Project 6 Report: Particle Swarm Optimization

Patrick Elam

CS420

May 6, 2015

Introduction:

Particle swarm optimization (PSO) is a computational optimization method that utilizes a swarm of potential solutions to a problem. These potential solutions are manifested as particles, each with given X and Y coordinates on a simulated plane. These particles are randomly initialized on the plane, and then the position of each particle is iteratively updated towards a better position. The qualification for a better position is based on a fitness function. With solutions on an X and Y plane, the fitness function must take two variables, X and Y. A fitness function with 3 variables can be solved using a similar method, but in a 3 dimensional volume.

For every iteration of the particle swarm algorithm, each particle in the swarm is looped through, and each particle's position and velocity in the plane is updated. The new direction of each particle is set to be toward the current particle with the best fitness (or whose X and Y coordinates give the highest output when plugged into the fitness equations. Velocity and inertia are then accounted for and a new position for the particle is established. The particle's fitness is then recalculated and compared to the fitness of the best current particle. If the current particle now has a higher fitness, then it becomes the global best.

The advantage to this type of algorithm is that a wide range of potential solutions to the fitness equation(s) are considered. This includes solutions that humans might not have before considered. This type of algorithm is best suited to finding solutions for problems that need to be optimized, and not necessarily for problems that have only an exact solution. One disadvantage to this type of solution finding is that the problem given must have some quantifiable way of determining fitness.

Simulator and Calculations:

Alex Chaloux and I wrote a program in C++ that would find the solution to a problem using particle swarm optimization. We were instructed to solve the following two problems, which served as fitness functions for the particles:

Problem 1:

$$Q(p_x, p_y) = 100 \cdot \left(1 - \frac{pdist}{mdist}\right)$$

Problem 2:

$$Q(p_x, p_y) = 9 \cdot \max(0, 10 - pdist^2) + 10 \cdot \left(1 - \frac{pdist}{mdist}\right) + 70 \cdot \left(1 - \frac{ndist}{mdist}\right)$$

The definitions for pdist(), mdist() and ndist() are as follows:

$$mdist = \sqrt{\max_x{}^2 + \max_y{}^2} / 2$$

$$pdist = \sqrt{(p_x - 20)^2 + (p_y - 7)^2}$$

$$ndist = \sqrt{(p_x + 20)^2 + (p_y + 7)^2}$$

For these two above problems, we tried a myriad of configurations to run through our simulator. These variables were the following. 1) The number of particles in the swarm. 2) The inertia modifier for the particles, which affected how well the particles changed direction. 3) The cognition modifier, which affected how the velocity was changed with accordance to the particle's previous best. 4) The social modifier, which affected how the velocity was changed with accordance to the particle's relation to the global best. 5) The maximum velocity value for the particles.

For the configurations that we ran through the simulator, we wrote a script to generate every combination for the 5 variables. The possible values for the number of particles were: 10, 20, 30, 40, 50, and 100. The possible values for the inertia modifier were: 0.1, 0.2, 0.35, 0.5, 0.7, 0.85, and 0.95. The possible values for the cognitive parameter were: 0.5, 1.25, 1.75, 2, 2.25, 2.75, 3.5, and 4.5. The possible values for the social parameter were: 0.5, 1.25, 1.75, 2, 2.25, 2.75, 3.5, and 4.5. The possible values for maximum particle velocity were: 0.5, 1.5, 2.5, 3.5, 5, 7, and 10. Each variation of parameters was run through each problem, and thus we had a total of 37,632 configurations to run through our simulator. We also added the ability to limit the number of iterations, or epochs, that our simulation ran, such that the simulation would stop when either the number of epochs had been reached or the global error of all the particles was below 0.001. While the number of epochs was technically also an input for our simulator, it acted more like an output. This is because we ran the simulator until 1000 epochs was reached, and we could tell at what point the simulator stopped, which indicated that all the particles converged before 1000 epochs were reached. If the particles in the simulator did not converge before 1000 epochs, it was a good indicator that that set of input variables was not efficient. The output from these configurations was thrown out during graphing.

After running all of the configurations through the simulator, we obtained the following data from each run and compiled it into a master spreadsheet, alongside the input variables for that run. We pulled the following information from each run of the simulator. 1) Number of epochs it took the simulation to converge. 2) Average X position of all the particles. 3) Average Y position of all the particles. 4) Percentage of particles with an individual error less than the error threshold of 0.001. 5) Average fitness of the

population. 6) The global best fitness, or the fitness of the most fit individual. 7) The X position of the most fit individual. 8) The Y position of the most fit individual.

Once we had all of the above information in a master Excel sheet, we found the correlation value between each input variable and each output variable. We then sorted through all of the simulation runs and picked a four of the more interesting configurations with which we made a few more graphs. In these graphs, we compared the average X and average Y values to the global best X and Y values against time (number of epochs). We also compared average fitness and global best fitness of the swarm against time (number of epochs). With these four hand picked configurations, we also developed a system to generate animations of the swarms over time.

Results:

First, here is a chart of the correlations between the input and output variables.

	Number of Epochs	Percentage Within Error	Average Fitness	Adjusted Average Fitness	Global Best Fitness	Adjusted Global Best Fitness
Number of Particles	0.0789	-0.0376	0.0083	0.0108	0.2395	0.2006
Inertia	0.3822	-0.3054	-0.1832	-0.2200	0.0929	0.0771
Cognition	0.3912	-0.3633	-0.2384	-0.2910	0.0733	0.0611
Social	0.3961	-0.3433	-0.1893	-0.2230	0.0647	0.0532
Problem Number	0.1511	-0.1992	-0.5689	-0.0135	-0.2822	0.5915
Maximum Velocity	-0.0830	0.0494	-0.3643	-0.4096	-0.0001	-0.0001

T

Below the same chart as above, except the correlation values are the absolute values and color coded for easier viewing.

	Number of Epochs	Percentage Within Error	Average Fitness	Adjusted Average Fitness	Global Best Fitness	Adjusted Global Fitness
Number of Particles	0.0789	0.0376	0.0083	0.0108	0.2395	0.2006
Inertia	0.3822	0.3054	0.1832	0.2200	0.0929	0.0771
Cognition	0.3912	0.3633	0.2384	0.2910	0.0733	0.0611
Social	0.3961	0.3433	0.1893	0.2230	0.0647	0.0532
Problem Number	0.1511	0.1992	0.5689	0.0135	0.2822	0.5915
Maximum Velocity	0.0830	0.0494	0.3643	0.4096	0.0001	0.0001

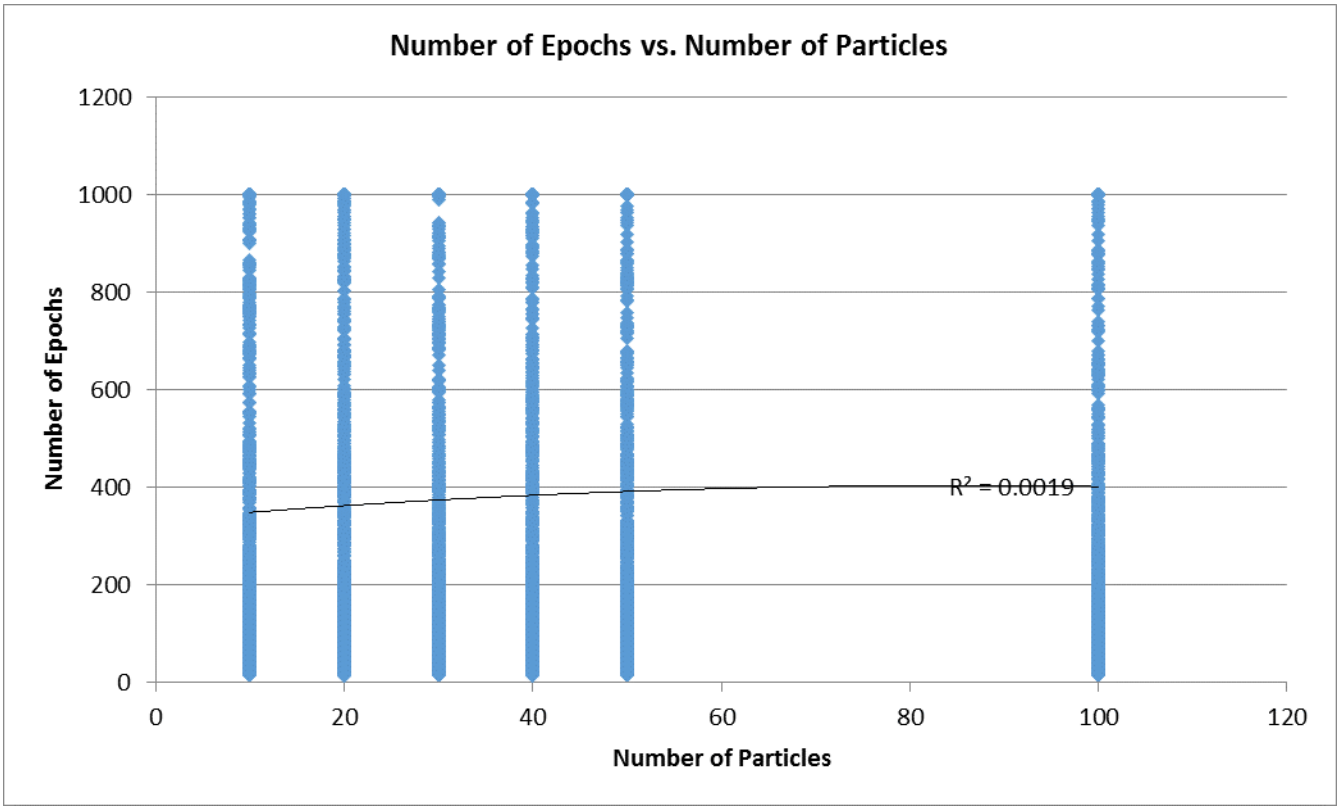
After charting the correlations between the variables and the outputs, we had Excel calculate a quadratic line of best fit, and charted the r^2 values of these lines. Below is the resulting chart.

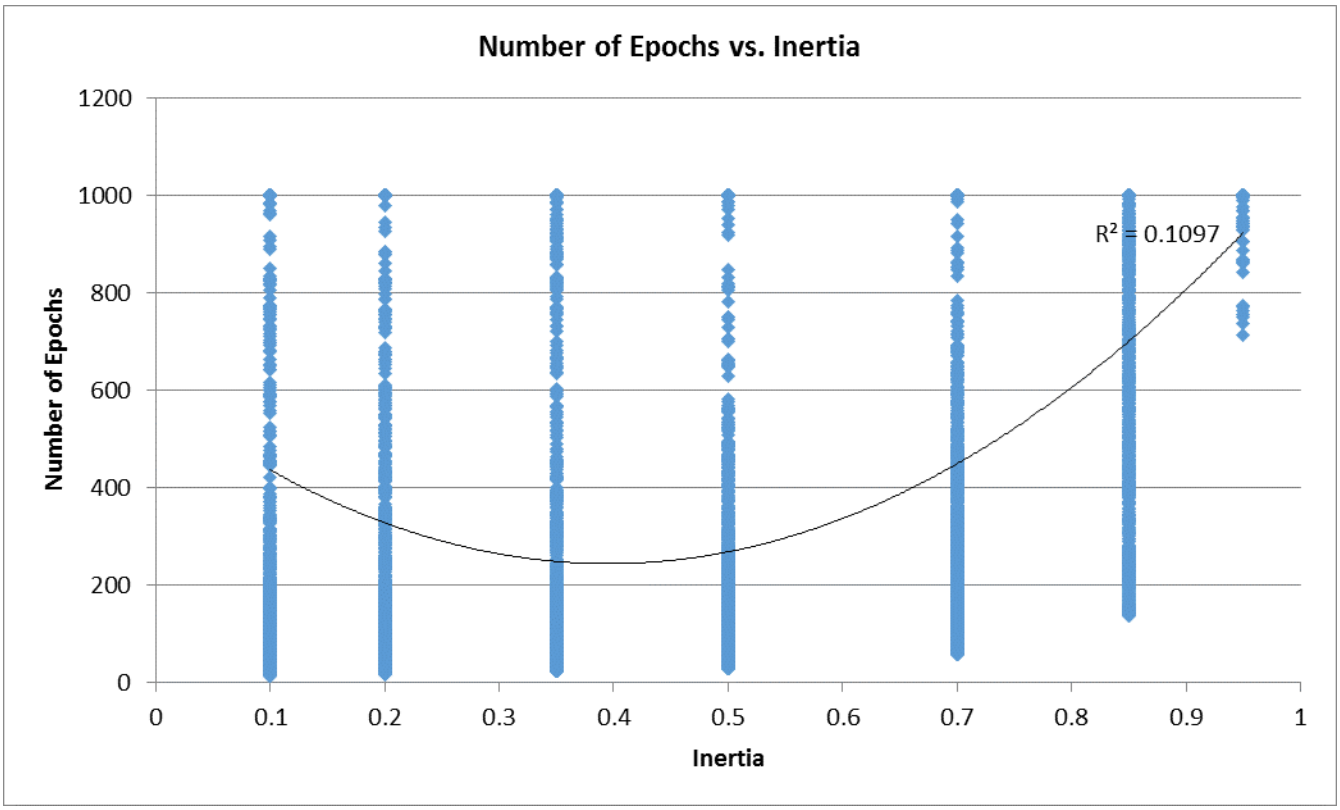
	Number of Epochs	Percentage Within Error	Average Fitness	Adjusted Average Fitness	Global Best Fitness	Adjusted Global Best Fitness
Number of Particles	0.0066	0.0022	0.0001	0.0001	0.0610	0.0427
Inertia	0.1700	0.1289	0.0489	0.0706	0.0090	0.0062
Cognition	0.1532	0.1343	0.0568	0.0848	0.0058	0.0040
Social	0.1577	0.1307	0.0395	0.0554	0.0042	0.0028
Problem Number	0.0228	0.0397	0.3237	0.0002	0.0796	0.3499
Maximum Velocity	0.0090	0.0025	0.1436	0.1820	0.0112	0.0080

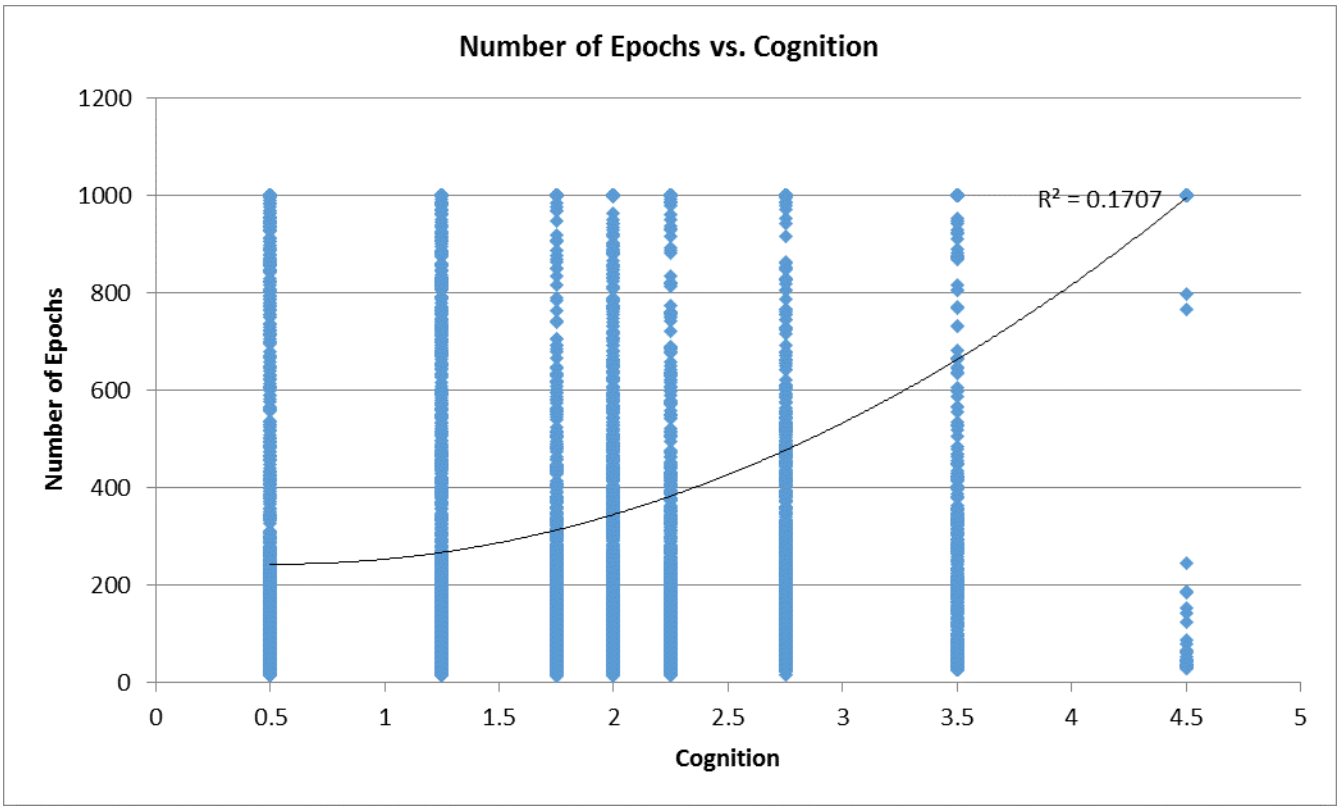
Again, below is the same chart, but color coded for easier viewing.

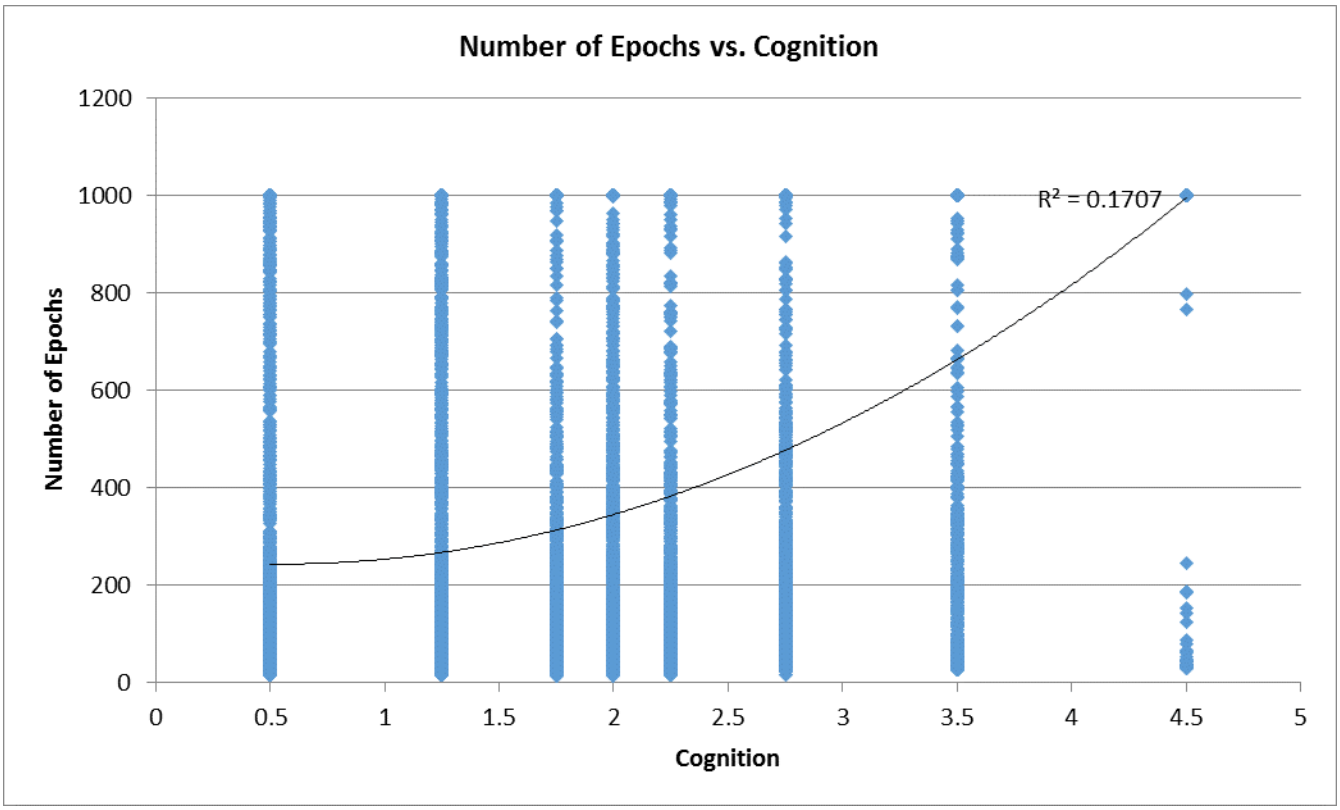
	Number of Epochs	Percentage Within Error	Average Fitness	Adjusted Average Fitness	Global Best Fitness	Adjusted Global Fitness
Number of Particles	0.0066	0.0022	0.0001	0.0001	0.0610	0.0427
Inertia	0.1700	0.1289	0.0489	0.0706	0.0090	0.0062
Cognition	0.1532	0.1343	0.0568	0.0848	0.0058	0.0040
Social	0.1577	0.1307	0.0395	0.0554	0.0042	0.0028
Problem Number	0.0228	0.0397	0.3237	0.0002	0.0796	0.3499
Maximum Velocity	0.0090	0.0025	0.1436	0.1820	0.0112	0.0080

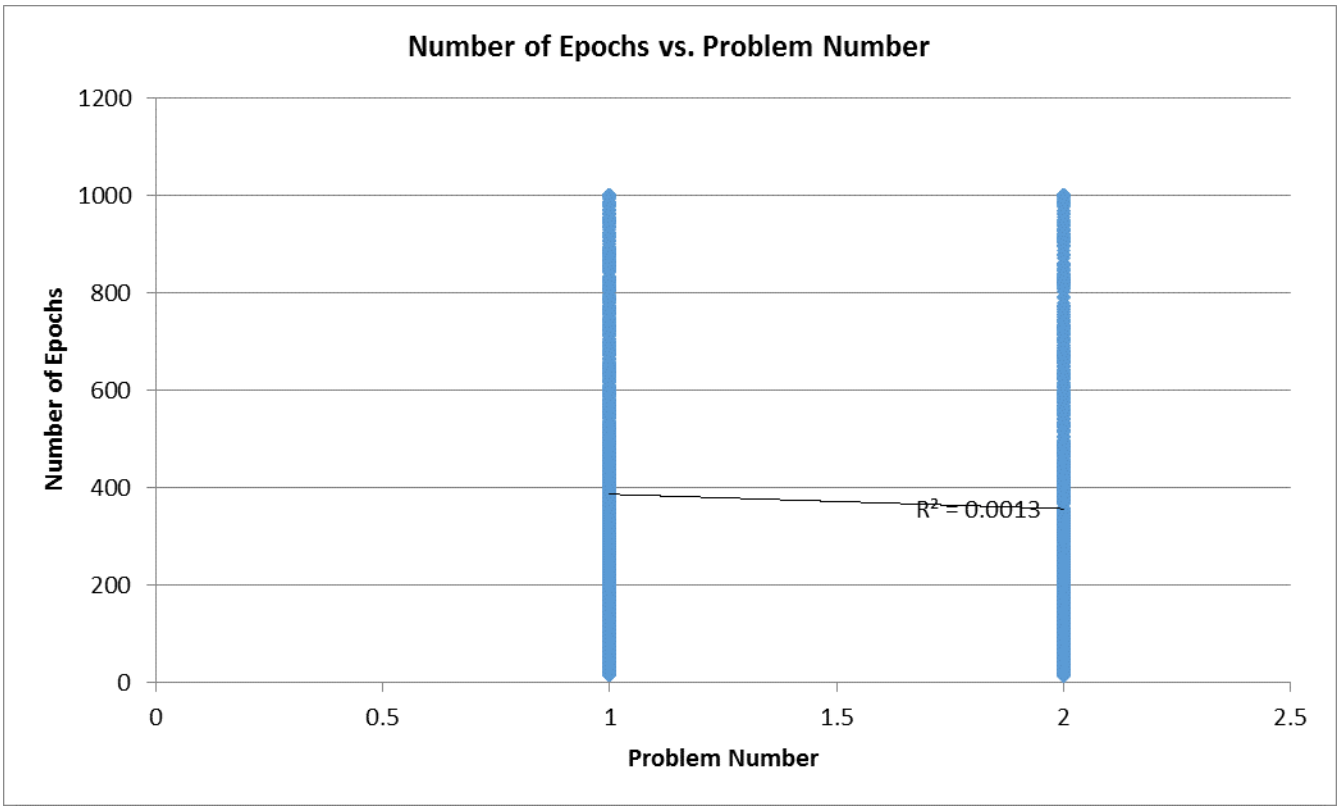
For each combination above, we removed data from runs that did not converge within 1000 epochs, and then graphed each input variable against each output variable. Below are some of the more interesting graphs. All can be seen on in the master Excel sheet.

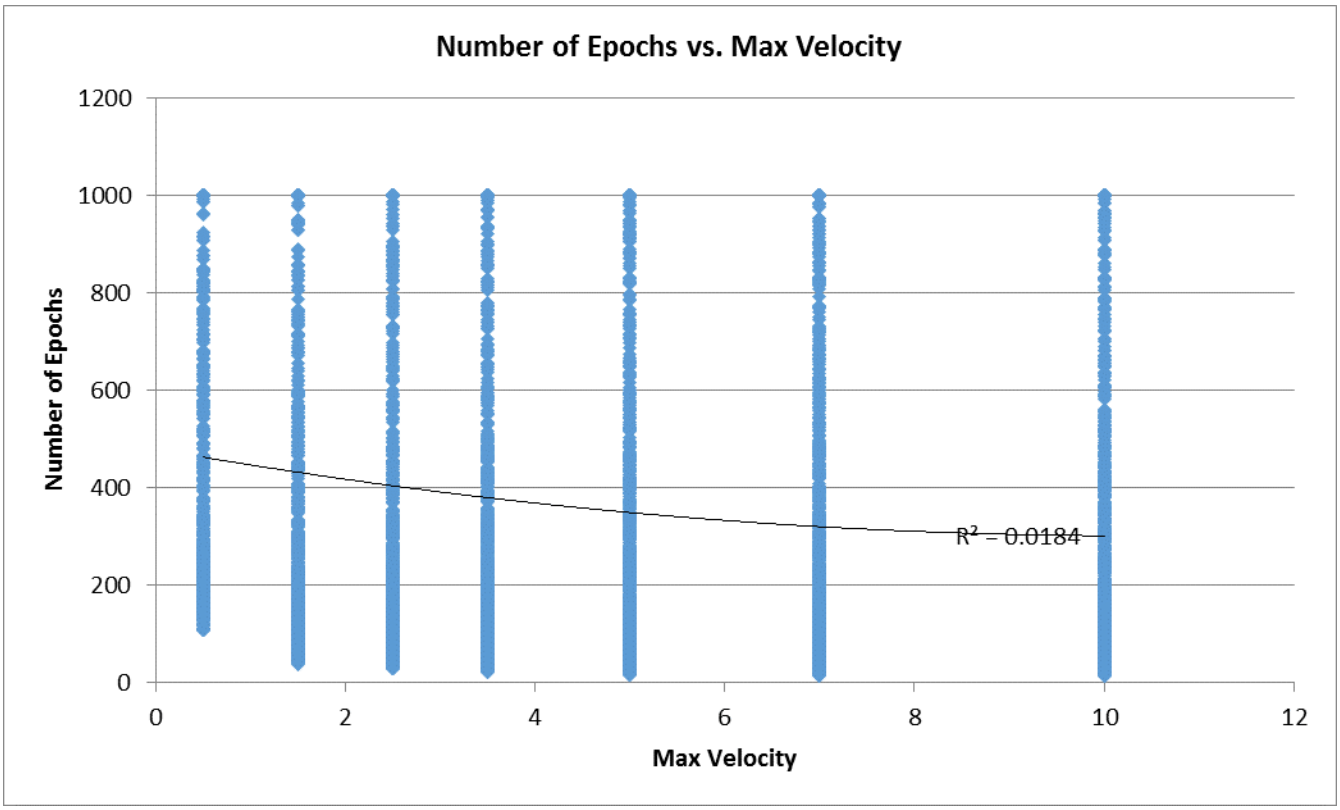


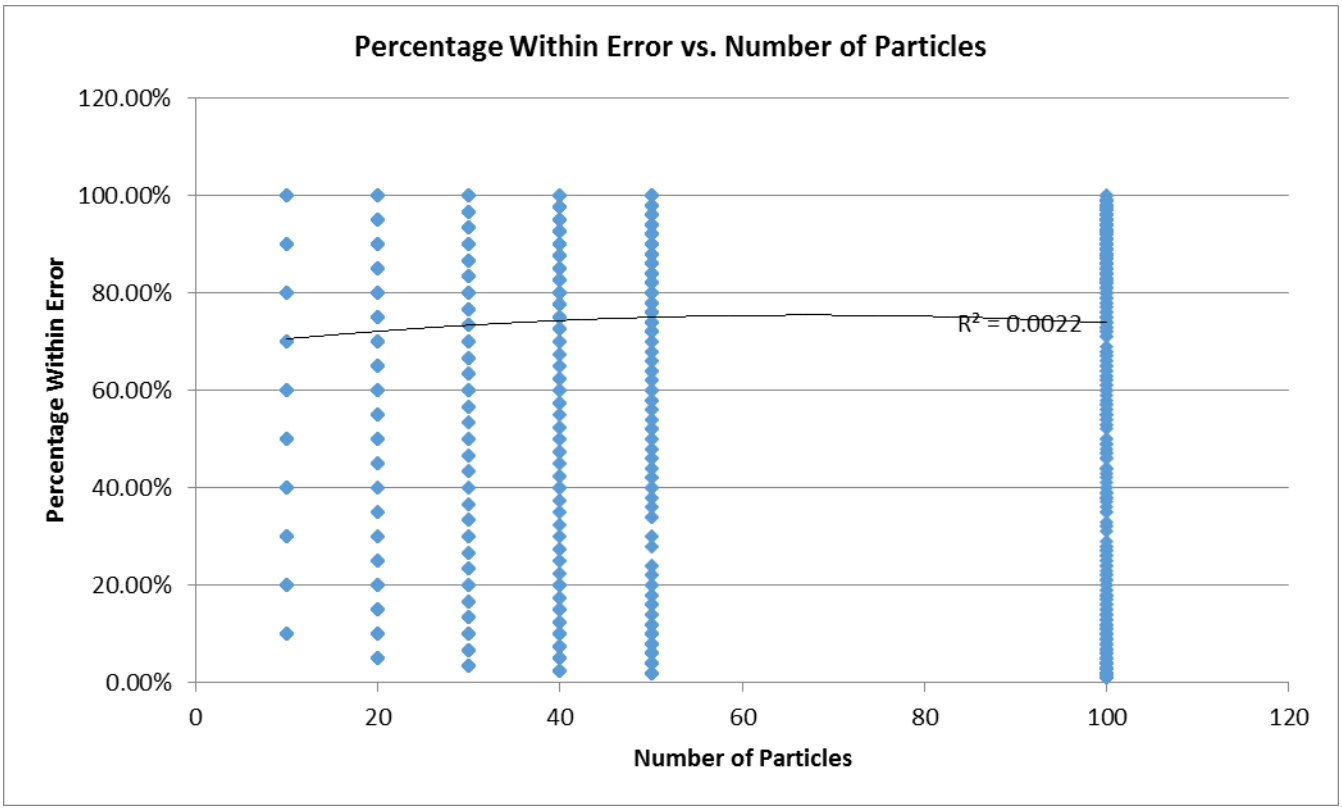

Number of Epochs vs. Inertia
Number of Epochs
Inertia
R² = 0.1097
0
200
400
600
800
1000
1200
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

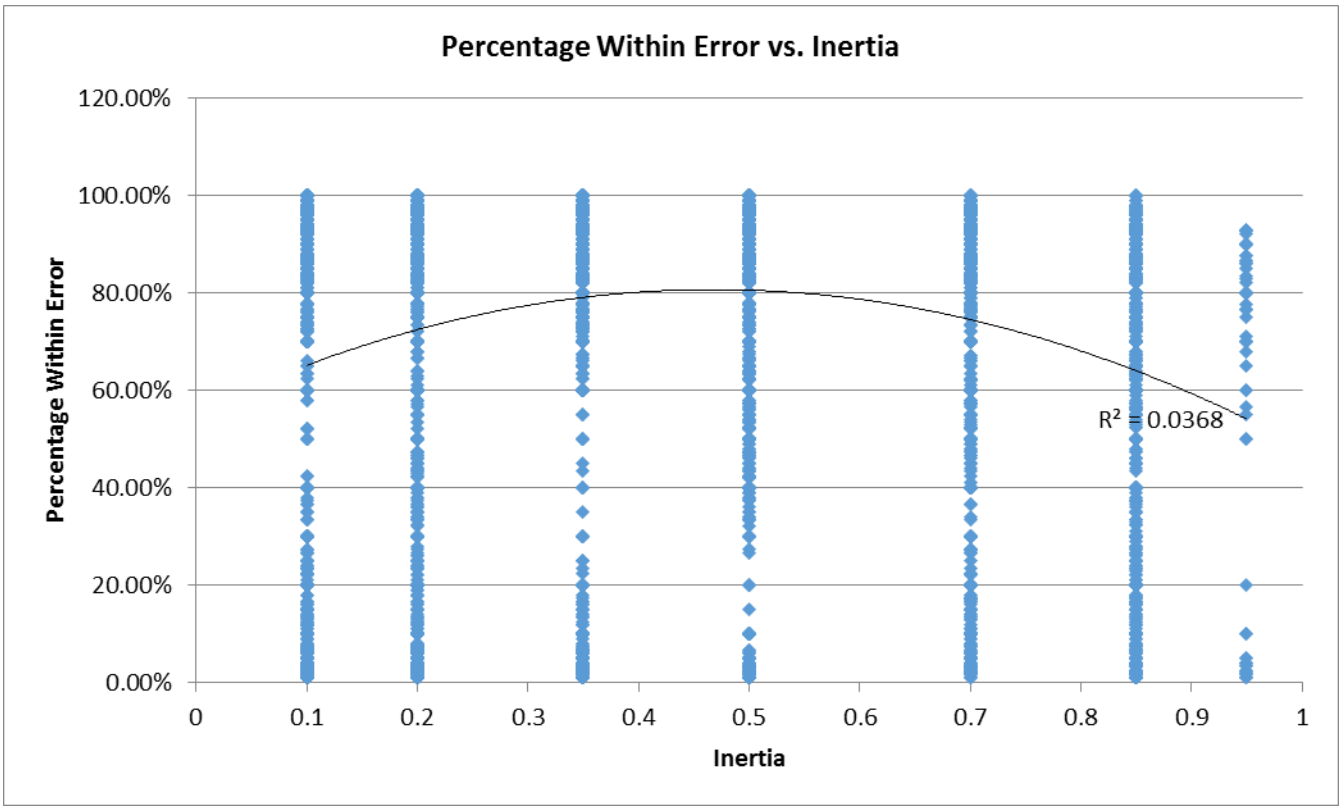

Number of Epochs vs. Cognition
Number of Epochs
Cognition
R² = 0.1707
0
200
400
600
800
1000
1200
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

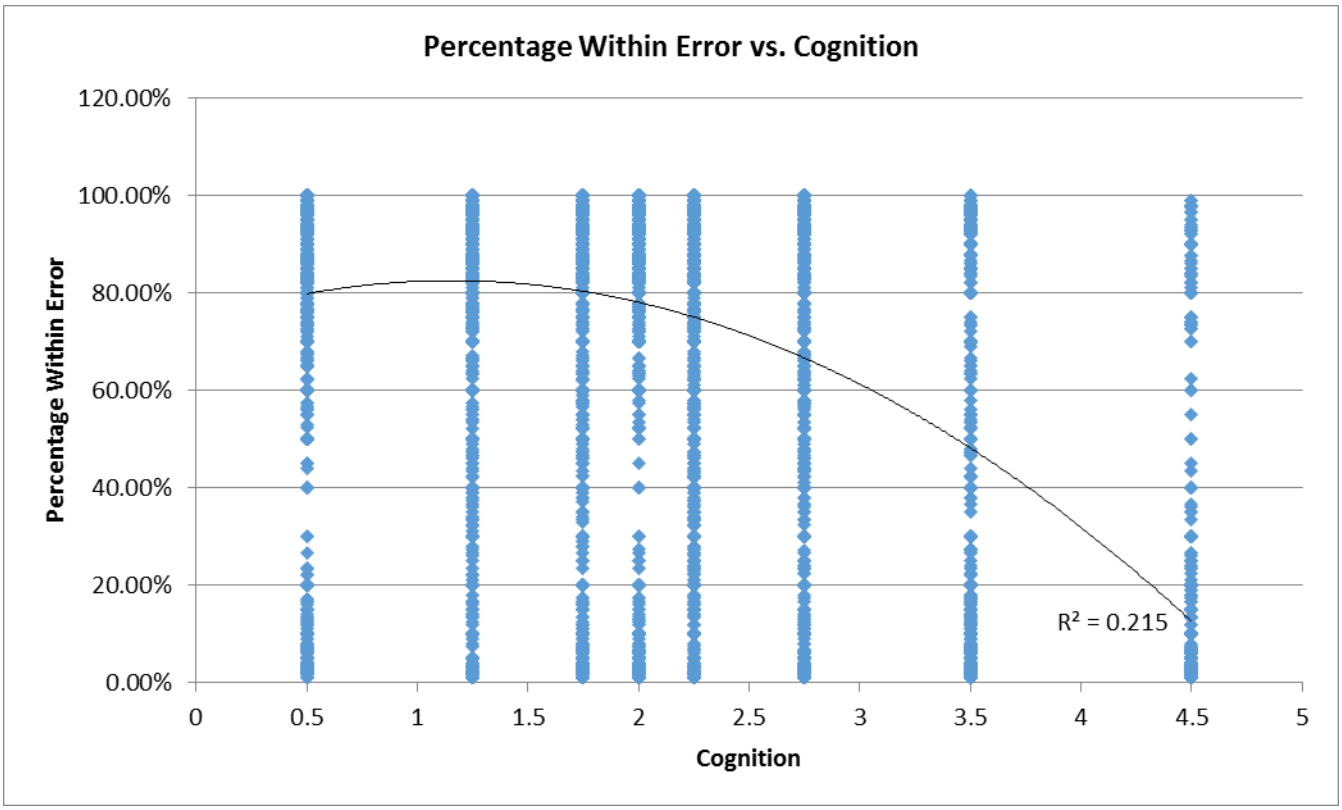

Number of Epochs vs. Cognition
Number of Epochs
Cognition
R² = 0.1707
0
200
400
600
800
1000
1200
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

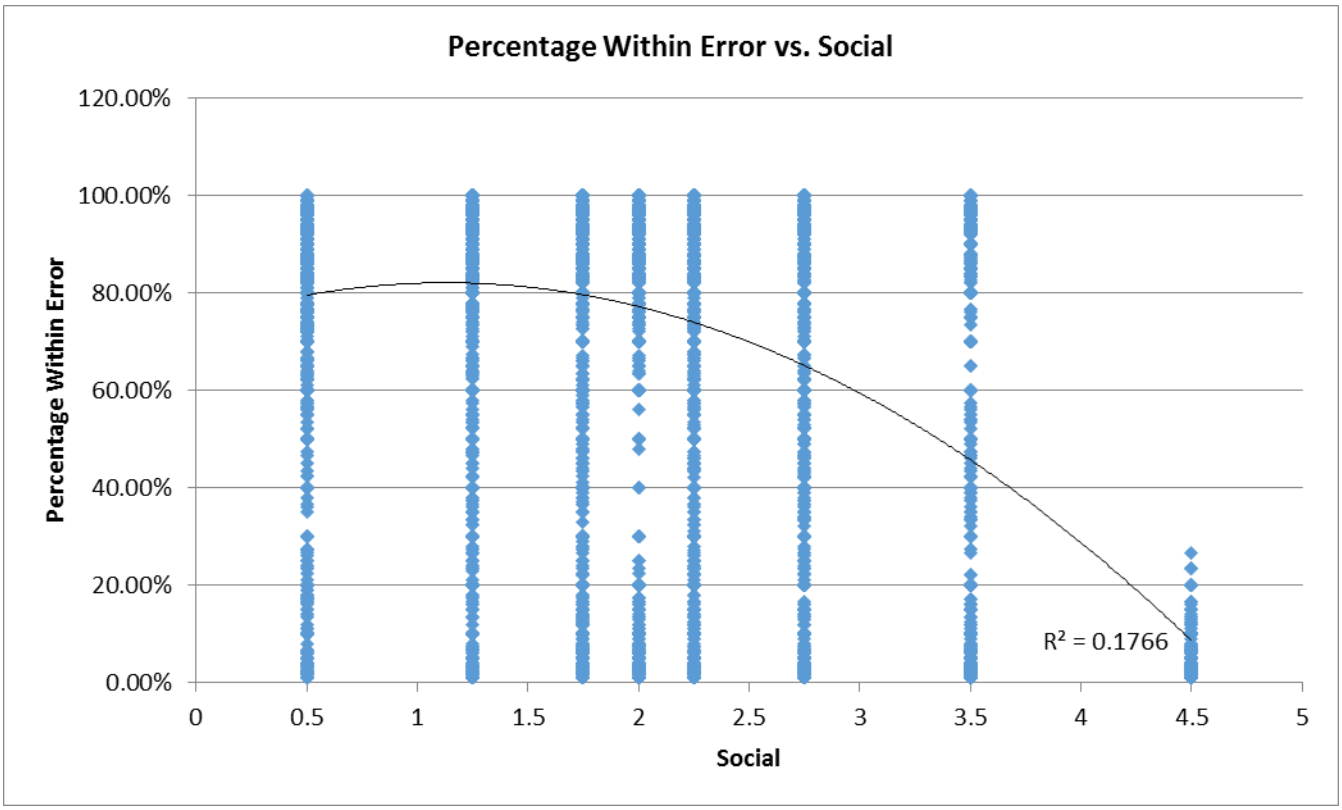

Number of Epochs vs. Problem Number
Number of Epochs
Problem Number
R² = 0.0013
0
200
400
600
800
1000
1200
0
0.5
1
1.5
2
2.5

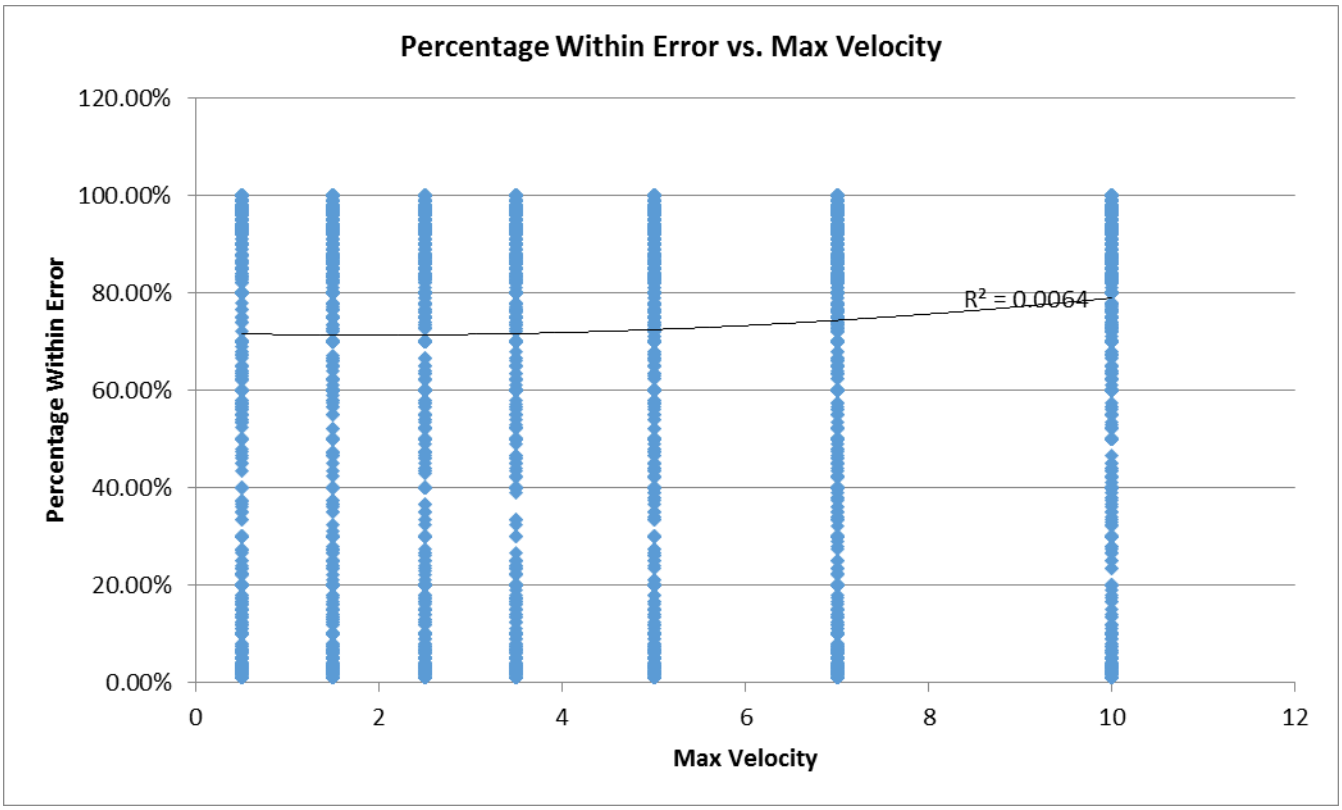

Number of Epochs vs. Max Velocity
Number of Epochs
Max Velocity
1200
1000
800
600
400
200
0
0
2
4
6
8
10
12
R² = 0.0184

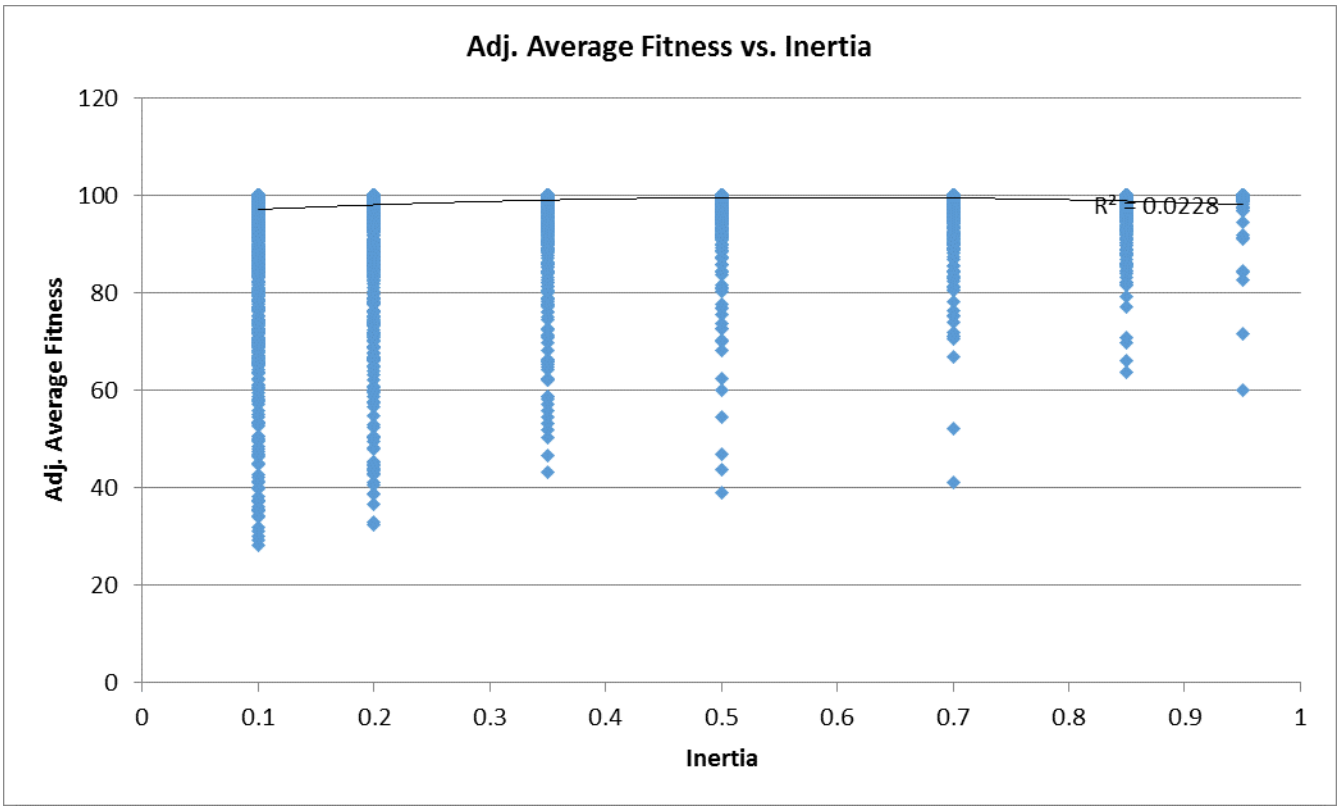

Percentage Within Error vs. Number of Particles
Percentage Within Error
Number of Particles
120.00%
100.00%
80.00%
60.00%
40.00%
20.00%
0.00%
0
20
40
60
80
100
120
R² = 0.0022

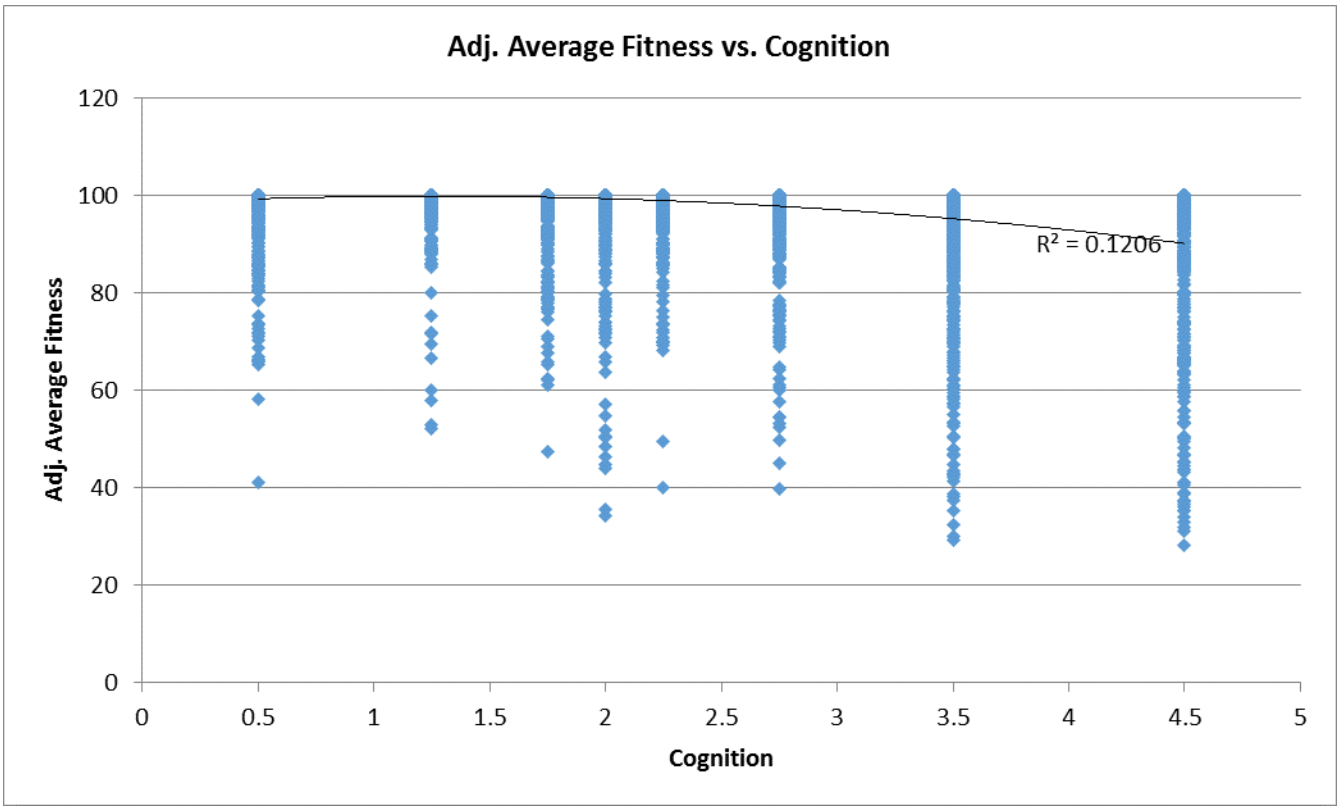

Percentage Within Error vs. Inertia
120.00%
100.00%
80.00%
60.00%
40.00%
20.00%
0.00%
Percentage Within Error
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Inertia
R² = 0.0368

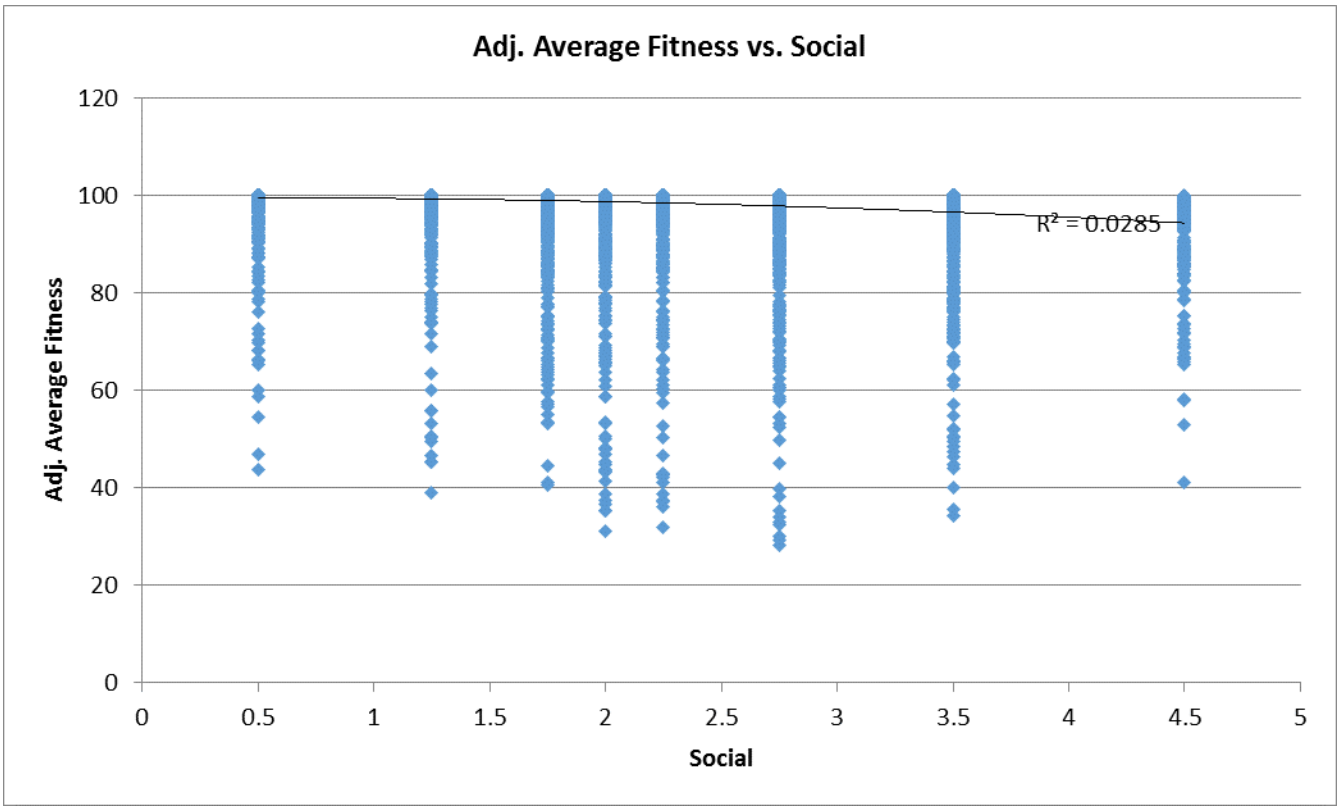

Percentage Within Error vs. Cognition
120.00%
100.00%
80.00%
60.00%
40.00%
20.00%
0.00%
Percentage Within Error
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
Cognition
R² = 0.215

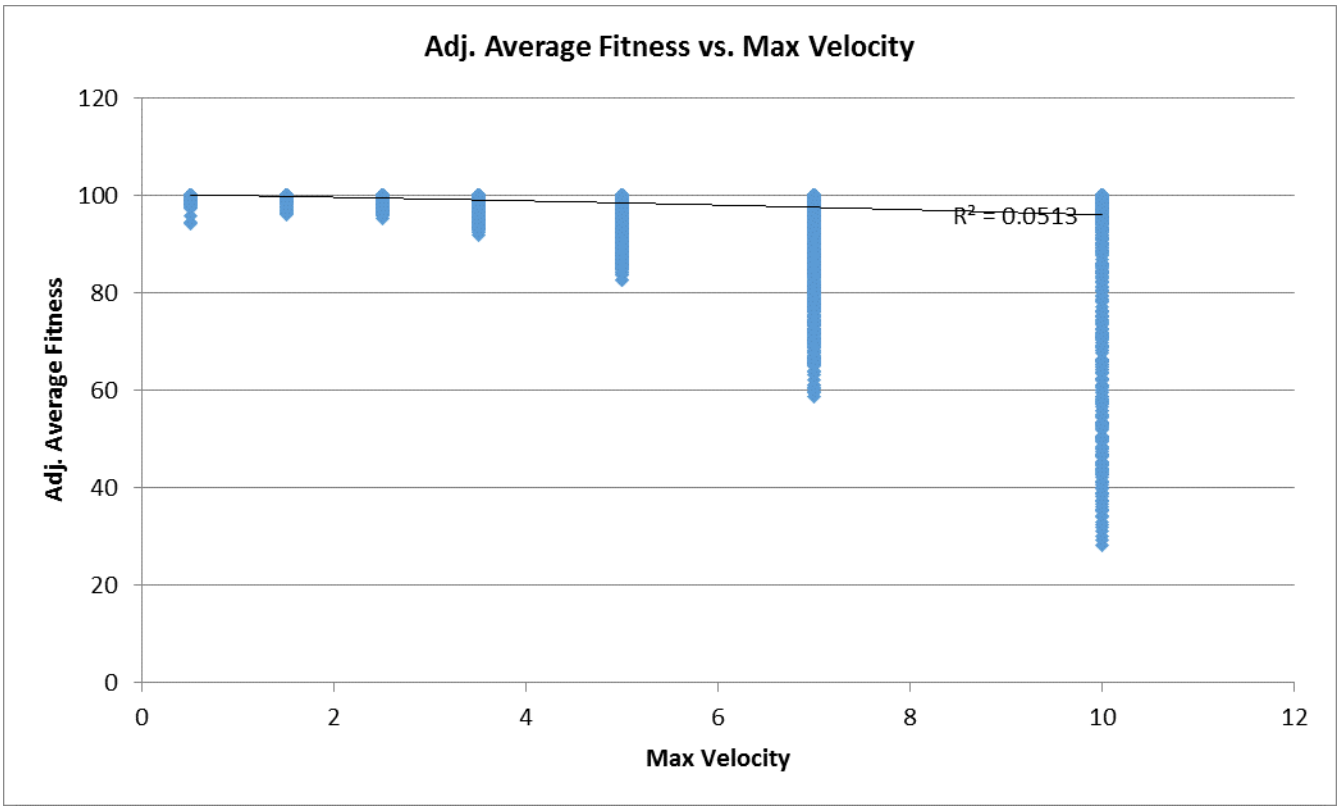

Percentage Within Error vs. Social
120.00%
100.00%
80.00%
60.00%
40.00%
20.00%
0.00%
Percentage Within Error
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
Social
R² = 0.1766


Percentage Within Error vs. Max Velocity
120.00%
100.00%
80.00%
60.00%
40.00%
20.00%
0.00%
Percentage Within Error
0
2
4
6
8
10
12
Max Velocity
R² = 0.0064


Adj. Average Fitness vs. Inertia
Adj. Average Fitness
Inertia
R² = 0.0228
120
100
80
60
40
20
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Adj. Average Fitness vs. Cognition
Adj. Average Fitness
Cognition
R² = 0.1206
120
100
80
60
40
20
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

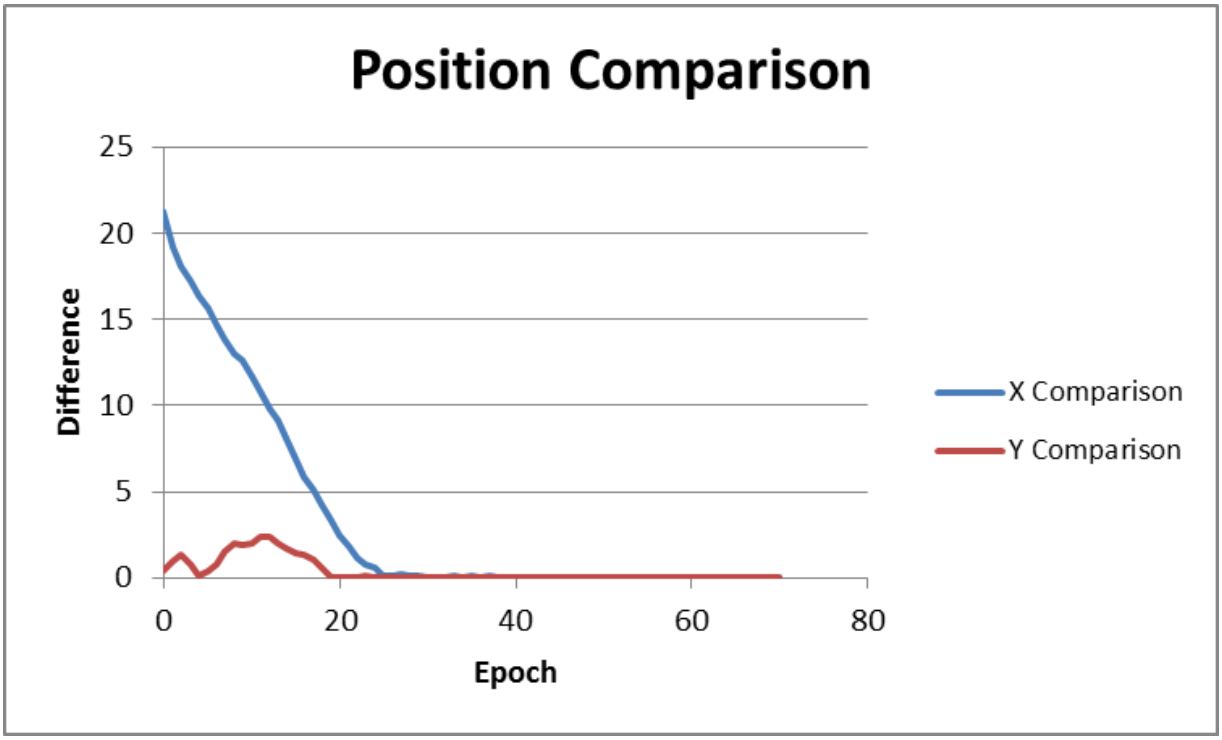

Adj. Average Fitness vs. Social
Adj. Average Fitness
Social
R² = 0.0285
120
100
80
60
40
20
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


Adj. Average Fitness vs. Max Velocity
Adj. Average Fitness
Max Velocity
R² = 0.0513
120
100
80
60
40
20
0
0
2
4
6
8
10
12

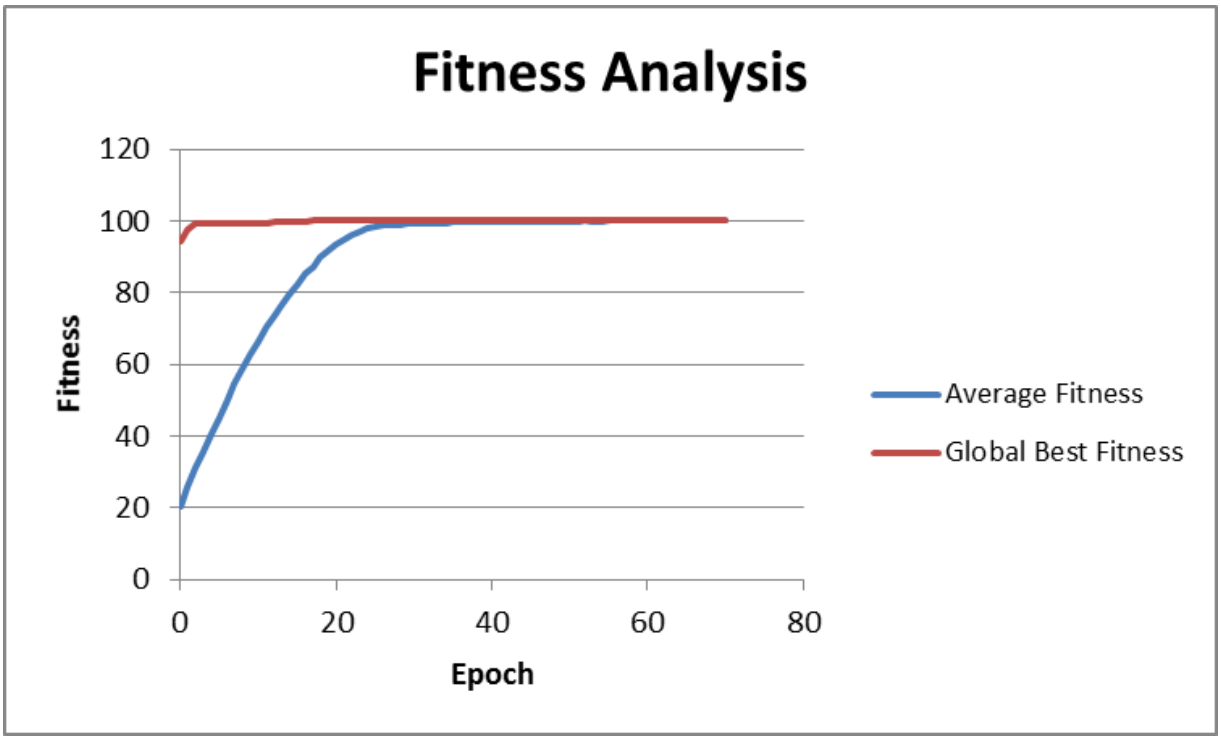
The first hand-picked configuration used the following variables:

Number of Epochs	1000
Number of Particles	40
Inertia	0.5
Cognition	2
Social	1.75
Problem Number	1
Maximum Velocity	2.5

This configuration represented a fairly average run, with the particles converging after 70 epochs. Below is a graph of the difference between the Average X and Y values of the swarm and the Best X and Y values of the swarm against the number of epochs.



Below is a graph of the Average Fitness and Global Best Fitness for the same simulation run.

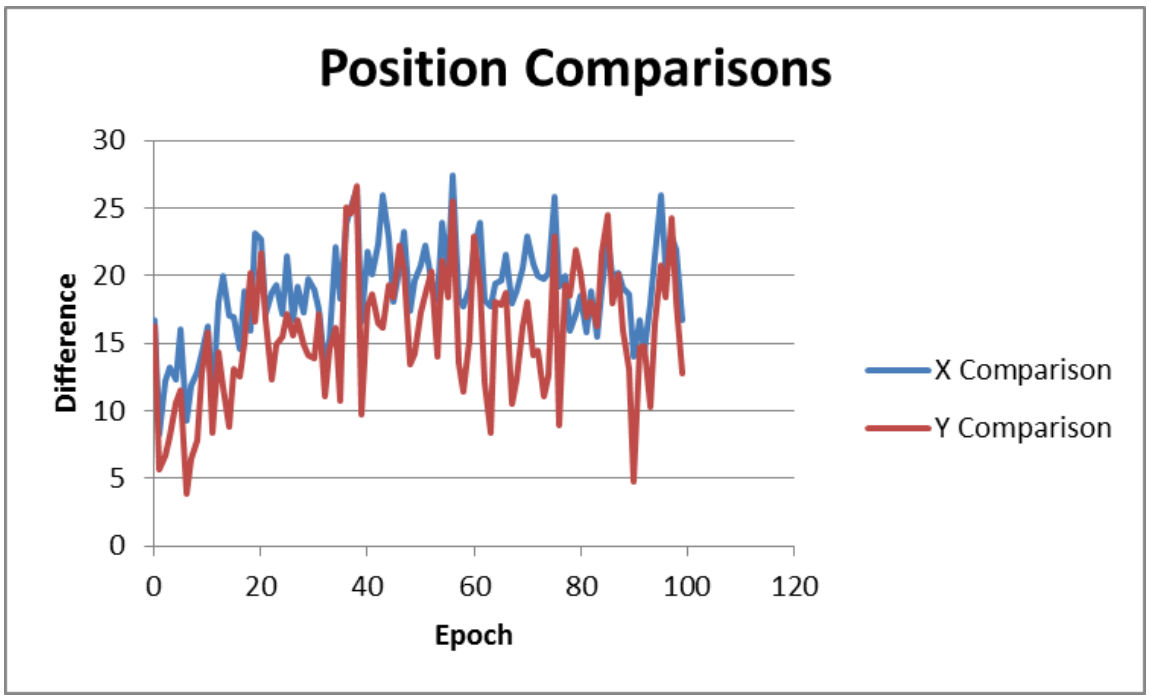


Finally, an animation of this simulation run can be seen here.

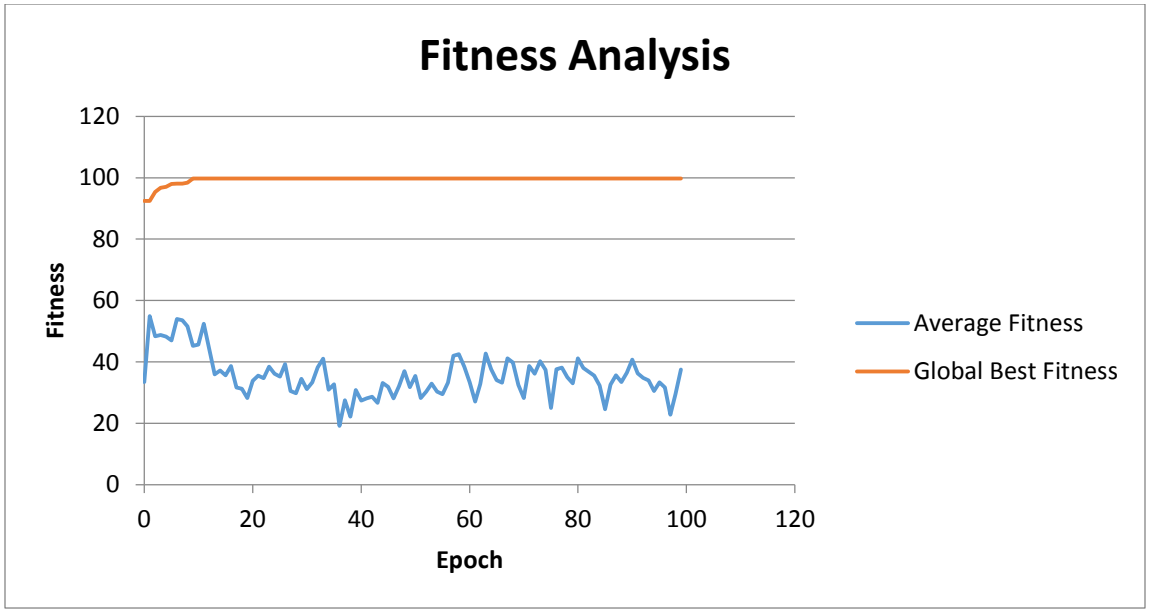
As for the second configuration, we used the following variables:

Number of Epochs	100
Number of Particles	40
Inertia	0.2
Cognition	2.75
Social	3.75
Problem Number	1
Maximum Velocity	10

This configuration represented a case that did not converge, and thus we limited the animation to show only 100 iterations. Below is the plot of Average X and Y values of the swarm and the Best X and Y values of the swarm against the number of epochs.



Below is a graph of the Average Fitness and Global Best Fitness for the same simulation run.

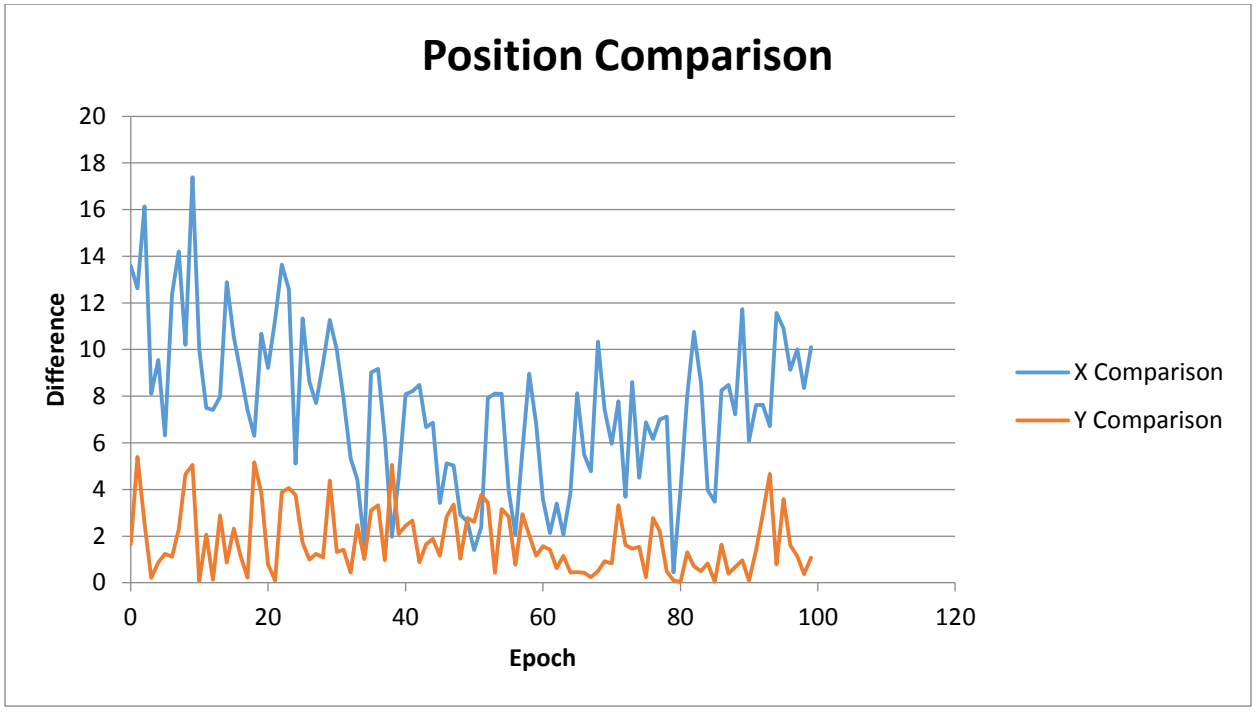


Finally, an animation of this simulation run can be seen here.

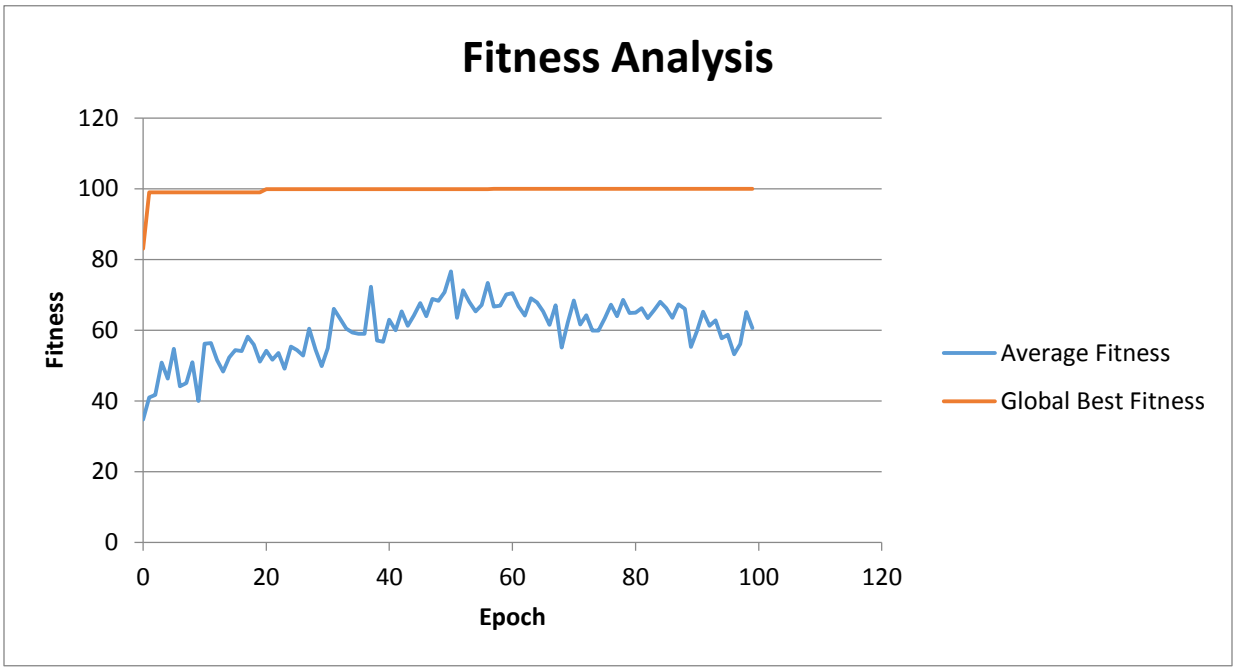
Our third configuration used the following variables:

Number of Epochs	100
Number of Particles	40
Inertia	0.95
Cognition	0.5
Social	1.75
Problem Number	1
Maximum Velocity	7

This configuration represented another that did not converge before 1000 epochs. Again, the animation was limited to first 100 epochs. Below is a graph of the difference between the Average X and Y values of the swarm and the Best X and Y values of the swarm against the number of epochs.



Below is a graph of the Average Fitness and Global Best Fitness for the same simulation run.

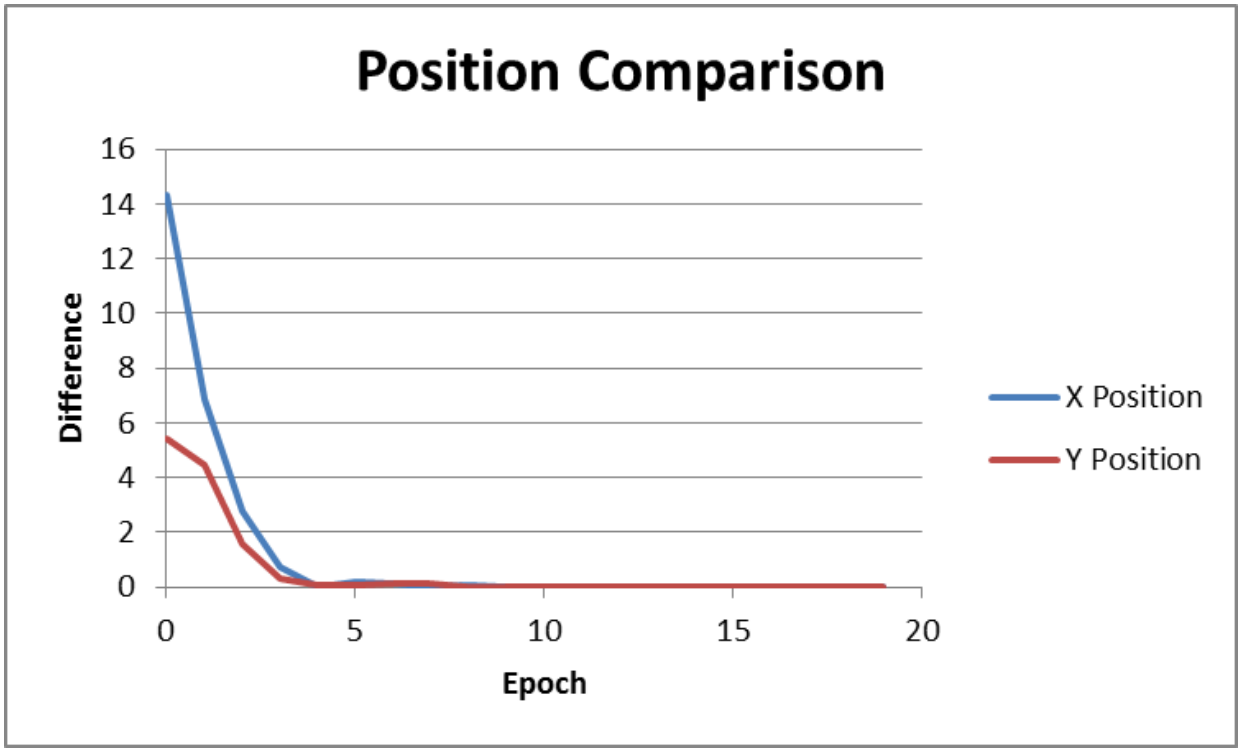


Finally, an animation of this simulation can be seen here.

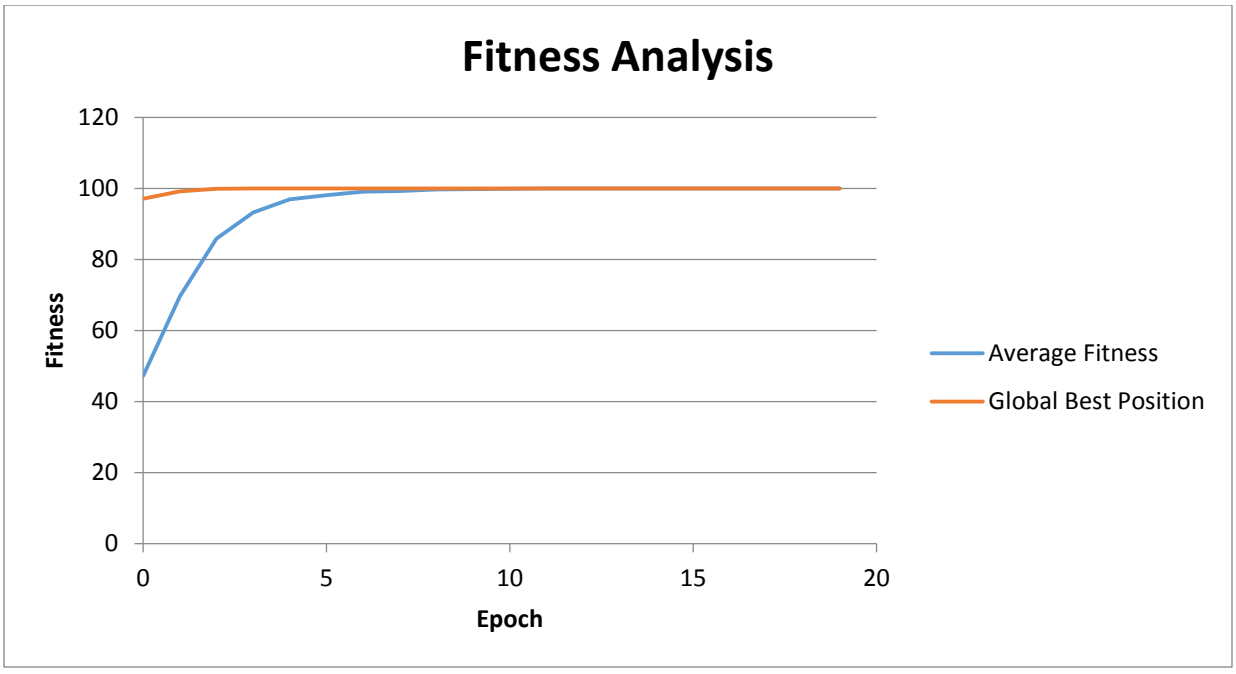
Our final configuration used the following values:

Number of Epochs	100
Number of Particles	100
Inertia	0.1
Cognition	2
Social	1.25
Problem Number	1
Maximum Velocity	7

This configuration was the best performing configuration of all, with the particles converging within 19 epochs. Below is the plot of Average X and Y values of the swarm and the Best X and Y values of the swarm against the number of epochs.



Below is a graph of the Average Fitness and Global Best Fitness for the same simulation run.



Finally, an animation of this simulation run can be seen here.

Discussion and Conclusions:

Below is a discussion of each of the input variables, and its effect on the fitness of the swarm over time:

Number of Particles: As seen in the correlation tables and the graphs, the number of particles in the swarm did not affect the fitness of the swarm.

Inertia: The inertia value represented how fast a particle could change direction. A higher value meant that a particle was more likely to fly past a more fit position. A low value meant that the particle was too easily swayed by quick changes in the global best. As seen in the correlation tables, the inertia had a moderate effect on the fitness of the swarm. Both a small and large inertia modifier resulted in a greater number of epochs needed to converge. It can also be seen in all the graphs that the most optimal value for inertia is around .5.

Cognition Modifier: The cognition modifier represented how much the particle tended to drift towards its own best X and Y values. A higher cognition modifier meant the particle might not be swayed enough by the global best, as seen in the second hand-picked configuration. As seen in the correlation table, the cognition modifier was moderately influential in the fitness of the swarm. As seen from the graphs, a cognition modifier between 1.5 and 2 was optimal.

Social Modifier: The social modifier represented how much the particle tended to drift towards the global best X and Y values. A higher social modifier meant the particle might be swayed too much be the global best values and not enough by its own best values. As seen from the correlation table, the social modifier was moderately influential in the fitness of the swarm. As seen from the graphs, a social modifier between 1.5 and 2 was optimal.

Maximum Velocity: The maximum velocity simply limited the speed of each particle. Generally, a higher maximum velocity was better, as it allowed the particles to converge on a point more quickly. However, if the particles also had a high inertia modifier, then the particles were at risk of overshooting a more fit value.

Below is a discussion of each output variable, and its significance.

Number of Epochs: The number of epochs was a good representation of the fitness of the swarm. If the swarm converged within a small number of epochs, it mean that the swarm was well optimized. If the swarm did not converge within 1000 epochs, it was unlikely to ever converge.

Percentage within Error: The percentage within error was a good factor to keep track of when measuring fitness, as it determined how tightly the particles were converging on a solution. The more tightly packed the particles, the more precise the solution could be considered.

Average Fitness: The average fitness of the swarm was another good measure. The higher the average, the more fit the swarm, as dictated by the fitness functions, Problem 1 and Problem 2.

Adjusted Average Fitness: The adjusted average fitness was arguably more important than the average fitness when considering more than one problem. This is because the maximum fitness of the second problem was higher than the maximum fitness of the second problem. The adjusted fitness scaled both values to be out of 100%. This allowed us to see how influential different values really were.

Global Best Fitness: Finally, the fitness of the most fit particle was not a good indicator of the fitness of the swarm or the accuracy of the solution. As can be seen in the correlation table, the global best fitness was not much effected by the input variables, as one particle was not necessarily indicative of the entire swarm.