac Mutation*
Mean	195.75	210.25
Variance	400.9166667	827.5833333
Observations	4	4
Pooled Variance	614.25	
Hypothesized Mean Difference	0	
df	6	
t Stat	-0.82739029	
P(T<=t) one-tail	0.219837748	
t Critical one-tail	1.943180281	
P(T<=t) two-tail	0.439675496	
t Critical two-tail	2.446911851	

As Figure 3 indicates, the two variables are assumed to have equal variances because |mean(v1) < mean(v2)| and F > F Critical. Thus, a two- sample T-Test assuming equal variances is employed with the null hypothesis being that the mean difference between the two variables is zero. Because |t Stat| > | t Critical two-tail - Tail|, we reject the null hypothesis and can conclude that the algorithm that produced Variable 2 is statistically better than the algorithm that produced Variable 1. (see Table 2 in Appendix for data)

Figure 5. F-Test Two-Sample for Variances

	High Ghost Mutation	*Same Mutation Rate*
Mean	195.75	178.75
Variance	400.9166667	910.25
Observations	4	4
df	3	3
F	0.440446764	
P(F<=f) one-tail	0.259068445	
F Critical one-tail	0.107797789	

Figure 6. t-Test: Two-Sample Assuming Unequal Variances

	High Ghost Mutation	*Same Mutation Rate*
Mean	195.75	178.75
Variance	400.9166667	910.25
Observations	4	4
Hypothesized Mean Difference	0	
df	5	
t Stat	0.938966217	
P(T<=t) one-tail	0.195428331	
t Critical one-tail	2.015048373	
P(T<=t) two-tail	0.390856663	
t Critical two-tail	2.570581836	

As Figure 5 indicates, the two variables are assumed to have unequal variances because |mean(v1) > mean(v2)| and F > F Critical. Thus, a two- sample T-Test assuming equal variances is employed with the null hypothesis being that the mean difference between the two variables is zero. Because |t Stat| > | t Critical two-tail - Tail|, we reject the null hypothesis and can conclude that the algorithm that produced Variable 1 is statistically better than the algorithm that produced Variable 2. (see Table 3 in Appendix for data)

V. Discussion and Conclusion

As the statistical analyses in Section IV convey, the experiments in which Ms. Pac-Man had a higher mutation rate produced significantly better results. In context, this makes sense. With a higher mutation rate than the ghosts, Ms. Pac-Man was able to escape local optima more often, and thus was able to find better results.

It is interesting to note the third experiment in which the algorithm with higher ghost mutation produced better results than the algorithm with the same ghost and mutation rate. This means that higher ghost mutation led to better resulting Ms. Pac-Man controllers. The experimenter hypothesizes that this is due to the selective pressure put on the Ms. Pac-Man controllers by the high ghost mutation rate, which drove the Ms. Pac-Man controllers to evolve faster than the ghosts.

Zachary Bahr
zjb998@mst.edu

Appendix

Run	High Ms. Pac-Man Mutation Best Fitness	High Ms. Pac-Man & Ghost Mutation Best Fitness
1	213	194
2	197	199
3	182	134
4	249	188
Std. dev.	28.76774814	30.17034968

Table 1

Run	High Ghost Mutation Best Fitness	High Ms. Pac-Man Mutation Best Fitness
1	220	213
2	195	197
3	171	182
4	197	249
Std. dev.	20.02290355	28.76774814

Table 2

Run	High Ghost Mutation Best Fitness	High Ms. Pac-Man & Ghost Mutation Best Fitness
1	220	194
2	195	199
3	171	134
4	197	188
Std. dev.	20.02290355	30.17034968

Table 3