A Comparison of Neighboring Topologies in Particle Swarm Optimization

Ethan Zhou, Jake Adicoff, and Mac Groves

Abstract: In this paper, we compare the effectiveness of certain *neighborhood topologies* in the performance of Particle Swarm Optimization. A *neighborhood topology* determines the set of particles in a swarm that can influence the velocity of a particular particle. We compare the quality of answers produced by one global topology, two local topologies, and one random topology. Our results suggest global and random topologies work best, and that their individual effectiveness is problem-dependant.

1. Introduction

Particle Swarm Optimization (PSO) is a nature-inspired optimization method that searches for the global minimum (or maximum) of a function. PSO operates on a set of moving *particles*, each of which represent a single position in the search space. The particles are initialized with random positions and velocities, and each individual particle keeps track of the best (lowest valued) position it has visited. Furthermore, subsets of the particle population (called *neighborhoods*) are able to communicate with each other, such that each neighborhood keeps track of the best location among all its members' visited locations. Using these two pieces of information, each particle's velocity is changed in the direction of its personal and neighborhood best.

The neighborhood technique is important in allowing the particle swarm to explore a large area within the search space before converging to a single point. Without the neighborhood component influencing the velocity, each particle will simply move towards a local minima, and fail to search for the global minimum. There are many methods used to define neighborhoods within the population, and the performance of the PSO algorithm may depend heavily on the neighborhood technique it employs.

In this paper, we test the effectiveness of different neighborhood definition techniques on various PSO benchmark search functions. Our results suggest that a global topology does best at optimizing unimodal functions and a random topology yields the best results for multimodal functions. In Section 2, we describe the PSO algorithm in detail. In Section 3, we describe the neighborhood definition techniques we will evaluate. In Section 4, we describe the search functions used to test the different algorithms. In Section 5, we explain the experimental methods used in this study. In Sections 6 and 7, we present a detailed description and analysis of the results of our experiments. In Section 8, we suggest further research in the use of neighborhoods in PSO algorithms, and in Section 9, we conclude the study with a summary of our findings.

2. Particle Swarm Optimization

Each individual within the swarm holds four pieces of information: position, velocity, personal best position, and neighborhood best position. The *best position* in this context refers to the lowest valued position within the search space that has been found by the individual or group of individuals so far. As described in Section 1, the swarm population is initialized throughout the search space with random positions and velocities. For the purposes of our experiment, we will restrict the position and velocity values such that no particle will be initialized near the minimum of the search function, to avoid biasing the algorithm towards the center of the search space. Each individual's personal best position is initialized to its current position, and the neighborhood best will be initialized after the neighborhood is defined.

The specific neighborhood topologies used in this study will be described in greater detail in Section 3. Within a neighborhood, each particle knows the personal best position of all the particles in the neighborhood, and is partially influenced towards the that best position. It is important that the topologies are defined such that the neighborhoods overlap; sharing particles among neighborhoods allows the swarm to move cohesively towards a global minima, rather than each neighborhood converging towards their own minima. However, if all the particles are in the same neighborhood and gravitate towards a single minima, the final result would be more likely to converge on a *local* minima without conducting a thorough exploration of the search space. On the other hand, if the particles did not communicate with each other at all (neighborhood size of 1), they would not converge because they would not exploit the swarm's best positions. In essence, neighborhoods help PSO balance exploration and exploitation.

Once the neighborhoods and neighborhood minima values are initialized, the PSO algorithm explores the search space by iterating through position and velocity updates. Upon each iteration, the velocity of each particle is updated via the following equation:

$$\mathrm{v}_i = \mathrm{v}_i + c_1\varepsilon_1(p_i - x_i) + c_2\varepsilon_2(p_g - x_i)$$

Where: v_i is the n-dimensional velocity vector of particle *i*
c_1 and c_2 are constants called *learning factors*
ε_1 and ε_2 are random n-dimensional vectors with components between 0 and 1
p_i is personal best position found by particle *i*,
p_g is the best position located by particle *i*'s neighborhood (including itself), and
x_i is the current position of particle *i*.

This velocity update accelerates the particles in such a way that the particle mostly continues in its current direction and speed, but is steered a little towards both its personal best position and neighborhood best position. The position of the particle is simply updated with the velocity vector v_i via the following equation:

$$x_i = x_i + \mathrm{v}_i$$

After the position and the velocity vectors are updated, the fitness of particle *i* is evaluated. If the particle fitness is better than the fitness of the personal best position, the particle updates its personal best position to its current position. Similarly, if the fitness of the new particle best position is better than the fitness of the neighborhood best position, all the particles in the neighborhood update their neighborhood best position to particle *i*'s current position.

The particle swarm balances exploration with exploitation by clustering the particles in neighborhoods and giving each particle a certain degree of randomness. Ideally, the swarm performs a thorough exploration of the n-dimensional search space and eventually converges on a global best position.

3. Neighborhood Topologies

Four neighborhood topologies are tested in this study. Each topology gives a different relationship between the particles, potentially leading to differences regarding the swarm convergence.

In the *global* topology, all the particles are included in the same neighborhood. This means that each particles knows where the current global best position is, so the swarm is likely to converge on the same point earlier than in a more localized topology. This is used as a control to compare against the following topologies.

In the *ring* topology, the particles are organized in a continuous loop and each particle's neighborhood include itself and its direct neighbors. Therefore, each particle has two neighbors, and each particle is a member of three neighborhoods (including the one centered on itself).

In the *von Neumann* topology, the particles are arranged in a two dimensional grid. Each particle's neighbors are the particles directly above, below, to the right, and to the left of the particle. Wrap-around (*e.g.* if a particle is on the top edge of the array, its neighborhood will include the particle at the bottom of the array in the same column) ensures that all the neighborhoods are the same size. Therefore, each particle has four neighbors and is a member of five neighborhoods (including the one centered on itself).

In the *random* topology, each particle has a neighborhood of *k* particles. The neighborhoods are initialized by picking groups of *k* - 1 particles randomly from the swarm without replacement. In each iterations, the neighborhoods may be regenerated in the same way with the probability of 0.2. A key difference between the random topology and the local topologies is the regeneration of the neighborhoods during the iterations. Each particle is only a member of 1 neighborhood, so the regeneration is important in spreading information globally.

4. Search Functions

The search functions we are using to evaluate neighborhood topologies are continuous and exist in R^n for any natural number n. (Note: it is not necessary for a PSO's search function to be continuous.) For a function to evaluate effectiveness of PSO, it should have many many local optima. This increases the chance for poorly constructed topologies to converge on areas that are not global minima. Functions may also have poorly defined optimizers, where a large area around the optimum is nearly flat. The search functions (whose global minima are all located at the origin) used in this study to evaluate the different algorithms are:

The N-dimensional *Rosenbrock* function:

$$f(\mathbf{x}) = f(x_1, x_2, \ldots, x_N) = \sum_{i=1}^{N/2} \left[100(x_{2i-1}^2 - x_{2i})^2 + (x_{2i-1} - 1)^2\right].$$

The minimum for this function lies in a quadratic surface. This function is difficult to optimize because the area around the minima is very flat, making it difficult to converge upon the absolute minimum. A three-dimensional Rosenbrock function is pictured in Figure 1.

The N-Dimensional *Ackley* function:

$$f(\mathbf{x}) = -a \exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right) + a + \exp(1)$$

Where: a = 20, b = 0.2 and c = 2π. (d is the dimension in this equation)

This function resembles an egg carton, with a large depression centered at the origin. Thus, a hill climbing optimization method can easily get trapped in one of the local minima. A three-dimensional Ackley function is pictured in Figure 2.

The N-Dimensional *Rastrigin* function:

$$f(\mathbf{x}) = An + \sum_{i=1}^{n}\left[x_i^2 - A\cos(2\pi x_i)\right]$$

This function resembles something between an egg carton and a dish. This presents the same optimization difficulty as the Ackley function. The three-dimensional Rastrigin function is pictured in Figure 3.

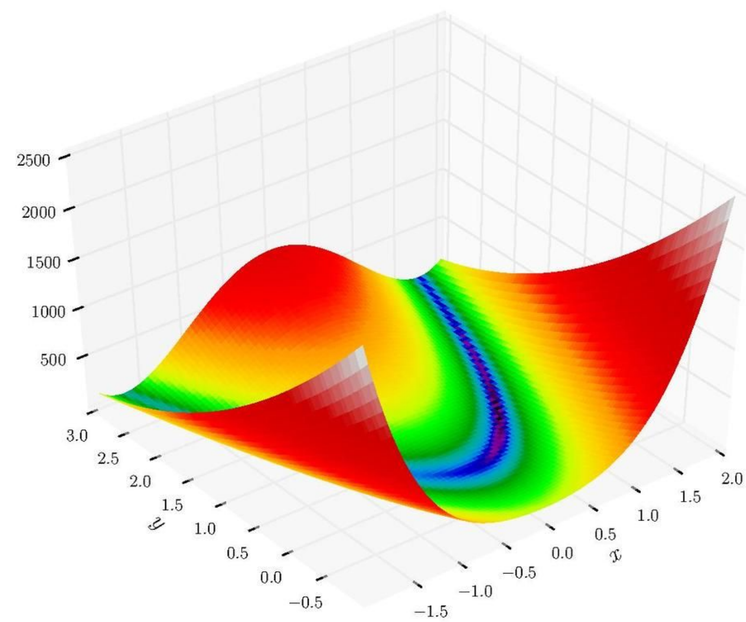

Figure 1: A three-dimensional Rosenbrock Function. The minima is in the center of this image, within the parabolic valley. The area near the valley is very flat, making it difficult to locate for the minimum.

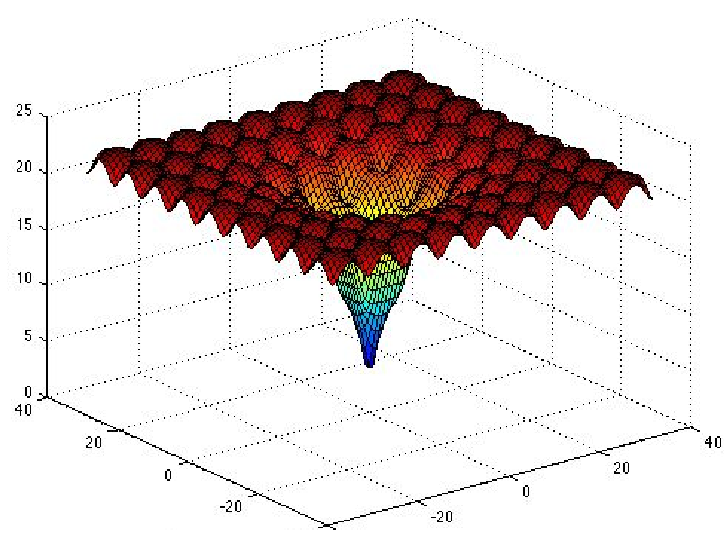

Figure 2: A three-dimensional Ackley Function. The minima is within a significant depression in the center of a relatively even field. The field is covered in bumps, creating many local minima.

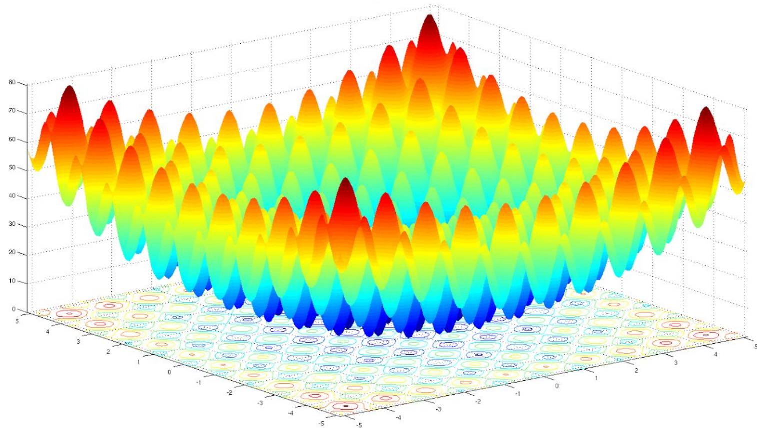

Figure 3: A three-dimensional Rastrigin Function. The minima is at the center of the bowl-like depression, and numerous local minima exist in the curved walls.

5. Experimental Methods

In all testing of the PSO, we ran 10,000 iterations, and tested in a 30 dimensional search space. In order to determine the effectiveness of each neighborhood topology, we tested each topology on all three search functions and with swarm sizes of 16, 30 and 49. This format allowed us to determine which neighborhood topology works best with a specific set of conditions. Testing only on a single search function would have been inadequate, since the characteristics of one search function may allow for good results from on neighborhood topology while the same topology on another search function might yield bad results. The same reasoning goes for swarm size. In short, we needed to test in many cases, so we could make more accurate conclusions on the viability of each topology.

In a single testing case, randomness in the creation of a swarm can produce large variance in the resultant minimum value found. Because of this, it was necessary to test each case multiple times to precisely determine how a neighborhood topology performed. We tested each case (e.g. Random, Ackley, 16 particles) 20 times and took the average over these 20 minimum values. In our results, we considered standard deviation over these 20 tests as a measure of variance in a certain test case. In this case, a low average minimum value is favorable. Low standard deviation is also a favorable measure since it indicates a consistent performance for the algorithm. If a test has low standard deviation, we know that it is unlikely to see outliers, and potentially bad results.

6. Results

The results of the PSO neighborhood topologies can be found in Figures 4 through 7. In general, increasing swarm size improved the average result quality for each topology and search space. The exceptions to this trend are in the Rosenbrock search space, where increasing swarm size from 36 to 49 did not improve performance for the Global and Random topologies, and in the Ackley search space, where the same behavior occurred for the Random topology.

The Rosenbrock function proved quite difficult for the PSO to reliably solve. While the average minimum values discovered were close to zero for the Global and Ring topologies, and near 20 for the Von Neumann and Random topologies, the standard deviation in the results was significant. In each of the four tested topologies, the PSO returned values around 100, very far from the global minimum of 0. Neither of the other two search spaces displayed as severe worst-case search results.

The Ackley Function was difficult to optimize using the Global, Ring, and Von Neumann topologies; the swarms using these topologies often got trapped in the local minima, failing to find the global minimum at the center. When increasing swarm size from 16 to 36 to 49, Global and Ring topologies displayed negligible improvement when increasing swarm size from 16 to 36 to 49, while the Von Neumann topology displayed observable improvement. Only the

Random function was successful in consistently locating the minimum, being able to do so even with a small swarm size of 16.

The Rastrigin Function proved difficult for all four topologies to solve, with none of them consistently locating the global minimum. However, all three topologies displayed regular improvement with increasing swarm size. The topologies ordered from best to worst mean discovered minimum is as follows: Random, Von Neumann, Ring, Global.

The Global topology was significantly better at locating the minimum of the Rosenbrock function than all the other topologies at low swarm size, with only Ring topology outperforming Global at the swarm size of 49. However, Global was the poorest topology for Rastrigin and Ackley.

Similarly, Ring topology was significantly better in the Rosenbrock function than Von Neumann and Random at larger swarm sizes, but was consistently worse in the other functions.

Von Neumann topology was comparable to Random topology in the Rosenbrock function, but was consistently worse than Random in the other two search spaces.

Finally, Random was the only topology that consistently discovered the minimum in the Ackley function. In fact, the Random topology on the Ackley function is the only combination of topologies/search space that consistently returned the minimum.

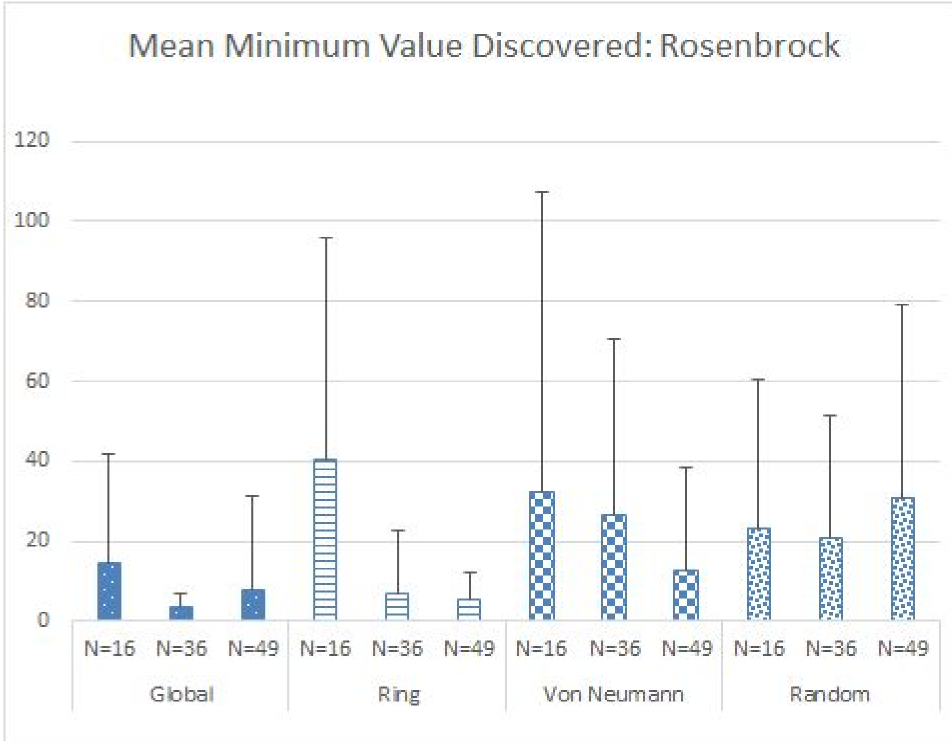

Figure 4: Mean minimum Rosenbrock Function value discovered by the PSO, per swarm size (top horizontal axis label) and topology (bottom horizontal axis label). Each bar represents the mean of twenty trials. Error bars represent standard deviation.

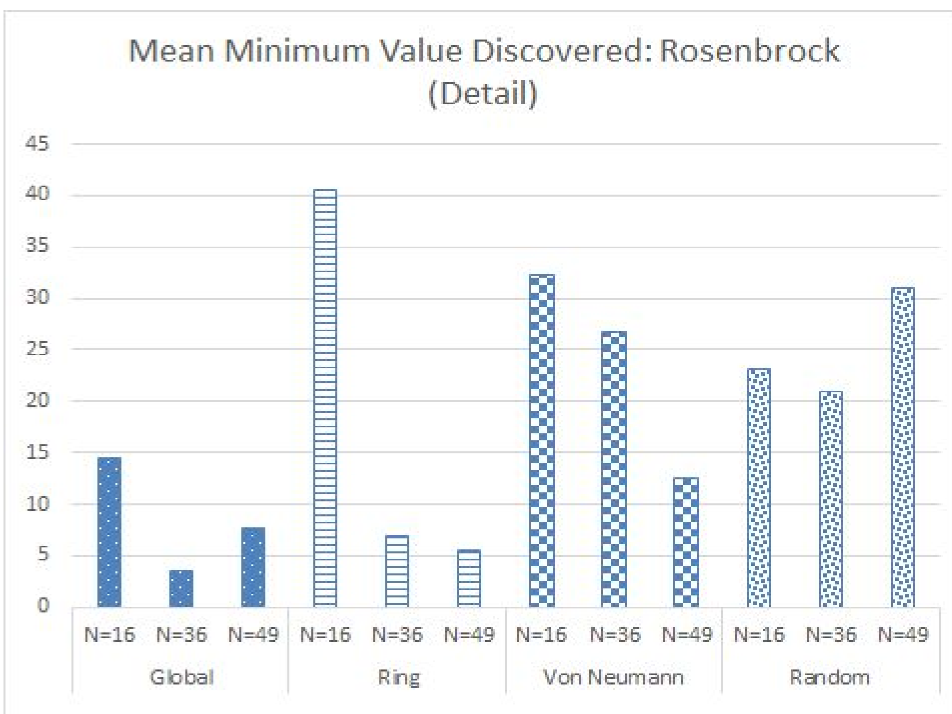

Figure 5: Detail of Figure 4 – error bars bars removed to shorten y-axis scale for clarity.

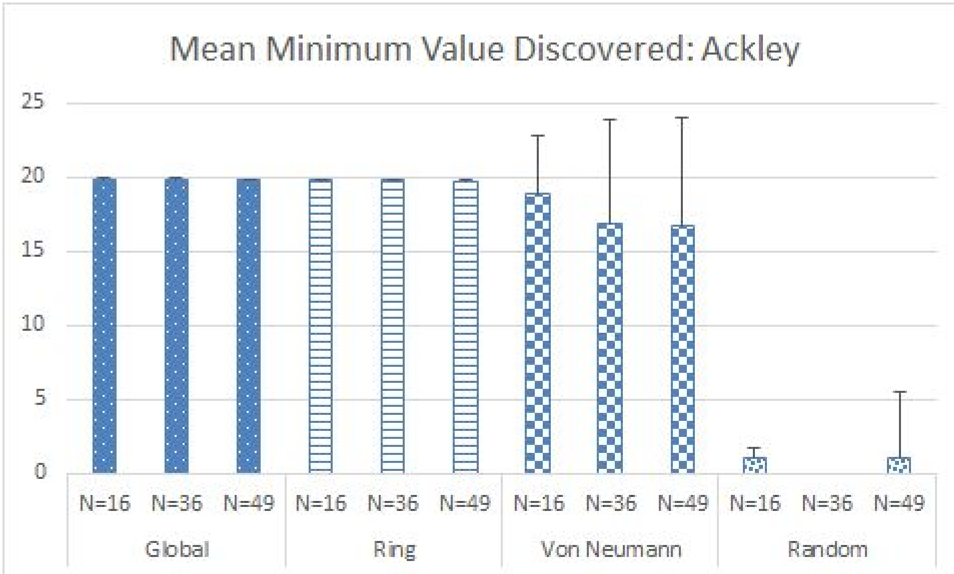

Figure 6: Mean minimum Ackley Function value discovered by the PSO, per swarm size (top horizontal axis label) and topology (bottom horizontal axis label). Each bar represents the mean of twenty trials. Error bars represent standard deviation.

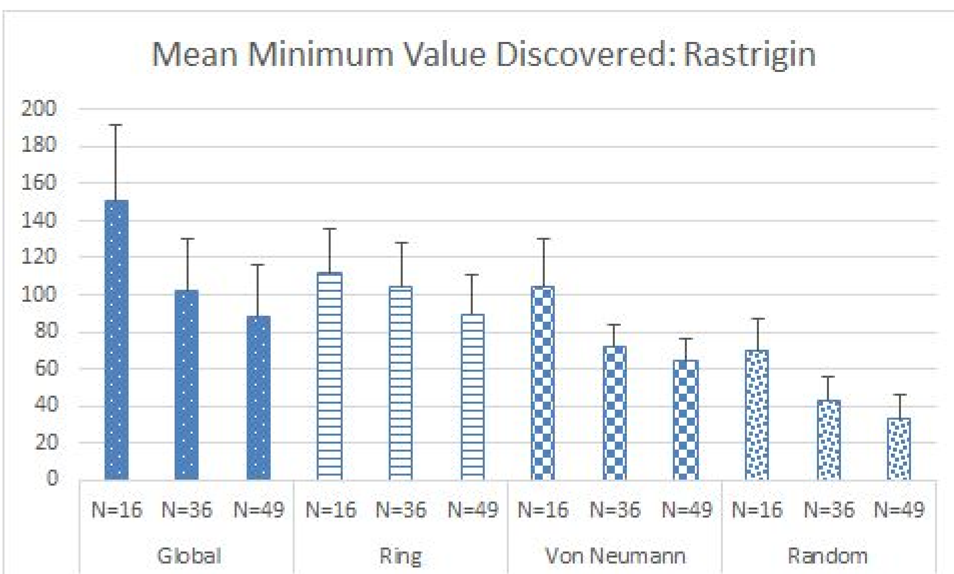

Figure 7: Mean minimum Rastrigin Function value discovered by the PSO, per swarm size (top horizontal axis label) and topology (bottom horizontal axis label). Each bar represents the mean of twenty trials. Error bars represent standard deviation.

7. Discussion

Given the three search spaces tested in this experiment, it is unclear which of the topologies is the overall “best”.

It appears that the topologies exhibit different behavior on the Rosenbrock search space than the Ackley and Rastrigin search spaces. This may be because the Rosenbrock function is unimodal – there is only one local minimum, while the other two functions are multimodal – they are shaped like egg cartons, and have many depressions that contain local minima.

It is likely that the global topology performs very well when there is no possibility of being caught in local minima, while it is a poor choice for functions with local minima. This makes sense, because in Global topology, the entire swarm is drawn towards the global minimum-so-far. If the minimum-so-far is a local minimum of the search function, this can result in a bad result. However, if there are no local minima to be caught in, the swarm will converge near the actual minimum sooner than the other topologies would. In other words, Global Topology emphasizes exploitation over exploration.

In the Ackley and Rastrigin functions, the Random topology outperformed all of the other topologies. Among the Ring, Von Neumann, and Random topologies, Random is the best at disseminating information across the entire swarm, since neighborhoods are constantly recreated. The more localized neighborhood topologies avoid the problem that the Global topology faces: the members of the swarm are not initially drawn to a single point. Instead, they only have knowledge of a few other members of the swarm, so information about the global minimum-so-far propagates slowly across the swarm. This limiting behavior allows the swarm to exhibit more exploration behavior before exploiting the best results.

Ring, Von Neumann, and Random utilize different strategies to restrict the flow of information among the swarm, but the first two used fixed neighborhoods, while Random does not. This may allow information to propagate faster throughout the swarm, while still limiting enough to avoid local minima. For multi-modal search spaces, Random topology may provide the best balance between Exploration and Exploitation to deliver good results.

8. Further Research

In The future, we would look to solidify some of the ideas we set forth above. Specifically, we would like to test the PSO on more search functions. With a selection of functions, both unimodal and multimodal, we could better determine if the global topology is always better with unimodal functions. We could also see if the random topology is a clear choice in functions with many local minima. This would allow us to better understand the behaviors of particles as they act in these respective neighborhoods. It would also enable us to generalize about when to use these different neighborhood topologies.

We would also look at the effects of the random reshuffling done in the random neighborhood topology. In all tests, the random neighborhood of all particles in the swarm was reselected with a probability of 0.2 in each iteration. Since this value remained fixed, we are yet unsure if it is optimal. We would run a number of tests on various unimodal search spaces with varying swarm sizes to attempt to find an optimal probability with which to reshuffle the random neighborhoods.

9. Conclusion

With the testing we have done, we are able to conclude that a global topology is best for optimizing the Rosenbrock function, while a random topology is best for the Ackley and

Rastrigin functions. This is an exceptionally narrow conclusion. However, in optimizing unknown functions with PSO, we may offer suggestions. If run-time is not an issue, test with both global and random topologies. If our conjectures are correct, a global topology will optimize a unimodal function well, while a random topology will yield the best results for unimodal functions. If this sort of testing is infeasible, we suggest using only a random topology. Most difficult search functions are multimodal - that is what makes them difficult. In an unknown space, them, we recommend the neighborhood topology that we have seen perform the best with our limited testing on multimodal functions. However, since all search functions are unique, we cannot offer definite information on how to use PSO in a general case.