CS 420 Lab: Particle Swarm Optimization Analysis

Introduction:

In this report, I investigated the effects of various parameters on the performance of the PSO algorithm for two classic optimization problems of the Rosenbrock function and the Booth function. The goal was to understand how different settings of parameters like the number of particles, inertia, cognition parameter, and social parameter affects the convergence behavior and solution quality in PSO.

Setting up the experiment:

We were supposed to explore two optimization problems described below:

1. Rosenbrock function: f(x, y) = (1 - x)^2 + 100(y - (x^2))^2
2. Booth function: f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2

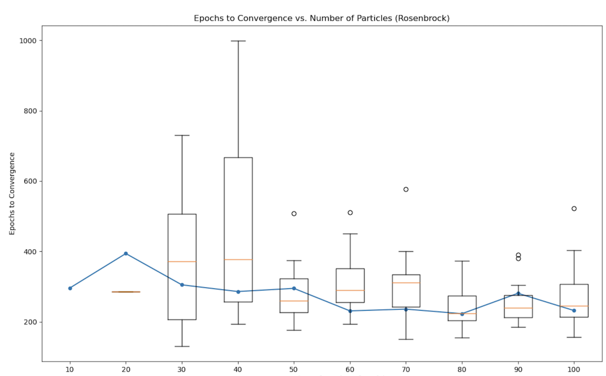
I studied the following parameters:

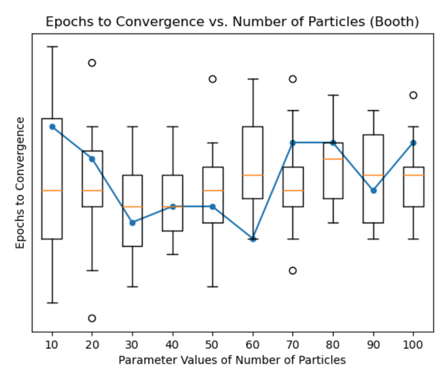
1. Number of particles: 10 to 100 with increments of 10
2. Inertia: 0.1 to 1 with increments of 0.1
3. Cognition parameters: 0.1 to 4 with increments of 0.1
4. Social parameters: 0.1 to 4 with increments of 0.1

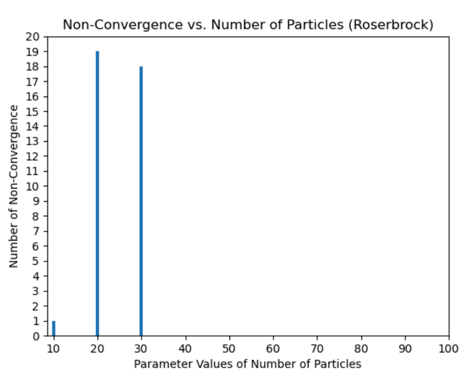
For this analysis, I implemented a script to run with the pso.py code that we were given. The algorithm terminated either upon convergence or after a maximum of 1000 epochs. For each run, I recorded the number of epochs to convergence, the solution that I found, and its fitness values.

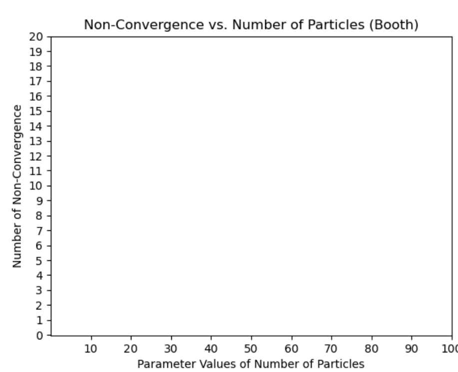
Graphs:

Number of particles:

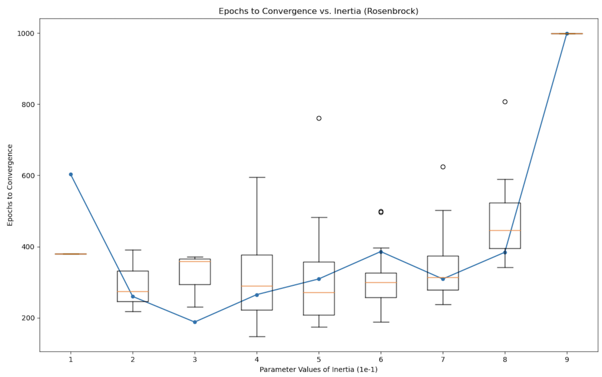


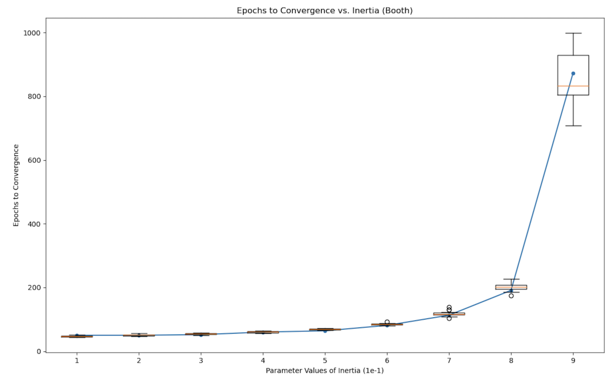


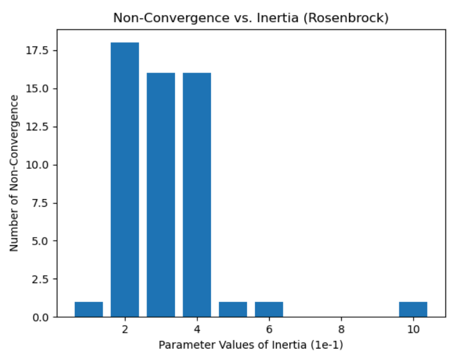


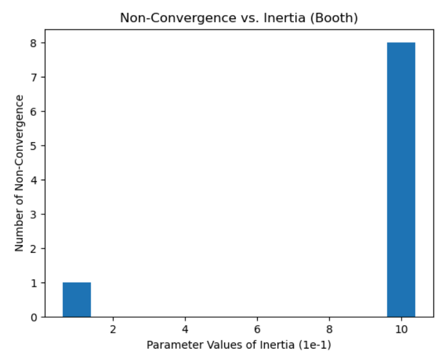


Inertia:

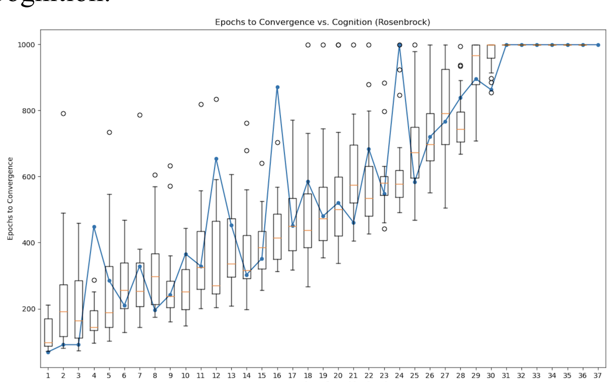


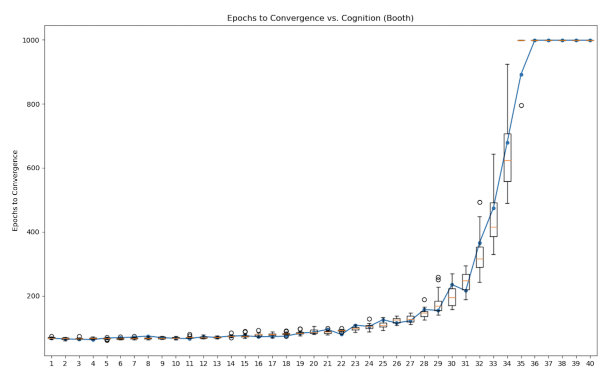


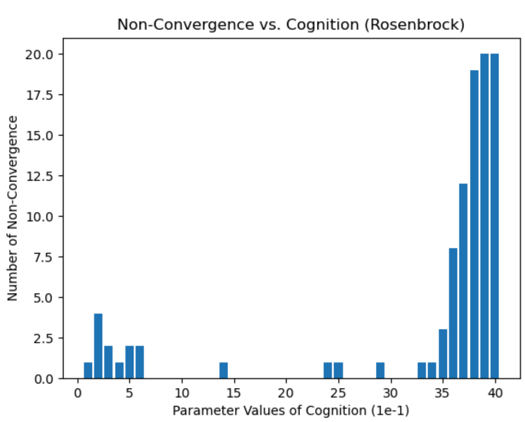


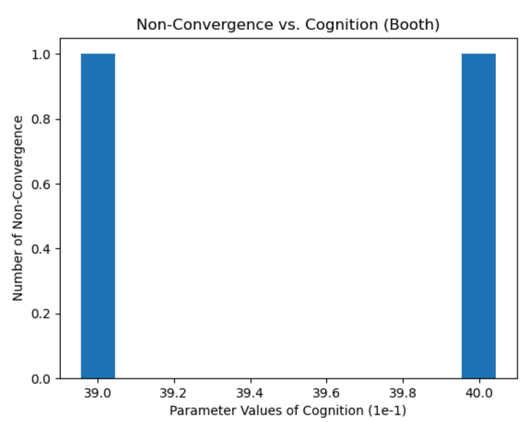


Cognition:

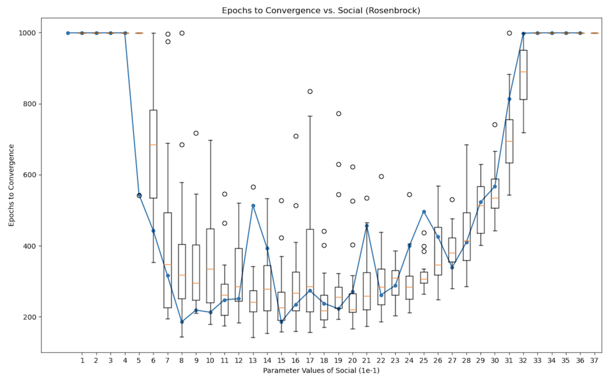


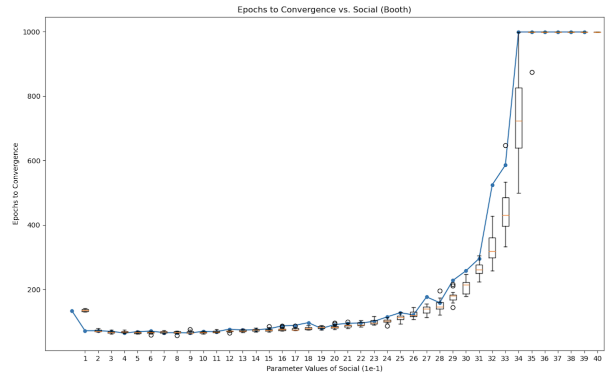


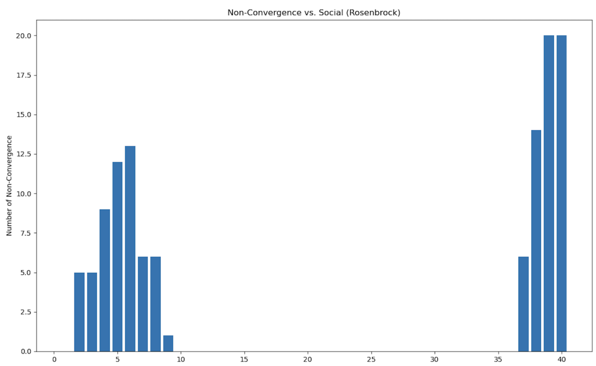


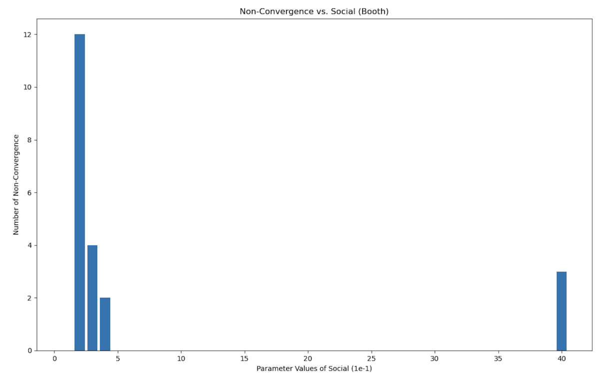


Social:









For the number of particles, the graph shows us the relationship between the number of particles and the convergence speed of the PSO algorithm. As the number of particles increases the convergence speed generally improves but some things it does not. This shows us that increasing the swarm size can help improve exploration-exploitation balance, which leads to faster convergence.

For the inertia graph, it shows us the impact of the inertia parameter on PSO algorithm performance. It was interesting to see an optimal range of inertia which was kind of around 0.5 where convergence is fastest. I also learned that lower inertia values lead to more exploration, while higher values favor exploitation, but excessively high inertia can hinder the convergence.

For the cognition parameter, I saw the imbalance between individual and local best positions' influence on particle movement. It was interesting to see that performance tends to improve with moderate cognitive values, which allows particles to explore both individual and local best solutions in the best way possible.

For the social parameter, we can see that they affect the influence of the global best position on particle movement. In the graphs, the optimal convergence is often observed within a moderate range of social parameter values. Low social values emphasize individual exploration, while high values emphasize swarm convergence. Which potentially leads to premature convergence or exploration stagnation.

Discussion:

How big of an impact can parameter selection have on performance?

If we think about the impacts of parameter selection on performance then we can safely say that it affects the performance significantly. Parameter selection for sure has a significant impact on the performance of the PSO algorithms as the choosing of parameters directly influences the exploration and exploitation capabilities of the algorithm which affects the convergence speed, solution quality, and robustness. In this analysis, I observed notable variations in convergence behavior and solution quality across different kinds of parameter selections. An example could be when examining the number of particles. For the number of particles, increasing the particle count usually improves convergence speed, but we can also see diminishing returns beyond a certain threshold. In a similar way, variations in inertia or cognition or social parameters resulted in distinct convergence patterns and solution qualities. We also learned that suboptimal parameter choices could lead to a slower convergence rate or even sometimes cause failure to convergence to the correct solution as it can be seen in the non-convergence graphs. This highlights the important role of parameter selection in PSO algorithm performance.

Do there appear to be optimal values for each parameter? Are the “best” performing values (i.e., those that converge the fastest) the same for each problem?

Optimal parameter values for PSO algorithms are very interesting as they might vary depending on the characteristics of the optimization problem being solved. Like in our analysis, we found that while certain parameter values tended to perform in a much better way across different problems than others, the optimal setting could differ between the Rosenbrock and Booth functions. Let us take an example, like for instance moderate values around the default settings, an example could be 0.5 for inertia, 1 for cognition and social parameters, often showed us good performance across both problems. But, the specific values that converged the fastest varied quite a lot depending on the problem's complexity. Like some parameters may require a bit of tuning to help them adapt to the unique features of each problem, but that shows us that a one-size-fits-all kind of approach might not be the best optimal for PSO parameter selection.

Why do some values perform better than others?

This is a hard question to answer because the performance of parameter values in PSO is influenced by the ability of the algorithm to balance exploration and exploitation. To best answer this question let us talk about some factors that cause the performance to be different.

So first things first are the parameters like inertia, cognition, and social parameters which basically control the balance between exploration and exploitation. Optimal parameter values maintain a balance between these two aspects which allows the algorithm to efficiently explore the search space while exploiting promising solutions.

The optimal parameter values do depend on the characteristics of the optimization problem, such as its dimensionality, topology, and the presence of local optima. Different problems might require us to use different exploration and exploitation methods to navigate their solution landscapes effectively.

Some parameter values may lead to faster convergence by promoting a rapid exploitation of promising regions, while others may facilitate better exploration by encouraging diverse search trajectories. But there is one important thing, so the effectiveness of parameter values is often evaluated based on their ability to guide the algorithm towards the global optimum efficiently.

Robust parameter values make sure that that the algorithm can adapt to variations in the problem landscape and initial conditions. Values that are too extreme may lead to premature convergence or poor convergence behavior in certain scenarios.

Basically, the overall performance of parameter values in PSO reflects their influence on the algorithms’s search behavior and also its ability to find complex optimization landscapes effectively. And because of that run experiments and analyzing them is important for identifying the optimal parameter values tailored to a specific optimization problem.

Conclusion:

After doing these experiments and analyzing the results, I understand the importance of parameter selection in the performance of PSO algorithms for any kind of optimization problems. It was fun to learn about the impact of parameters and identifying optimal values that can be chosen for specific problems. We can improve the effectiveness and efficiency of PSO in solving complex optimization problems.