Project 5: Particle Swarm Optimization

Chris Mobley COSC 420

Due: 4/24/2020

1 Introduction

Particle swarm optimization (PSO) is an optimization technique based on the behaviors of flocks of birds or schools of fish. This method of optimization uses a population of particles to search the space of solutions to its given fitness function to find its global maximum value. In each iteration of the population, the velocities of particles are determined based on their current velocity, a particles personal best solution, and the population's best solution so far. The particles' positions are then updated using this velocity value, and the personal bests and global best positions are updated. The iteration of this population slowly steps the population towards some local maximum of the solution space (ideally the global maximum). This project aims to explore PSO's and how their performance is affected by the input parameters of population size, inertia, cognitive weight, and social weight.

2 Theory

Each particle has some position in this search space representing the inputs to the population's fitness function. The main idea of PSO is that the population of particles will be able to move their position around the search space to find the global maximum to their fitness function. The particles move throughout the space by updating their velocity using an acceleration function and using this new velocity to then update their position. The acceleration function updates a particle's velocity based on three factors: current velocity, distance to the particle's personal best solution, and distance to the population's best solution.

The current velocity of a particle represents the current search direction of the particle. The acceleration takes this into account, dampened by some value of inertia. This inertia value represents the particle's "desire" to continue its current search path.

The distance to the particle's personal best solution is used to push a particle's velocity towards its best solution so far. This acceleration is weighted by the cognition parameter of the particle and represents the individual's trust of its personal experience.

The distance to the population's best solution is used to push a particle's velocity towards the global best solution so far. This acceleration is weighted by the social parameter of the particle and represents the individual's trust of the total population's experience.

Both the cognition and social parameters of the particles are randomly set between zero and some maximum value on each epoch. This allows individuals to try out different strategies of searching the space of solutions.

The final factor influencing the effectiveness of the swarm is its population size. As each particle's initial position is random, a large population size would increase the chances that a particle would be initialized near the search space's global maximum. Large populations come with a downside, however, in that large numbers of particles mean that it takes longer for every particle to converge on the global maximum. In addition, large populations are more computationally intensive to run.

3 Methods & Calculations

To determine the effectiveness of the network with its given input parameters, each parameter set was run until it either converged on a global best solution or it reached a max epoch number of 1000. Convergence was determined by checking if the root mean square divergence of the population to the global best solution was below an epsilon of 0.01. Each parameter set was run for 10 trials, and the population's average performance was determined based on epochs required for convergence, percent of particles converged, and the average distance in the X and Y dimensions of particles to the global best solution.

The population was initialized with a random position in the search space (bounds of -50 to 50 for both X and Y dimensions) for every particle and their velocities were set to zero at the start.

Updating the Particles

In each epoch the positions of all particles must be updated based on their velocity. The velocity for each epoch is determined by Eq. 1.

$$\vec{v}_i(t+1) = I * \vec{v}_i(t) + c_1 * r_1 * \left(\vec{b}_{p,i} - \vec{p}_i(t)\right) + \ c_2 * r_2 * (\vec{b}_g - \ \vec{p}_i) \tag{1}$$

Where $\vec{v}_i$ is a particle's velocity, $\vec{p}_i(t)$ is a particle's position at time t, $\vec{b}_{p,i}$ is a particle's personal best position, $\vec{b}_g$ is the global best position, I is the inertia of the system, c_1 and c_2 are the max cognitive and social parameters of the system, and r_1 and r_2 are random numbers in the range of [0, 1).

In order to keep positions within reasonable bounds and prevent the system from behaving wildly, the velocity of particles must be scaled within the bound of its maximum velocity. This is done with Eq. 2 using the new velocity value.

$$\vec{v}_i = \begin{cases} \frac{v_{max}}{\|\vec{v}_i\|} * \vec{v}_i & \|\vec{v}_i\| > v_{max} \\ \vec{v}_i & otherwise \end{cases} \tag{2}$$

Where v_{max} is the maximum velocity of a particle.

The new position of a particle is then found using this new velocity value using Eq. 3

$$\vec{p}_i(t+1) = \vec{p}_i(t) + \vec{v}_i(t) \tag{3}$$

After updating all the particle's positions, their personal best position is updated with their position if $Q(\vec{p}_i) > Q(\vec{b}_{p,i})$. Where $Q(p)$ is the fitness function of some position, p, in the search space. In addition, the global best is updated with a particle's position if $Q(\vec{p}_i) > Q(\vec{b}_g)$.

Stopping Condition

The root mean square divergence ($RMSD$) of the particle's position compared to the global best position was used to determine if the population had converged to a specific point. This was done in the X and Y dimensions independently with Eq. 4 and 5.

$$RMSD_x = \frac{\sum_i (p_{x,i} - b_{g,x})^2}{2 * N} \tag{4}$$

$$RMSD_y = \frac{\sum_i (p_{y,i} - b_{g,y})^2}{2 * N} \tag{5}$$

Where N is the size of the population.

When the RMSD for both the X and Y dimensions gets below a threshold of 0.01, the program stops its iteration.

Fitness Functions

In this project, two fitness functions, Q_1 and Q_2, were used. For these functions, intermediate values $mdist$, $pdist$, and $ndist$ must be found using Eq. 6, 7, and 8.

$$mdist = \frac{\sqrt{max_x{}^2 + max_y{}^2}}{2} \tag{6}$$

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

$$ndist = \sqrt{(p_x + 20)^2 + (p_y + 7)^2} \tag{8}$$

Where max_x and max_y are the maximum boundaries of the search space and p_x and p_y are the X and Y components of a particle's position.

Problem 1's fitness function is defined as:

$$Q_1(p_x, p_y) = 100 * \left(1 - \frac{pdist}{mdist}\right) \tag{9}$$

Problem 2's fitness function is defined as:

$$Q_2(p_x, p_y) = 9 * \max(0, 10 - pdist^2) + 10 * \left(1 - \frac{pdist}{mdist}\right) + 70\left(1 - \frac{ndist}{mdist}\right) \tag{10}$$

Problem 1 has one maximum at (20, 7), and Problem 2 has two maxima: one at (20, 7) (a global maximum) and one at (-20, -7) (a local maximum).

4 Results

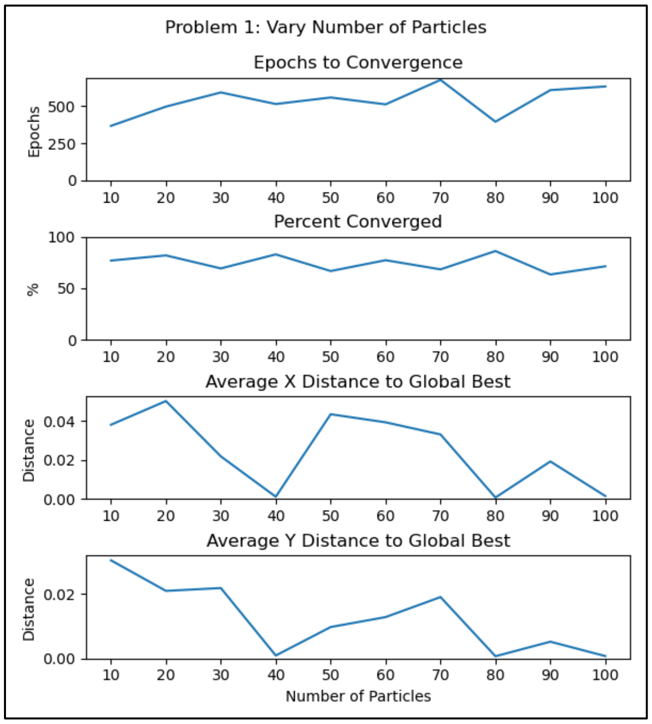

Figure 1. Graphs of performance of Problem 1 for varied population size, N=[10, 100] in increments of 10 (I=1.0, c_1=c_2=2)

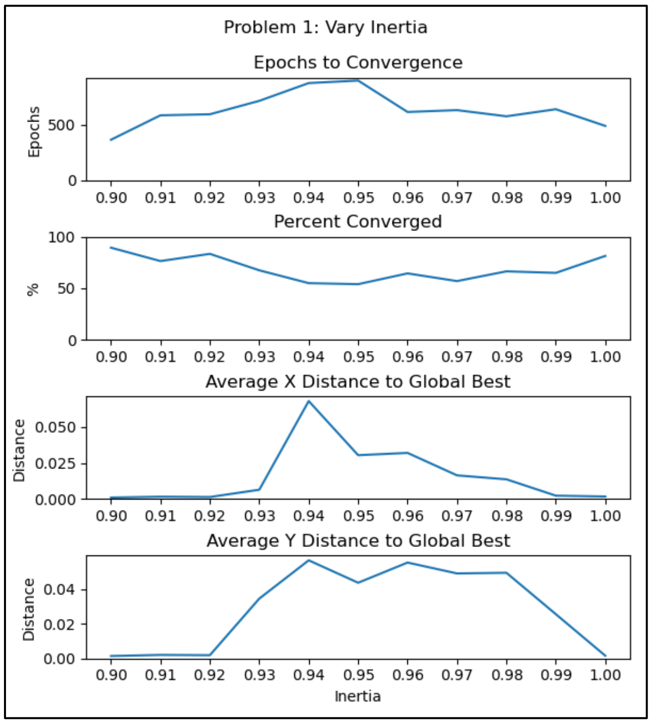

Figure 2. Graphs of performance of Problem 1 for varied inertia value, I=[0.90, 1.00] in increments of 0.01 (N=20, c_1=c_2=2)

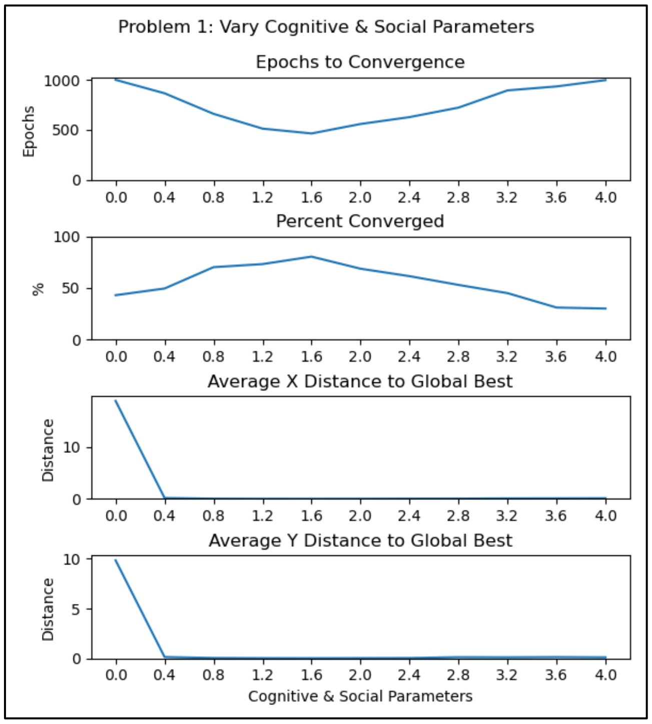

Figure 3. Graphs of performance of Problem 1 for varied cognitive and social parameters, c_1=c_2=[0.0, 4.0] in increments of 0.4 (N=20, I=1.0)

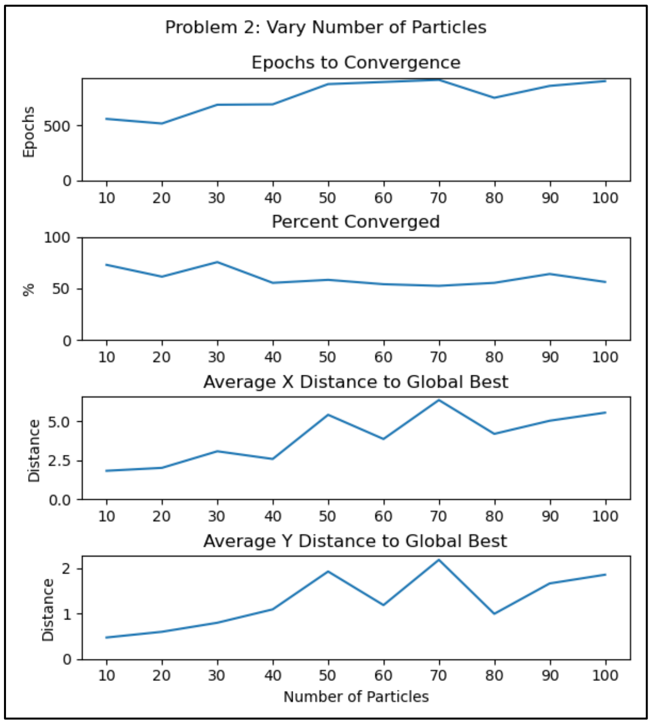

Figure 4. Graphs of performance of Problem 2 for varied population size, N=[10, 100] in increments of 10 (I=1.0, c_1=c_2=2)

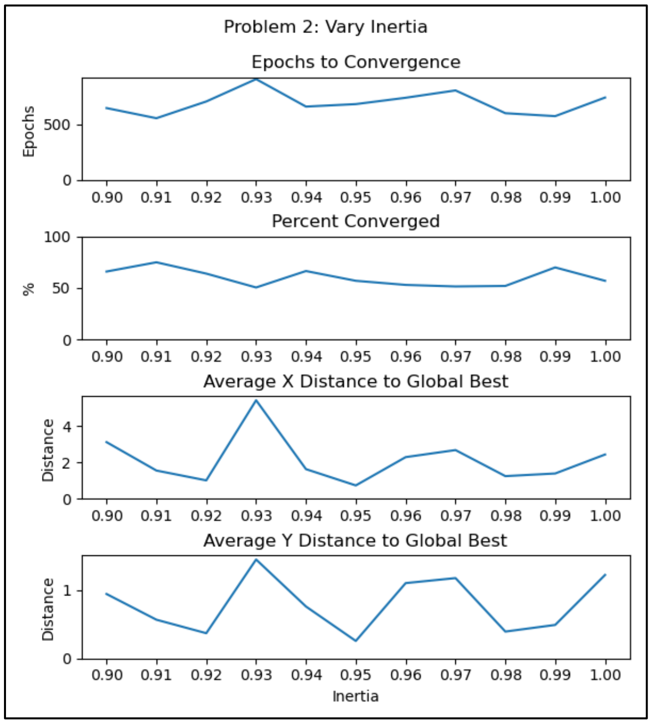

Figure 5. Graphs of performance of Problem 1 for varied inertia value, I=[0.90, 1.00] in increments of 0.01 (N=20, c_1=c_2=2)

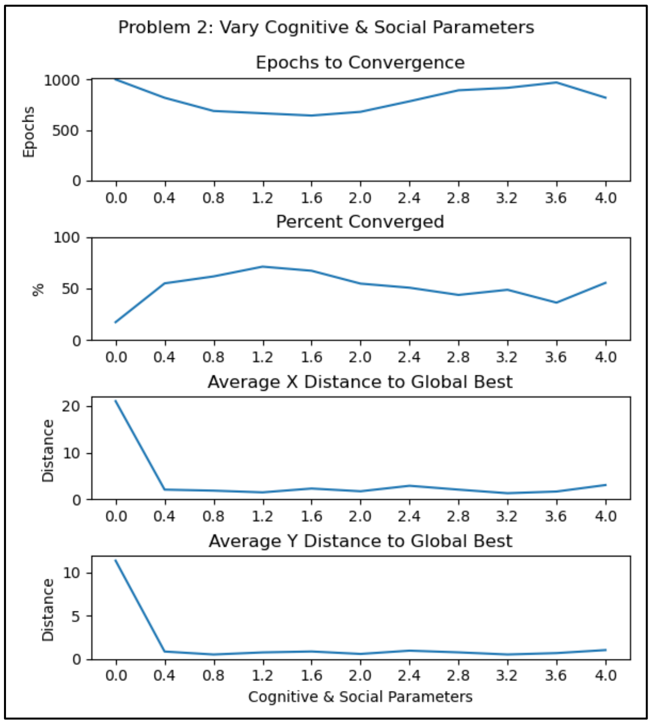

Figure 6. Graphs of performance of Problem 2 for varied cognitive and social parameters, c_1=c_2=[0.0, 4.0] in increments of 0.4 (N=20, I=1.0)

5 Discussion

The overarching goal of this experiment was to determine the effects that different input values would have on the behavior and performance of a basic particle swarm optimization (PSO) problem. To do this, population size, inertia, and a combination of the cognitive and social parameters were varied while keeping the other inputs constant. This was done to isolate each input parameter and study its effects independently.

The first problem tested with the PSO system was fitness function Q_1. This was the simplest of the two problems tested, only having a single maximum value at (x=20, y=7). As a result, the PSO had a significantly easier time in finding the global maximum of the searching space as there was not any competition between different local maximums.

Varying the population size in Problem 1 showed interesting results indicating that there is an ideal number of particles to be used for this problem. As seen in Figure 1, the number of epochs required for convergence and the percent of particles converged remain steady at around 500 epochs and 80%, however, the average distance to the global best varies widely depending on the population size. The average distance graphs for the X and Y dimensions show a significant drop at a population size of N=40 and N=80 indicating these are ideal candidates of population size for this problem.

Varying the inertia in Problem 1 showed that there was a loss in performance of the system when the inertia value was $\sim$0.95. This can be seen in Figure 2 as a drop in percentage of converged particles around this value as well as a large increase in the average distances to global best. In addition, it appears the ideal inertia value for this problem is around 0.90 as this is the point when the percentage of converged particles was at its highest and the epochs to convergence and the average distances to global best were at their lowest.

Varying the cognitive and social parameters also showed that picking correct values can have a significant effect on the performance of the system. As seen in Figure 3, the best performance of the system was around a value of 1.6 for both parameters as indicated in the drop in epochs to convergence and increase in percent converged. When these values were set to 0.0, the system was not able to work at all. Since the velocity of the particles was initialized to zero, without cognitive or social influence to move, the system would simply just stay stagnant. The poor performance of the system at high values indicates that the system was behaving too erratically. This is because the particles are so heavily influenced by their personal best and the global best positions that the inertia factor of their velocity cannot keep them on track in searching in new areas.

The second problem tested with the PSO system was fitness function Q_2. This function has two local maximums at (x=20, y=7) and (x=-20, y=-7) with the first being the global maximum of the space. The competition between these two local maximums resulted in the PSO system having a much harder time converging on the global maximum.

Varying the population size for Problem 2 illustrated how it is hard for a PSO system to converge when there are too many particles. This can be seen in an overall increase in epochs needed for conversion in Figure 4. This challenge arises because the chances of some particles becoming "stuck" in local maximums is increased. In addition, the larger number of particles means that it will take longer for enough particles to "escape" their local maximum and make it to the global maximum so that the system can converge within its stopping threshold. As seen in Figure 4, a population size of 10 to 20 particles would work best for this problem.

Varying inertia for Problem 2 showed some increased performance at a few inertia values, as seen in Figure 5. While the percent converged values remained around only 50% across the inertia values tested, some inertia values provided better average distances to the global best position. This is because these values provided enough dampening to the particle's velocity so that they could home in on a specific point without overshooting too much. These average distance graphs indicate the best inertia value for this problem would be around 0.95.

Varying the cognitive and social parameters for Problem 2 showed increased performance in the range of around 1.2 to 1.6. This can be seen in the drop of epochs to convergence as well as increase in percent converged graphs of Figure 6. Like in the last problem, a value of 0.0 for these parameters produced a stagnant system. The improvement in the performance with values of 1.2 to 1.6 indicate that this is where the particles are given enough freedom of movement to explore their area while still being able to escape local maximums to make it to the global best position for convergence.

Overall, it appears that the parameters which had the largest effect on the PSO system were the population size, cognitive parameter, and social parameter. One issue with the testing method used for this experiment was that all other input parameters were kept constant through the trials. In future experiments it would be interesting to see if different combinations of inputs could produce better results.