Neighborhoods in Particle Swarm Optimization

A Comparison of Four Neighborhood Topologies in PSO

Friday 5, 2018

Sean Cork, Kamaal Palmer, Luca Ostertag-Hill

Nature Inspired Computation
Project 2

***Abstract*—This project examined the effectiveness of different neighborhood topologies in Particle Swarm Optimization (PSO). The objective of PSO is to search a solution space and find the maximum or minimum value (depending on the given function). After creating a general PSO, we implemented four different neighborhood topologies including Global, Ring, Von Neumann, and Random. We then tested these four neighborhood topologies on three different swarm sizes and three different evaluation functions. This was done to determine the most effective neighborhood topology and swarm size for each evaluation function. Our research shows that static neighborhoods (Global, Ring, and von Neumann) perform well on non-convex valley functions such as Rosenbrock. Conversely, dynamic neighborhoods such as random work well on the multi-modal problems with many local minimums, including both Ackley and Rastrigin. Increasing swarm size had a negative effect on the topologies when paired with the Rosenbrock function. The inverse is true with the Rastrigin function. With the Ackley function, swarm size had no effect on static topologies and only a minimal effect on random.**

I. INTRODUCTION

PARTICLE Swarm Optimization (PSO) is a form of an optimization algorithm that utilizes a swarm of particles (similar to a population of individuals in Genetic Algorithms) to search a solution space. The fitness of the particle at any time is calculated as the value of a d-dimensional function that is the solution space. As the particles explore the solution space, they tend to keep moving in the same direction. However, the velocity of each particle is also biased towards the best value each particle has individually seen and the best value the neighborhood containing that particle has found. In this project we will be generating results and experimenting on four neighborhood topologies (Global, Ring, Von Neumann, Random) on three different test problems (Rosenbrock, Ackley, Rastrigin). The first topology, the global topology, describes all particles belonging to the same neighborhood and sharing the gbest. The second topology, the ring topology describes the neighborhood as particles in a ring formation with each particle having at most two neighbors. The third von Neumann topology is characterized as a four cell cellular automata with the particles neighbors being above, below, to the right and to the left of the particle. The last neighborhood topology being used is a random topology of a given size that gets reconstructed with a given probability each iteration.

The objective of this project was to determine what swarm size and neighborhood topology pairing worked best for certain types of evaluation functions. During our experiments we limited iteration length to 10,000 and the number of dimensions to 30. With these set parameters, we tested three different swarm sizes (16, 30, 49) against the four neighborhood topologies listed above for each of the three evaluation functions. The alteration of swarm sizes is critical as is allows us to weigh the cost of particles against the optimality of the solution. The different neighborhood types allows us to determine if certain problem types work better with static or dynamic topologies.

From the tests involving Rosenbrock, we found that the global topology was the most successful, but all static neighborhoods were effective. The random neighborhood had continuous convergence but required more iterations to reach a comparable solution to that of the static neighborhoods. For all four neighborhood topologies, an increase in the swarm size lead to a decrease in the optimality of the solution.

When testing on Ackley, we found that the three static neighborhood topologies (Global, Ring, von Neumann) had little to no convergence to the optimum over 10,000 iterations irrespective of swarm size. All three of these functions got stuck in local minima and produced poor results. The dynamic random topology was extremely effective in dealing with these local minima. In terms of swarm sizes on the random topology, our results showed that a swarm of 16 particles was more effective than one with 30 and 49 particles. However, all three swarm sizes were effective and showed continuous convergence up to 10,000 iterations.

The tests on Rastrigin produced more complex results. The random neighborhood was still more effective than static neighborhoods, but the difference was not as large as in Ackley. While the global and ring topology consistently produced poor results across all swarm sizes, von Neumann produced decent results, especially for smaller swarm sizes. Though an increase in the swarm size did have a positive effect on the random topology, the solutions did not display the same continuous convergence as in Ackley. Rather, the random topology found its best solution in early iterations, ¡ 4,000.

In the next section of this paper, we give a detailed description of the Particle Swarm Optimization algorithm. We also explain why the four neighborhood topologies are suitable for achieving positive results. In section 3 we proceed to explain our methodology behind our testing of the evaluation functions. Section 4 contains our findings and analysis of our results. Finally, we conclude with a section on further work and a section summarizing our results.

II. Particle Swarm Optimization

Particle Swarm Optimization (PSO), invented by Kennedy and Eberhart in 1995, is a stochastic optimization algorithm that shares similarities with traditional Genetic Algorithms. The inspiration for this optimization technique came from the social behavior of bird flocks looking for the highest concentration of food in an environment. In order for the birds to determine the location with the highest concentration, they communicate with each other the quality of their position to birds nearby and as they explore they remember the best location they have found thus far. These two principles became the foundation of PSO. Analogous to Genetic Algorithms, the key for effective PSO is the balance of exploration and exploitation of the search space.

The objective of the algorithm is to determine the optimum of a given function. Explanations of the types of evaluation functions that will be implemented in this project are given at the end of this section. PSO utilizes a swarm of particles to conduct continuous function optimization (a search of the solution space of a given function) to locate the optimum. At the start of the algorithm, each particle in the swarm is initialized at a random position and velocity. The particles then follow a three-pronged strategy of exploration, exploitation, and communication. To effectively explore the space, each particle has a velocity that pushes it forward in its current direction. The particles also exploit their history by tending towards the best value it has found. This value is called the pbest and is the cognitive component. Each particle knows and keeps track of its own pbest to allow for some measure of local search. Lastly, the particles tend toward the best position its neighbors have found. This value is called the nbest and is the social component. How a neighborhood is defined depends on the topology implemented by the algorithm. The simplest neighbor topology is a global one, where each particle tends toward the best value found by any particle in the swarm. The topologies implemented in this project are discussed further below.

Throughout each iteration, the particles update their position based on their previous velocity with influence towards the pbest and nbest values. The particles then reassess the value at their current location, updating the pbest and nbest variables if necessary. The specific procedure of PSO is outlined below.

1) Initialize each particle in the swarm with a random position and velocity in the specified number of dimensions.
2) Evaluate the function at the current position of each particle and update the pbest and gbest variables if better values are found.
3) Update the velocity of each particle using the previous velocity with bias towards the pbest and gbest values.
4) Update the position of each particle using their velocity.
5) Repeat steps 2 to 5 until the optimum is located or the maximum number of iterations has been reached.

The position and velocity update is based on predefined functions. The update function for position for particle i is shown below.

$$\vec{x_i} \leftarrow \vec{x_i} + \vec{v_i}$$

In this function $\vec{x_i}$ is the position of particle i and $\vec{v_i}$ is its newly updated velocity. The function of velocity for particle i is exhibited below.

$$\vec{v_i} \leftarrow \chi(\vec{v_i}+\vec{\mu}(0,\phi_1)\otimes(\vec{p_i}-\vec{x_i})+\vec{\mu}(0,\phi_2)\otimes(\vec{n_i}-\vec{x_i}))$$

In this function $\vec{v_i}$ is the velocity at iteration i and $\vec{\mu}(0,\phi_1)$ is a vector of random real values in the range $[0,\phi_1)$. The value for ϕ_1 and ϕ_2 is commonly 2.05, which allows for convergence of the particles. The velocity is biased towards the personal best p_i in the first element-wise multiplication and towards the neighborhood best n_i through the second. The ϕ_1 and ϕ_2 variables scale the bias towards the pbest and nbest respectively. χ is the constriction factor of the velocity and is usually set at 0.7298. Thus the updated velocity of the particle is dependent on its current velocity and the pbest and nbest values.

The swarm of particles is guaranteed to converge if $\phi_1 + \phi_2 > 4.0$ and if $0.0 < k \leq 1.0$ holds true for the function below. This is why ϕ_1 and ϕ_2 are set to 2.05 in this project.

$$\chi = \frac{2k}{\phi - 2 + \sqrt{\phi_2 - 4\phi}}$$

In this function the k controls the exploration factor of the swarm. If k is on the larger end, then more exploration will occur. Inversely, if k is smaller then more exploitation will occur. Based on these functions the particles in the swarm will continue to update their position and velocity while communicating with their neighborhood to alert other particles of a new nbest or to be alerted of a new nbest. PSO is considered a metaheuristic and works well even on large search spaces. In comparsion to traditional Genetic Algorithms, PSO is relatively easy to implement and requires optimization of fewer parameters. However, PSO is prone to premature convergence. Although the algorithm is particularily poor at local search, it is fairly efficent at solving continuous problems.

A. PSO Topologies

Neighborhood implementation is a critical part of the effectiveness of PSO. A neighborhood defines which other particles a certain particle can communicate with. Each of these neighborhood has a nbest value, that each of the particles in the neighborhood are biased towards. The first solution was to utilize a global topology. Though this technique led to quick convergence, it often generated sub-optimal solutions. Thus, many more forms of PSO topologies have been developed to overcome the grasp of local maximums. The topologies implemented in this project are described below.

1) Global Topology: Global topology is the most common form of PSO. In this topology each particle's neighborhood is the entire swarm. Thus the nbest of each neighborhood is actually the gbest (global best) of the swarm. Generally speaking, communication between all particles in the algorithm allows for faster convergence than local topologies. However, this faster convergence time comes at the cost of the algorithm possibly getting stuck in a local minima and thus having a suboptimal solution.

2) Ring Topology: The ring topology is one of the most common forms of a local topology. In contrast to global topologies in which all particles are in the same swarm, the ring topology describes each particle as having only two neighbors. Consequently, in execution, particles are typically in a ring formation. In comparison to the global topology, the ring topology typically has a longer execution time. However, this longer execution time can result in better performance and minimizes the danger of converging on a local minima. Additionally, ring topologies sometimes require more evaluation to result in the aforementioned improved performance. This topology is static, as the neighborhood of each particle is initialized at the start and is not altered throughout the process.

3) von Neumann Topology: In contrast to Ring Topology, which only uses the left and right adjacent neighbors, the von Neumann topology uses the neighbors above and below the current particle, as well as its left and right neighbors. However, analagous to the Ring topology, von Neumann is static. The objective of the increased neighborhood size is to further limit the dangers local minima pose.

4) Random Topology: In this topology, neighborhoods are randomly created for each particle in the swarm. Given a neighborhood of size k, each particle's neighborhood includes $k - 1$ random neighborhood without duplicates. After each iteration, the neighborhood for each particle is recreated with a given probability. The probability for this project is 0.2. If the neighborhood is recreated, the nbest value is reset so that the value comes from the current neighborhood and not a past one. This makes the random topology a dynamic topology. The purpose of this recreation of neighborhoods, is that it continuously biases particles in different directions, forcing exploration of the solution space. This process minimizes the effect of local minimums that are present in many multi-modal problems (such as Ackley described below).

B. PSO Problems

The functions listed below are benchmark functions for PSO. Functions used by PSO can have anywhere from 10 to 200 dimensions, though the typical value is around 30 to 50 dimensions with around 10 to 100 particles in the swarm. With a given function with d dimensions, each element of a particle's velocity is constrained to the length of that d.

1) Rosenbrock Function: The Rosenbrock function possess a horseshoe valley where the minimum resides. Although optimization functions generally find a solution close to the optimal (of 0), it is described as an ever decreasing minimum. This means that many algorithms reach a value close to the optimal, but never reach it.

2) Ackley Function: The Ackley function is a multi-modal problem that possess many local minimum. However, the closer the local minimum is to the center of the search space (where the global minimum lies), the better that minimum is. It can be imagined as a bowl shaped function, with many egg shaped minimums in circles around the center. Because of its complex shape and central point of optimality the Ackley function poses some difficulty when attempted by many optimization problems and many problems get stuck in the many local minima located inside the graph.

3) Rastrigin Function: The Rastrigin function has components of both of the aforementioned problems. It possess many local minimums that circle the optimum. However, unlike Ackley, the particles do not receive the same pull towards the center by the decreasing local minimums. The Rastrigin function is a very complex function for optimization functions because of its rapidly changing local maxima and minima and its large search space. This leads functions to become trapped in the many local minima.

III. Experimental Methodology

For our experiments on the PSO algorithm, we investigated the effect of modifying swarm size and neighborhood topology on the three evaluation functions. We used the following three different evaluation functions in order to evaluate algorithm performance: Rosenbrock, Ackley, and Rastrigin. Additionally, we used values of 16, 30, and 49 for swarm size and global, ring, von Neumann, and random topologies for neighborhoods. We ran each distinct combination of swarm size, topology, and evaluation function 20 times (with static 30 dimensions as and max 10,000 iterations). Through the variation of the aforementioned elements, there were 36 experiments. These 36 experiments consisted of all permutations of the variables in swarm size [16, 30, 49], topologies [Random, Ring, Global, Von Neumann], and evaluation function [Ackley, Rosenbrock, Rastrigin].

Because the optimum (minimum) value for each of the evaluation functions was at 0.0, we initialized the position of the particles in specific ranges for each function. These ranges are $[15.0, 30.0]$ for Rosenbrock, $[16.0, 32.0]$ for Ackley, and $[2.56, 5.12]$ for Rastrigin. This helps overcome the tendency for PSO to perform better on functions with the minimum at the center of the search space. Research on these types of functions showed that PSO had superior effectiveness.[1] The initialization ranges reduce this increase in performance. Further we initialized the velocity of each particle with ranges $[-2.0, 2.0]$, $[-2.0, 4.0]$, and $[-2.0, 4.0]$ for Rosenbrock, Ackley, and Rastrigin respectively. We also gave the particles minimum and maximum velocities of $[-5.12, 5.12]$, $[-32.768, 32.768]$, and $[-2.048, 2.048]$ for Rosenbrock, Ackley, and Rastrigin respectively. This was done to prevent the particles from accelerating uncontrollably in one direction towards infinity.

In terms of implementation of the topologies, the neighbors for ring and von Neumann were selected at the start based on the order the particles were initialized (not based on the particle's initial position). Further for the random topology, we set k, the size of the neighborhood, at 5 for all of the experiments. The probability of recreation of the neighborhoods was set at 0.2.

Throughout the experiments, we used the evaluation functions to determine the fitness of the particle's location. For our results section, we took the mean and median values of 20 runs for each of the 36 experiments every 1,000 iterations. We graphed each of the 36 experiments to effectively display the effect both swarm size and neighborhood topology had on finding a solution for the evaluation functions. The median value was used in our analysis as opposed to the mean or standard deviation, because it is a better indication of the performance of the function as a whole.

Size 16	Iteration	Mean Function Value				Median Function Value			
		Global	Ring	von Neumann	Random	Global	Ring	von Neuman	Random
Rosenbrock	1,000	132.368	258.046	215.937	321.583	22.8776	75.8254	25.029	30.2439
	2,000	84.4843	177.318	129.647	215.334	16.9841	26.6335	22.6211	24.3706
	3,000	60.24876	156.448	89.503	131.113	8.44374	23.7138	16.4397	22.961
	4,000	51.576	122.156	78.5703	110.61	4.73602	21.1381	14.4149	20.8484
	5,000	40.636	96.4571	67.8358	103.905	3.76952	11.1365	12.6265	19.9424
	6,000	35.5904	79.9141	48.4963	86.2751	1.18851	6.32157	10.9086	18.9835
	7,000	28.6339	73.1722	41.0927	72.051	0.302874	3.55679	9.27616	16.9564
	8,000	23.3129	57.6574	34.6406	64.9834	0.0698363	2.25629	7.33694	15.8677
	9,000	14.2382	49.9011	28.1903	58.7912	0.0270281	1.14892	5.79453	14.5912
	10,000	12.3465	41.2706	25.1421	52.4483	0.0124335	0.47299	4.47097	10.5426
Ackley	1,000	39.7528	39.9686	38.0058	20.5246	19.8598	19.9046	19.8406	1.65982
	2,000	39.7538	39.8015	37.92	11.9243	19.8581	19.8709	19.8199	0.000781697
	3,000	39.7538	39.7652	37.9126	8.82118	19.8581	19.8514	19.8122	1.27984e-07
	4,000	39.7538	39.7567	37.9096	7.21247	19.8581	19.8426	19.8084	3.2252e-11
	5,000	39.7538	39.7452	37.9083	6.91779	19.8581	19.8426	19.8084	2.17604e-14
	6,000	39.7538	39.7413	37.9074	6.14332	19.8581	19.8426	19.8065	2.17604e-14
	7,000	39.7538	39.7407	37.907	6.13277	19.8581	19.8426	19.8065	2.17604e-14
	8,000	39.7538	39.7392	37.907	5.13933	19.8581	19.8426	19.8065	2.17604e-14
	9,000	39.7538	39.7391	37.907	5.08458	19.8581	19.84	19.8065	2.17604e-14
	10,000	39.7538	39.7356	37.905	5.08337	19.8581	19.8329	19.8049	1.46549e-14
Rastragin	1,000	238.142	190.162	154.529	99.3868	92.5309	80.7704	64.6732	53.7467
	2,000	238.142	179.155	153.87	80.5587	92.5309	75.6167	64.6732	49.7479
	3,000	238.142	177.397	153.422	80.492	92.5309	75.6167	64.6732	49.7479
	4,000	238.142	176.028	152.925	80.492	92.5309	73.6267	64.6732	49.7479
	5,000	238.142	175.783	151.681	80.492	92.5309	73.6267	64.6732	49.7479
	6,000	238.142	174.962	151.432	80.492	92.5309	72.6319	64.6732	49.7479
	7,000	238.142	174.842	150.885	80.492	92.5309	72.6318	64.6732	49.7479
	8,000	238.142	174.553	150.885	80.492	92.5309	72.6318	64.6732	49.7479
	9,000	238.142	173.801	150.885	80.492	92.5309	71.6369	64.6732	49.7479
	10,000	238.142	173.628	150.885	80.492	92.5309	71.6369	64.6722	49.7479

TABLE I

MEDIAN AND MEAN PERFORMANCE OF PSO WITH DIF NEIGHBORHOOD TOPOLOGY SWARM SIZE 16

Swarm Size 30	Iteration	Mean Function Value				Median Function Value			
		Global	Ring	von Neumann	Random	Global	Ring	von Neuman	Random
Rosenbrock	1,000	52.4656	77.3515	72.1832	192.107	23.4449	74.6016	76.3674	103.693
	2,000	33.6517	58.7165	60.419	120.264	20.2535	27.089	24.1397	85.6143
	3,000	27.4694	41.5619	48.0919	72.9868	16.1164	23.083	21.6426	75.3435
	4,000	21.5039	35.9406	34.5176	51.9402	13.3868	20.0944	19.2181	25.0854
	5,000	18.8659	29.222	28.3329	50.1541	10.6431	10.7281	17.2666	23.745
	6,000	16.4259	23.8751	26.1968	45.0009	8.19621	9.23646	16.097	21.5476
	7,000	14.4107	20.5994	24.2454	42.6428	6.58651	7.62041	14.5112	20.5565
	8,000	12.4326	13.4221	22.1559	38.0109	4.13498	7.05379	12.9113	19.5349
	9,000	11.1112	8.60299	17.6012	30.3195	3.70737	6.38769	10.9917	17.9677
	10,000	10.0657	7.71585	9.31093	25.8579	2.59359	5.77665	16.024	1.65E+01
Ackley	1,000	19.8909	19.9405	15.9679	1.20954	19.9165	19.9026	19.8524	0.0364799
	2,000	19.8895	19.8541	15.9457	1.0272	19.9037	19.8844	19.8126	3.71252e-06
	3,000	19.8895	19.8284	15.9335	0.037614	19.9037	19.8683	19.8524	4.56482e-10
	4,000	19.8894	19.8198	15.9332	1.69959e-06	19.9037	19.8659	19.8066	8.57092e-14
	5,000	19.8894	19.8198	15.9217	2.03611e-10	19.9037	19.8659	19.8066	1.46549e-14
	6,000	19.8894	19.8198	15.9156	2.12274e-14	19.9037	19.8659	19.8062	1.46549e-14
	7,000	19.889	19.8198	15.9156	1.2168e-14	19.9037	19.8659	19.8062	1.46549e-14
	8,000	19.889	19.8198	15.9156	1.2168e-14	19.9037	19.8659	19.8062	1.46549e-14
	9,000	19.889	19.8198	15.9156	1.2168e-14	19.9037	19.8659	19.8062	1.46549e-14
	10,000	19.889	19.819	15.9156	1.2168e-14	19.9037	19.8659	19.8062	1.46549e-14
Rastragin	1,000	106.41	77.9336	61.2557	93.4709	102.48	80.5916	58.7024	81.9536
	2,000	106.41	71.2701	56.8625	63.3421	102.48	74.5692	55.7176	41.7891
	3,000	106.41	70.8434	56.1157	58.3859	102.48	73.6305	55.7176	41.7882
	4,000	106.41	70.302	56.1156	52.0531	102.48	68.6735	55.7176	36.8135
	5,000	106.41	69.7197	56.0658	46.5508	102.48	68.652	55.7176	36.8135
	6,000	106.41	69.7197	56.0658	45.1536	102.48	68.652	55.7176	36.8135
	7,000	106.41	68.915	56.0658	41.3411	102.48	62.6822	55.7176	36.8135
	8,000	106.41	68.915	56.0658	40.5007	102.48	62.6822	55.7176	36.8135
	9,000	106.41	68.7062	55.4191	40.4104	102.48	62.6822	55.7176	36.8135
	10,000	106.41	68.7057	55.4191	40.3759	102.48	62.6822	55.7176	36.8135

TABLE II
MEDIAN AND MEAN PERFORMANCE OF PSO WITH DIF NEIGHBORHOOD TOPOLOGY SWARM SIZE 30

Size 49	Iteration	Mean Function Value				Median Function Value			
		Global	Ring	von Neumann	Random	Global	Ring	von Neuman	Random
Rosenbrock	1,000	44.0845	63.0329	62.3641	122.252	22.3425	79.3915	75.754	95.9309
	2,000	29.4642	52.5402	48.6973	63.0908	19.0249	37.9287	21.7587	41.0422
	3,000	19.622	32.4215	44.7031	59.9146	15.6434	18.2877	19.5499	28.0474
	4,000	16.8869	30.4253	36.1684	57.3053	13.3192	16.6643	17.6772	25.0013
	5,000	14.5177	19.827	33.9943	48.2995	10.8709	10.199	15.4853	24.0672
	6,000	12.3184	16.3695	25.5735	43.5871	8.69279	8.92774	13.7164	22.8171
	7,000	7.25005	11.3444	17.2521	42.4036	6.9349	7.94729	11.7773	21.7134
	8,000	5.45004	10.0282	12.3518	41.2427	4.60924	7.15418	10.1948	20.7799
	9,000	3.94654	8.98801	7.58777	40.0548	2.94707	6.36918	6.08555	9.8099
	10,000	3.11874	7.82451	6.09477	35.7999	1.48988	5.49405	4.56872	17.4654
ackley	1,000	18.9424	19.8981	19.8269	0.0178919	19.8689	19.9109	19.8424	0.0150306
	2,000	18.9369	19.8193	19.8038	1.41861e-06	19.8633	19.8502	19.8423	1.0126e-06
	3,000	18.9366	19.809	19.7901	1.82562e-10	19.8633	19.845	19.8401	1.66275e-10
	4,000	18.9347	19.8056	19.7858	3.33067e-14	19.8529	19.8376	19.8401	2.17604e-14
	5,000	18.9341	19.8054	19.7853	1.00364e-14	19.8529	19.8376	19.8401	7.54952e-15
	6,000	18.9341	19.8052	19.7768	1.00364e-14	19.8529	19.8376	19.8317	7.54952e-15
	7,000	18.934	19.8051	19.7716	1.00364e-14	19.8529	19.8376	19.8317	7.54952e-15
	8,000	18.934	19.8049	19.7709	1.00364e-14	19.8529	19.8376	19.8317	7.54952e-15
	9,000	18.934	19.8048	19.77	1.00364e-14	19.8529	19.8376	19.8317	7.54952e-15
	10,000	18.934	19.8041	19.7644	1.00364e-14	19.8529	19.8376	19.8317	7.54952e-15
Rastragin	1,000	69.1499	71.9932	47.0827	105.514	64.6722	69.0419	46.1106	111.695
	2,000	69.1495	70.6158	44.3254	71.2595	64.6722	68.652	42.7832	45.7478
	3,000	69.1495	69.3131	44.3254	56.8437	64.6722	68.652	42.7832	38.8033
	4,000	69.1495	69.058	43.8776	49.4742	64.6722	66.6621	40.7933	36.8135
	5,000	69.1495	66.6033	43.2327	42.8422	64.6722	66.6621	40.7933	31.8387
	6,000	69.1495	66.5459	42.7832	37.8333	64.6722	66.6621	40.7933	31.8387
	7,000	69.1495	66.5459	42.7832	34.5828	64.6722	66.6621	40.7933	31.8387
	8,000	69.1495	66.5459	42.7832	33.9005	64.6722	66.6621	40.7933	31.8387
	9,000	69.1495	66.5459	42.7832	33.4797	64.6722	66.6621	40.7933	31.8387
	10,000	69.1495	65.6759	42.7832	32.5417	64.6722	66.6621	40.7933	30.8437

TABLE III
MEDIAN AND MEAN PERFORMANCE OF PSO WITH DIF NEIGHBORHOOD TOPOLOGY SWARM SIZE 49

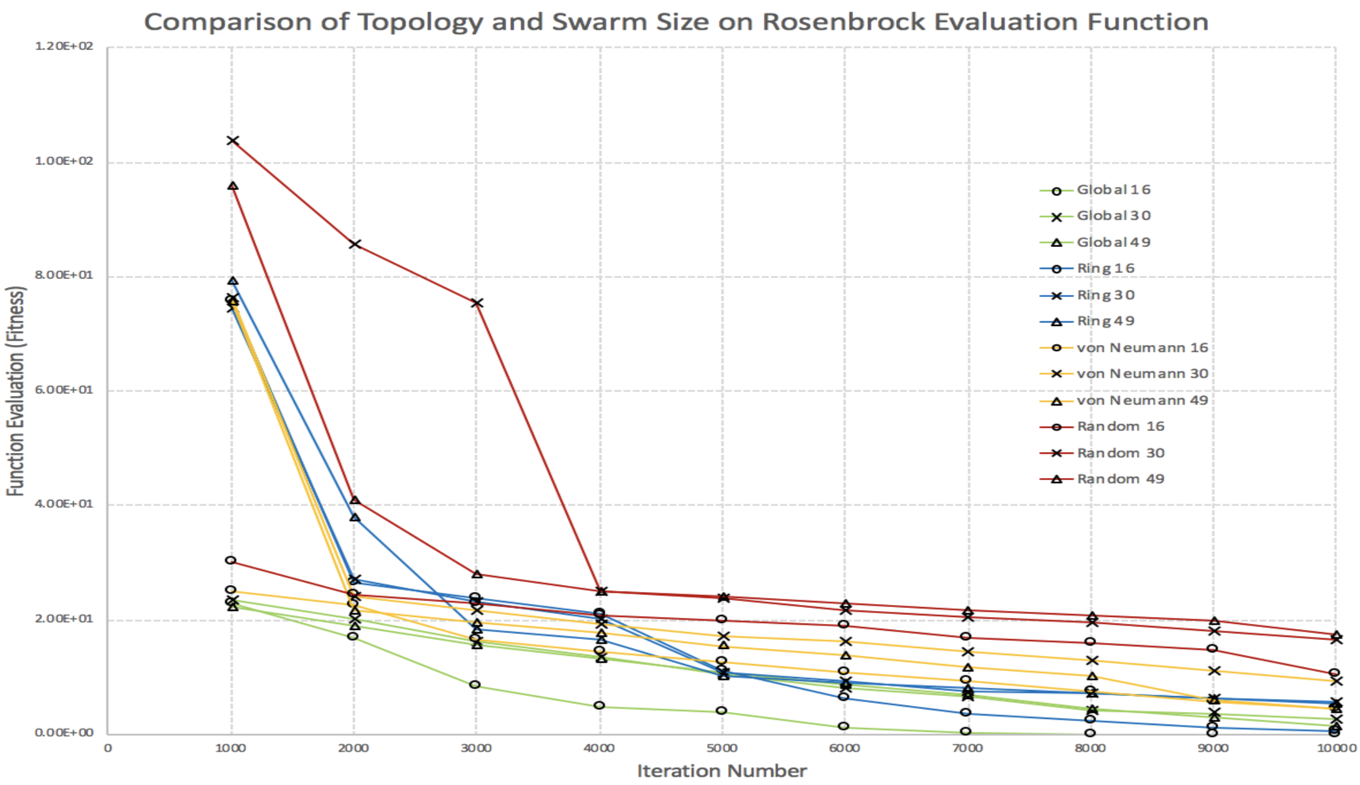

Fig. 1. Median simulation results of varying swarm sizes and neighborhood topologies on the Rosenbrock evaluation function. Parameters are controlled at $Iterations = 10000$ and $numDimensions = 30$.

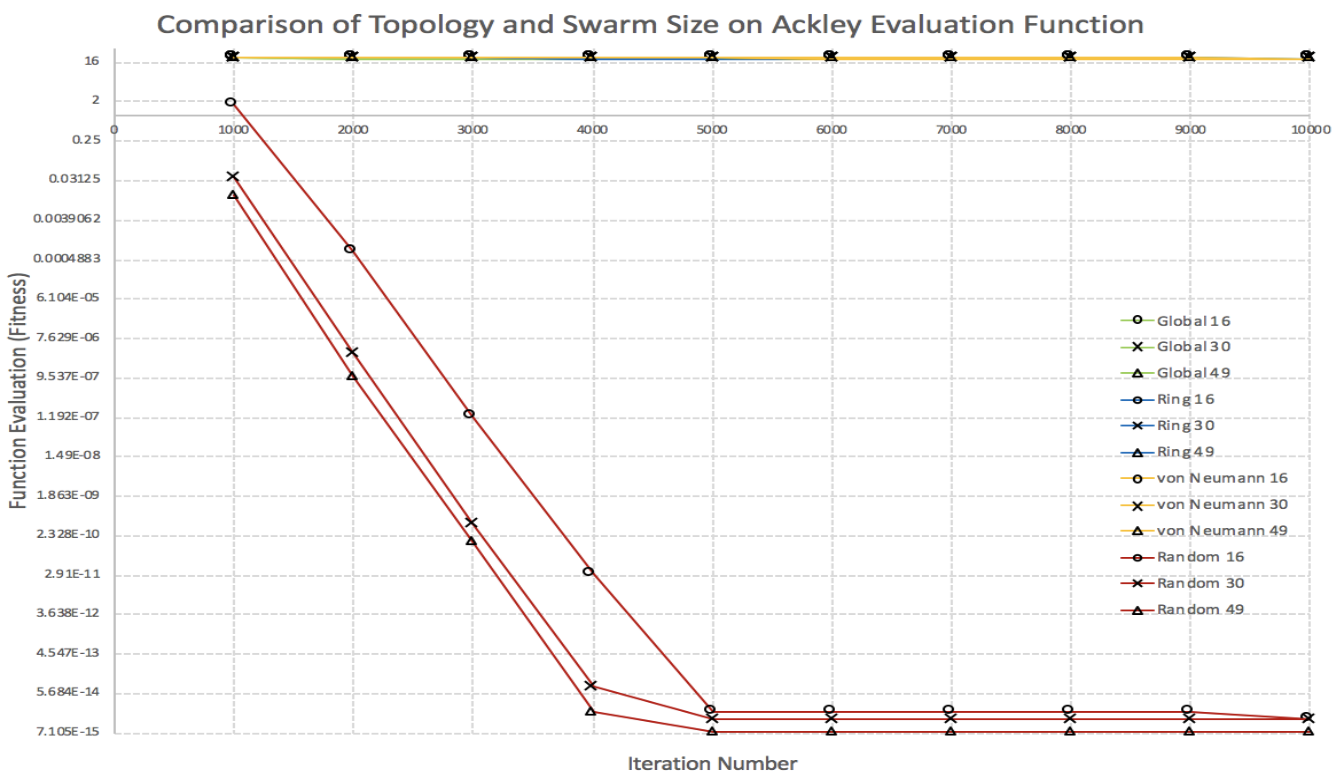

Fig. 2. Median simulation results of varying swarm sizes and neighborhood topologies on the Ackley evaluation function. Parameters are controlled at $Iterations = 10000$ and $numDimensions = 30$. The y-axis of the figure is logarithmic.

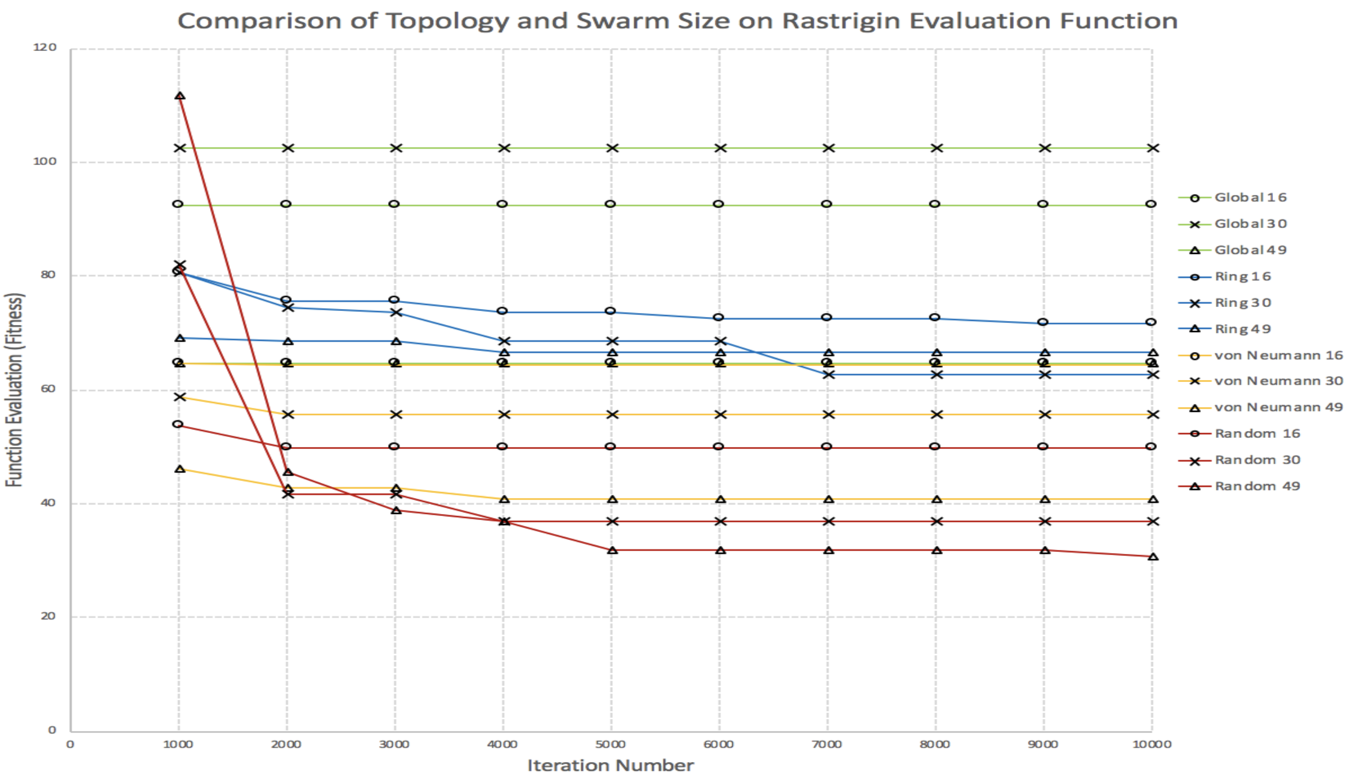

Fig. 3. Median simulation results of varying swarm sizes and neighborhood topologies on the Rastrigin evaluation function. Parameters are controlled at $Iterations = 10000$ and $numDimensions = 30$.

VI. Results

A. *Comparison of Swarm Size and Neighborhood Topologies on Evaluation Functions*

Figures 1 through 3 above present the median of our experiments for each of the three evaluation functions at 1000 iterations.

Figure 1 compares performance of different neighborhood topologies and swarm sizes on the Rosenbrock evaluation function. As exhibited by the figure, all 12 swarm size and neighborhood topology combinations lead to continuous convergence toward the optimum. This is because the Rosenbrock function is a uni-modal value function, where no local minima exist. Generally, the global topology found significantly better values in the early iterations of the algorithm, while the ring, von Neumann, and random topologies required further iteration to reach a comparable solution. This is likely due to the fact that in the global topology particles are quickly pulled toward fairly good values around the bottom of the valley. Conversley, in the other topologies, the neighborhoods are smaller and take more time to find the valley. There was little variance within topologies with different swarm sizes. However, the best median solution for each topology was found by the smallest neighborhood size of 16. The increase in swarm size leads to a larger likelihood of particles being continously pull towards new gbests. This discourages local search, which is necessary with the Rosenbrock function. This figure does show that all of the combinations of swarm size and topology were still converging at iteration 10,000. Thus, an increase in the number of iterations would likely have resulted in better median values and may have allowed the ring, von Neumann, and random topology to catch up to the global topology.

Figure 2 looks at the peformance of PSO on the Ackley function. Global, ring, and von Neumann topologies did not result in much variance in median value throughout the 10,000 iterations. Instead, as shown by Table 1, the median values for each of these functions was stuck at roughly 19.8. This indicates that the particles converged to a local minimum and are unlikely to find the true optimum, even with increased iterations. This issue stems from the fact that these topologies are static, as the particles struggle to overcome the local minimum if all its neighbors are stuck in the same minimum. Conversley, the random topology is dynamic. This, as exhibited by Figure 2, allows it to pull particles out of local minimum reaching a median value of $1.4e{-}14$ or better with each swarm size as shown by Tables 1, 2, and 3. Swarm size had no effect on the static topologies. Random neighborhoods did have a slight increase in performance with an increase in swarm size. This is because good results on the Ackley function are based around maximizing exploration early on. Increased swarm sizes all this to happen. Overall, random neighborhoods allow for greater exploration of the search space and consequently a greater chance of finding the global minimum.

Figure 3 looks at the peformance of PSO on the Rastrigin function. Generally the global topology produced the worst results with median values at iteration 10,000 of only 92.5 and 102.5 for swarm size 16 and 30 (Table 1 and 2). However, global topology with a swarm size of 49 showed a significant increase in performance with a median value of 64.6 at iteration 10,000 (Table 3). The global topology also lacked improvement after 1,000 iterations as a result of its exploitative nature, which trends towards the global best at all times. The next worst topology was ring, which produced results between 65-75. Swarm size had less of an effect on the ring topology, as they all consistently fell in the 65-75 range, though an increase in swarm size did have some positive effect on the solution. The von Neumann topology performed relatively well and was easily the best static topology. Its performance relied heavily on swarm size, as increase size from 16 to 49 shifted the median value from 64.6 to 40.8 (Table 1 and 3). This benefit of an increased swarm size was mirrored by the random topology, which produced the best results on the Rastrigin function. The advantage of an increased swarm size on the Rastrigin function is that particles will be spread out more across the search space, which will help particles stop from all converging at the first local minimum they encounter. Similar to the Ackley function, the random neighborhoods produced the best results. This is again due to the dynamic element of the topology, which pulls particles out of local minimums and forces greater exploration.

B. *Effect of Neighborhood Topologies on the Rosenbrock Function*

In our experiments for the Rosenbrock function, which contains a global optimum of 0.0, there were significant differences in performance dependent on the topology utilized. The three figures in this section outline the effect of these topologies on three different swarm sizes.

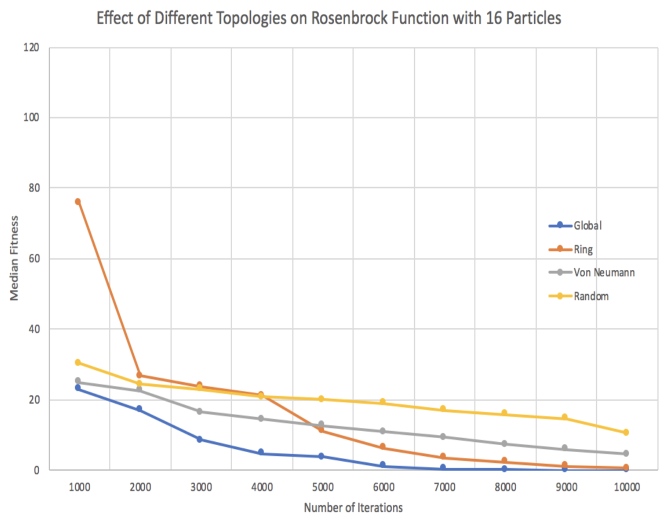

Fig. 4. Simulation results of varying neighborhood topologies on the Rosenbrock evaluation function. Parameters are controlled at $swarmSize = 16$, $Iterations = 10000$ and $numDimensions = 30$.

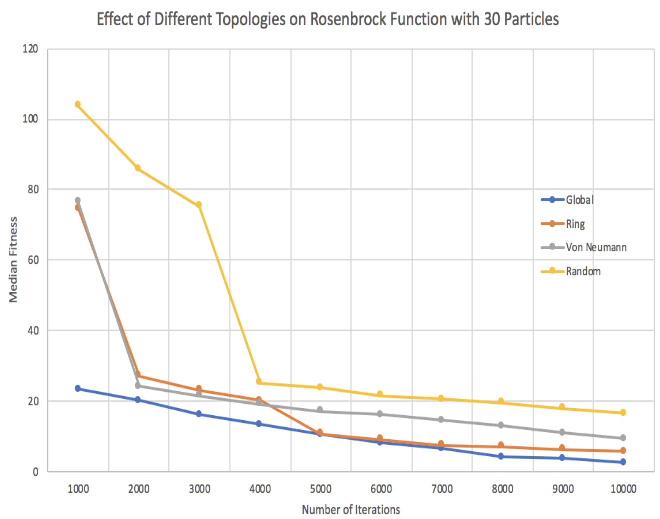

Fig. 5. Simulation results of varying neighborhood topologies on the Rosenbrock evaluation function. Parameters are controlled at $swarmSize = 30$, $Iterations = 10000$ and $numDimensions = 30$.

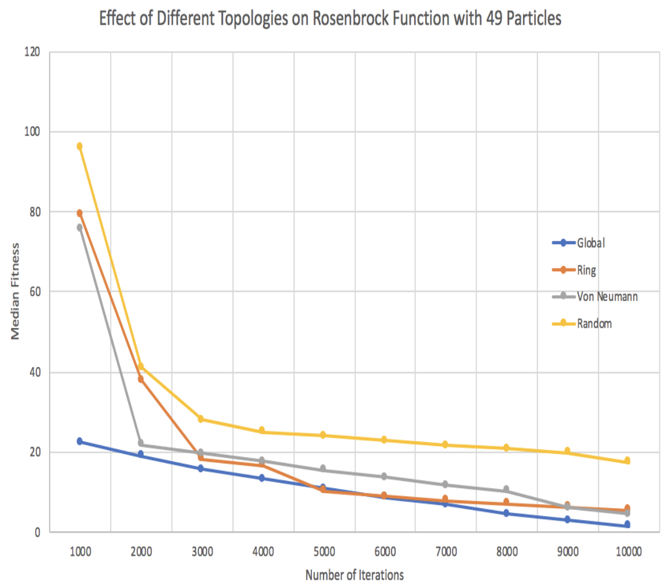

Fig. 6. Simulation results of varying neighborhood topologies on the Rosenbrock evaluation function. Parameters are controlled at $swarmSize = 49$, $Iterations = 10000$ and $numDimensions = 30$.

Figure 4 shows the effect of the four different topologies on Rosenbrock function performance with a swarm size of 16. In this case, the random topology had the worst final performance. The less than stellar performance of the random topology can be attributed to its explorative nature. On the other hand, the global topology, which has a exploitative nature, had the best overall performance. Similar to global, the ring topology showed similar final fitness, with a median value at 10,000 iterations of 0.47 as compared to global's 0.01 (Table 1). Rings performance can also be attributed to its more exploitative nature. The von Neumann topology forces greater exploration due to its increased neighborhood size. This shows in its final result, as its median value was only 4.5 (Table 1).

Figure 5 shows the effect of the four different topologies on Rosenbrock function performance with a swarm size of 30. The final order of topologies in terms of optimality remained the same as swarm size 16. The random topology produced the worst results, followed by von Neumann, ring, and then the global topology. Interestingly, the random topology had extremely poor values in iterations $< 3,000$. At roughly iteration 3,500 the random topology started producing significantly better results. This is due to the dynamic element of the topology, which forces exploration and does not exploit the valley when it is first encountered, unlike the static neighborhoods. The ring and von Neumann topologies started off with fairly weak yet similar initial values. However, between the 4,000th and 5,000th iteration, the ring topology separated itself from von Neumann and got closer to the global optimum. Nevertheless, once again, global shined as the best topology for Rosenbrock.

Figure 6 shows the effect of the four different topologies on Rosenbrock function performance with a swarm size of 49. Similarly to the previous experiment with 16 and 30 particles, random performed the worst overall out of the topologies. Additionally, ring once again found a better result between the 4,000th and 5,000th iterations. However, in this case, there was little distinction between ring and von Neumann in their final values. Once again, global had the best performance, with decent performance on the 1,000th iteration that continued to trend towards the global optimum.

C. *Effect of Swarm Size on the Rosenbrock Function*

In the experiments that dealt with varying swarm sizes for each of the topologies on the Rosenbrock function, we noticed significant differences for the global and ring topologies. The von Neumann and random topologies were not affected as much. The four figures in this section outline the effect these swarm sizes had on the four topologies.

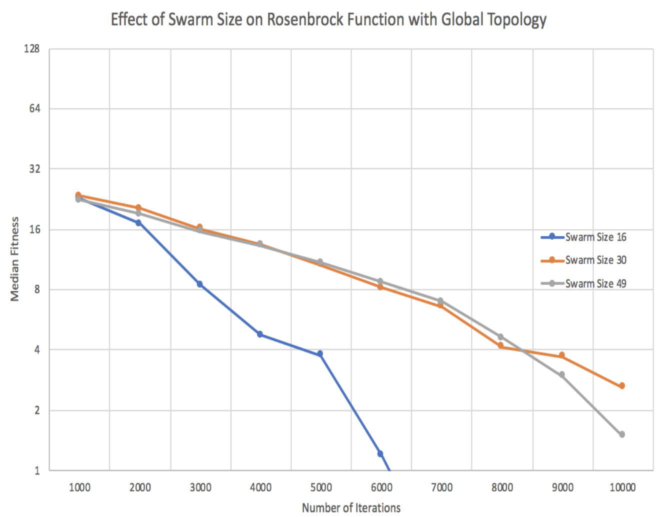

Fig. 7. Simulation results of varying swarm sizes on the Rosenbrock evaluation function. Parameters are controlled at $Topology = global$, $Iterations = 10000$ and $numDimensions = 30$.

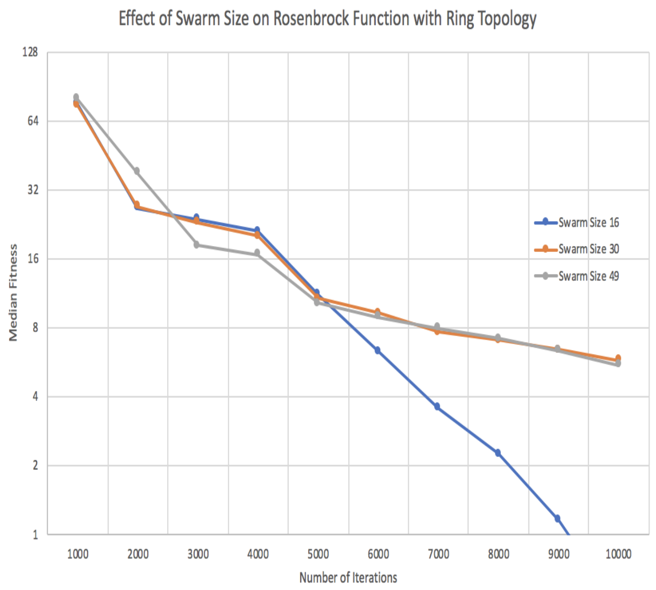

Fig. 8. Simulation results of varying swarm sizes on the Rosenbrock evaluation function. Parameters are controlled at $Topology = ring$, $Iterations = 10000$ and $numDimensions = 30$.

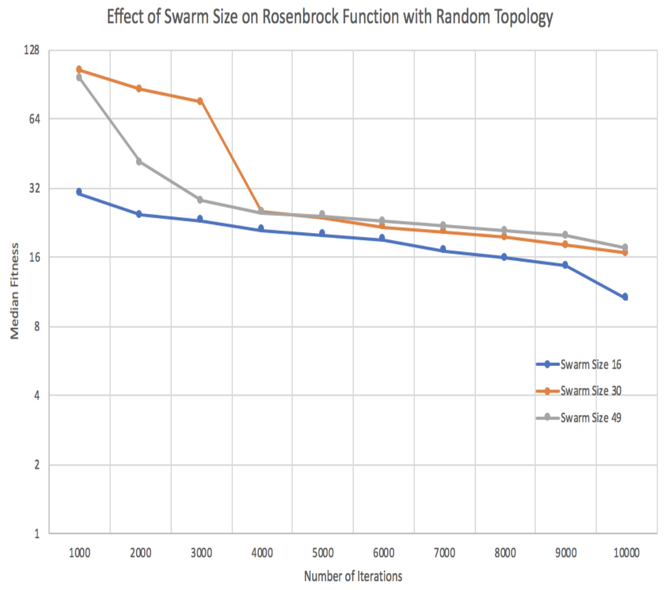

Fig. 10. Simulation results of varying swarm sizes on the Rosenbrock evaluation function. Parameters are controlled at $Topology = random$, $Iterations = 10000$ and $numDimensions = 30$.

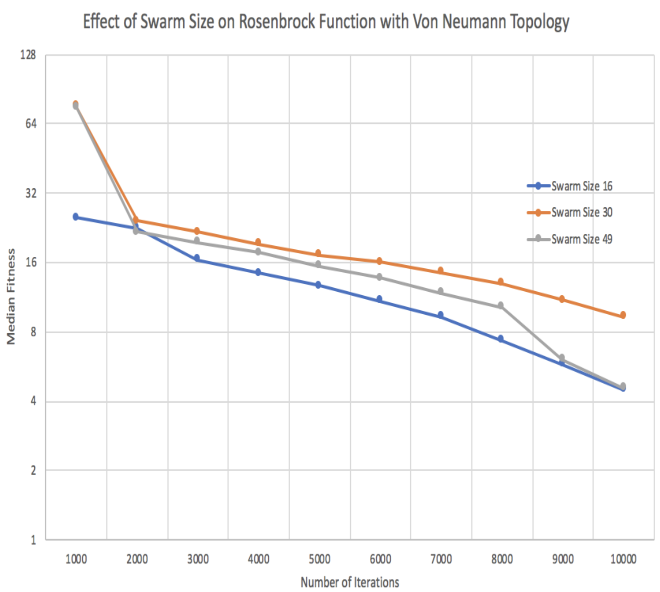

Fig. 9. Simulation results of varying swarm sizes on the Rosenbrock evaluation function. Parameters are controlled at $Topology = vonNeumann$, $Iterations = 10000$ and $numDimensions = 30$.

Figure 7 shows the effect of differing swarm sizes on the Rosenbrock function performance with a global topology. While there was not a discernible difference in performance on the 1,000th iteration between the topologies, as the iterations increased the swarm sizes showed to have significant effects. The swarm size of 16 seemed to best work for global topology, as it found a value of 0.30 by the 7,000th iteration compared with swarm size 30 and 49 that only found values of 2.59 and 1.48 on their 10,000th iteration. This result comes from the nature of particles being more spread out in larger swarms. When the nbest is altered with a large swarm, the particles are constantly pulled across the search space and they cannot focus on local search. With a swarm size of 16 the particles have a greater likelihood of being closer together and thus can focus more on local search, which is required with the Rosenbrock function.

Figure 8 shows the effect of modifying swarm sizes on Rosenbrock function performance with a ring topology. Up to iteration 5,000 all three swarm sizes performed similarly. At iteration 5,000 the median value for swarm size 16 dramatically drops off. The final value for size 16 is 0.47 which is significantly lower than 5.7 and 5.4 for size 30 and 49 respectively. The reason for this drop off is again due to spreading effect of large swarm sizes. Since particles in a ring neighborhood form a ring of particles where particle n is attached to particles $n-1$ and $n+1$. Then particle $n+1$ is connected to n and $n+2$. Though there is no gbest as in global, the connected ring nature of the swarm means that when one particle finds a new nbest it pulls its neighbors to that location, which in turn pulls its neighbors neighbors to that location. This continuous pull to locations around the Rosenbrock valley does not allow for local search, which effective algorithms rely on when evaluating uni-modal problems.

Figure 9 shows the effect of modifying swarm sizes on Rosenbrock function performance with the von Neumann topology. The von Neumann topology produces decent solutions, albeit slightly worse than the ring topology. Similar to previous experiments, the swarm size of 16 once again had the best overall performance. However, in this case, the 49 particle experiment showed similar values with the other two on the 2,000th iteration. Nevertheless, the swarm size of 49 maintained better performance than the 30 particle experiment but worse than the 16 particle experiment until the 9,000th iteration, where its performance mirrored the 16 particle experiment. The benefit of smaller swarms was discussed in relation to the two previous figures and once again applies to the von Neumann neighborhood. The results from swarm size 49 are significantly different then the ones we saw in global and ring. This may simply come from the fact that an increased swarm size has more neighborhoods and more particles, meaning it can explore the solution space faster.

Figure 10 shows the effect of differing swarm sizes on Rosenbrock function performance with a random topology. The random topology does not work very well on Rosenbrock as it focuses too much on exploration. Once again, the swarm size of 16 had the best performance. The swarm sizes of 30 and 49 had similar performance in the end. However, the experiments with 30 particles showed a drastic difference in performance between the 3,000th iteration and the 4,000th iteration. The experiments with 49 particles showed a more steady increase in performance from the 1,000th iteration to the 4,000th iteration. From the 4,000th iteration on, the experiments with 30 and 49 particles showed similar performance, with 30 particles performing slightly better. This can again be attributed to the fact that a greater swarm size constantly pulls particles to different locations in the Rosenbrock valley not allowing for effective local search.

D. *Effect of Neighborhood Topologies on the Ackley Function*

In our experiments for the Ackley function, which contains a global optimum of 0.0, there was a negligible difference in performance between parameters except when the random neighborhood topology was used. This lack of differences across the parameter changes can be explained by the multiple local minimum contained within the Ackley function. These local minimum are particularly difficult to overcome for static topologies, which often value exploitation over exploration. The three figures in this section outline the effect of these topologies on three different swarm sizes.

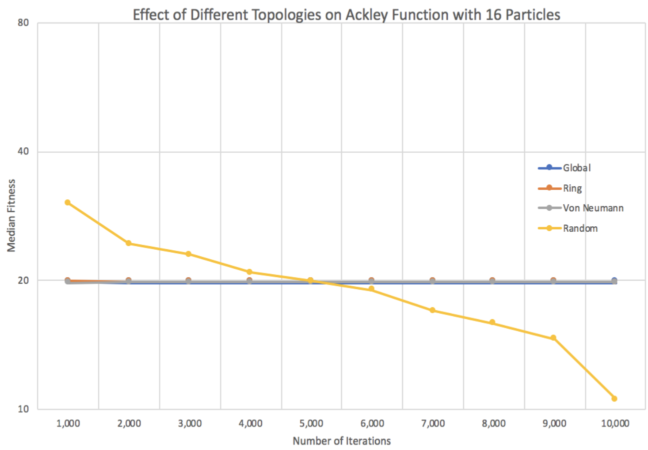

Fig. 11. Simulation results of varying neighborhood topologies on the Ackley evaluation function. Parameters are controlled at $swarmSize = 16$, $Iterations = 10000$ and $numDimensions = 30$.

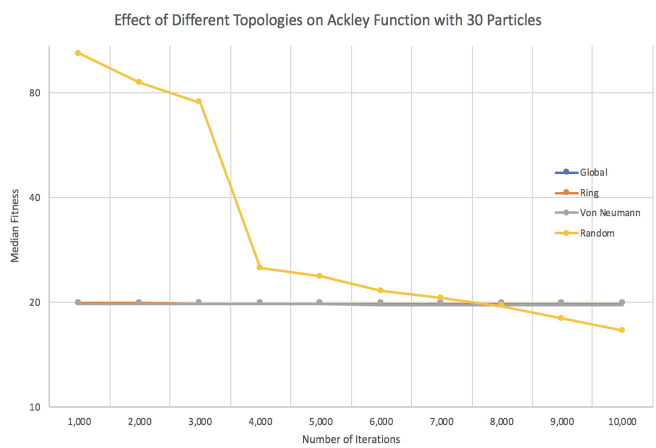

Fig. 12. Simulation results of varying neighborhood topologies on the Ackley evaluation function. Parameters are controlled at $swarmSize = 30$, $Iterations = 10000$ and $numDimensions = 30$.

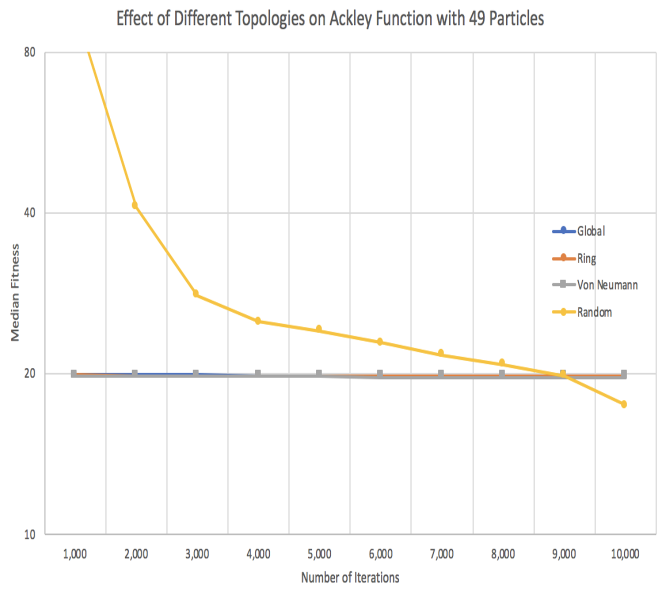

Fig. 13. Simulation results of varying neighborhood topologies on the Ackley evaluation function. Parameters are controlled at $swarmSize = 49$, $Iterations = 10000$ and $numDimensions = 30$.

Figure 11 shows the effect of the four different topologies on Ackley function performance with a swarm size of 16. On one hand, global, ring, and von Neumann showed no discernible variation between them over 10,000 iterations, as their values lay between 19.80 and 19.91 (Table 1). On the other hand, the random topology showed a worse initial fitness that surpassed the other topologies around the 5,000th iteration. The random topology continued to produce better results as the iterations went on. The positive performance of the random topology on the Ackley function can be attributed to the multi-modal nature of the problem. Static neighborhoods struggle to overcome local minimum because they focus more heavily on exploitation. The recreation of neighborhoods in random forces particles to continuously be pulled across the search space, which is very effective for problems like Ackley.

Figure 12 shows the effect of the four different neighborhood topologies on and their respective performance on the Ackley function with a swarm size of 30. Once again, global, ring, and von Neumann showed no variation between themselves and between iteration 1,000 and 10,000. The values lay between 19.80 and 19.92 (Table 2). Random topology showed a worse initial fitness. However, with 30 particles, random topologies performance surpassed global, ring, and von Neumann around the 8,000th iteration. Once again, the random topology continued to produce better performance as the iterations went on.

Figure 13 shows the effect of the four different neighborhood topologies and their respective performance on the Ackley function with a swarm size of 49. Once again, global, ring, and von Neumann showed no discernible difference between them, as their values lay between 19.83 and 19.92 (Table 3). Random topology once again showed a worse initial fitness. However, with 49 particles, random topologies performance surpassed global, ring, and von Neumann around the 9,000th iteration. Though the difference in the final value was not as drastic a difference as the difference with swarm size 16, the curve of the graph suggests that the experiment would have continued to find better values after iteration 10,000, increasing the gap between dynamic and static topologies again.

E. Effect of Swarm Size on the Ackley Function

In the experiments that dealt with varying swarm sizes for each of the topologies on the Ackley function, we noticed significant differences for the random topologies. The static topologies showed no increase in performance with more particles. The four figures in this section outline the effect these swarm sizes had on the four topologies.

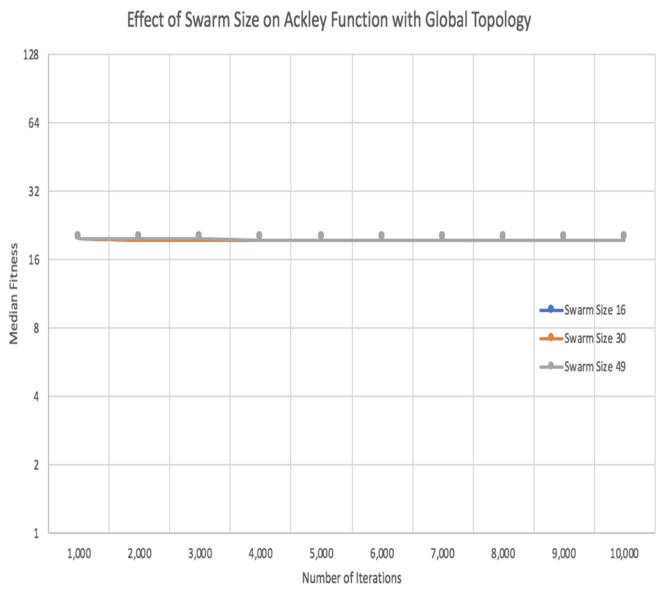

Fig. 14. Simulation results of varying swarm sizes on the Ackley evaluation function. Parameters are controlled at $Topology = global$, $Iterations = 10000$ and $numDimensions = 30$.

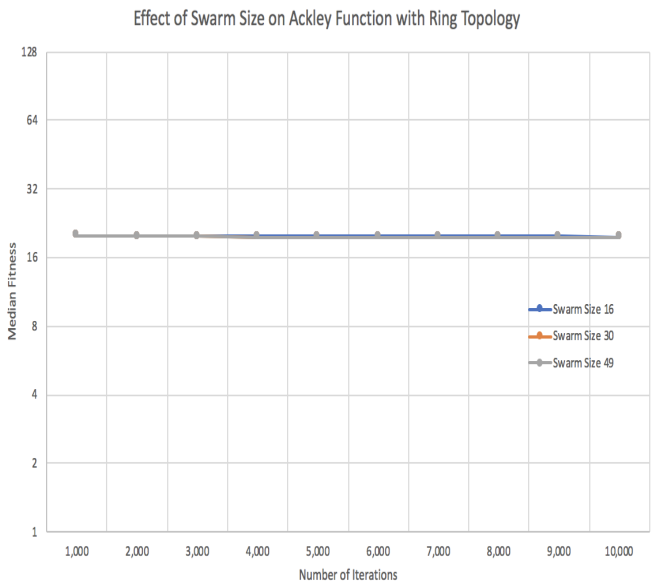

Fig. 15. Simulation results of varying swarm sizes on the Ackley evaluation function. Parameters are controlled at $Topology = ring$, $Iterations = 10000$ and $numDimensions = 30$.

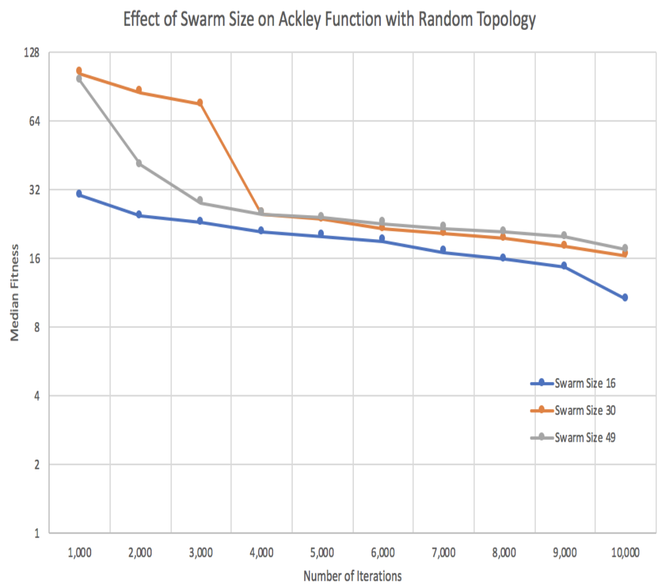

Fig. 17. Simulation results of varying swarm sizes on the Ackley evaluation function. Parameters are controlled at $Topology = random$, $Iterations = 10000$ and $numDimensions = 30$.

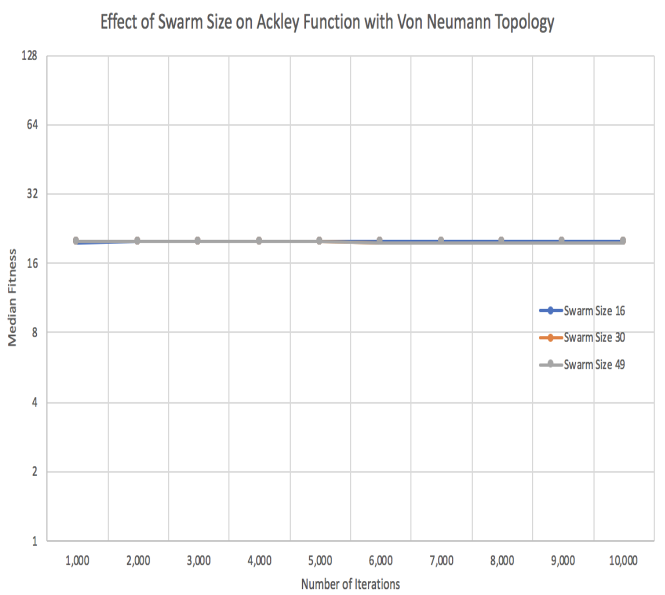

Fig. 16. Simulation results of varying swarm sizes on the Ackley evaluation function. Parameters are controlled at $Topology = vonNeumann$, $Iterations = 10000$ and $numDimensions = 30$.

Figures 14-16 show the effect of different swarm sizes on the performance of the Ackley function with global, ring, and von Neumann topologies. The modification of swarm sizes showed no tangible difference in performance for any of the static topologies. The lack of tangible difference in performance for these topologies can be explained by there not being enough exploration within them, as Ackley holds multiple local minimum. Even with increased swarm sizes, which pushes global exploration, the particles are not able to break from the pull of the local minima.

Figure 17 shows the effect of changing swarm size on Ackley function performance with the random topology. The modification of swarm sizes show differing results for the random topology. The experiments with swarm sizes of 49 and 30 both ended up with similar performance. A notable difference between the two swarm sizes lies in how they reached their final fitness. The experiment with 30 particles showed a drastic improvement between iterations 3,000 and 4,000, going from a value of about 75 to roughly 25. In the case of the experiment with 49 particles, it showed more steady improvement between iterations 1,000 and 4,000 than the drastic drop with 30 particles. In the case of the experiment with 16 particles, it had reduced improvement between iterations 1,000 and 3,000. This is due to the fact that its value at iteration

1,000 was significantly better than that of 30 and 49 particles. The experiments with higher swarm size showed increased total variation between the 1,000 and 10,000th iteration. Random neighborhoods perform well on the Ackley function because they favor exploration. However, the increase of swarm sizes multiplies this effect. To much exploration hurts PSO early on. Later on, the difference in the swarm size performance closes up. However the swarm size of 16 still produces the best final values and has the steepest curve at iteration 10,000 (possibly suggesting it would find even better values with more iterations).

F. *Effect of Neighborhood Topologies on the Rastrigin Function*

The following figures will compare the effects that modifying swarm size and topologies have on the Rastrigin function performance. The Rastrigin function contains multiple local minima and an extensive search space. Thus, this function is best suited to exploration-based approaches in order to escape local optimum and find the global optima. Along these lines, the experiments with topologies and swarm size combinations that focus on a balance of exploration and exploitation are best suited for Rastrigin. The three figures in this section outline the effect of these topologies on three different swarm sizes.

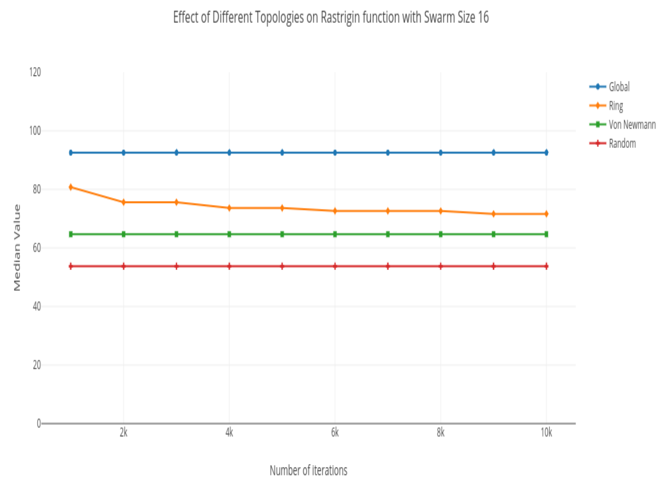

Fig. 18. Simulation results of varying neighborhood topologies on the Rastrigin evaluation function. Parameters are controlled at $swarmSize = 16$, $Iterations = 10000$ and $numDimensions = 30$.

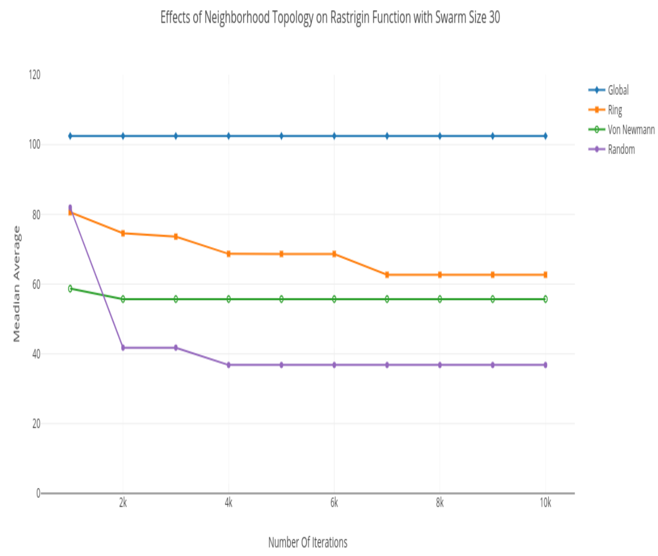

Fig. 19. Simulation results of varying neighborhood topologies on the Rastrigin evaluation function. Parameters are controlled at $swarmSize = 30$, $Iterations = 10000$ and $numDimensions = 30$.

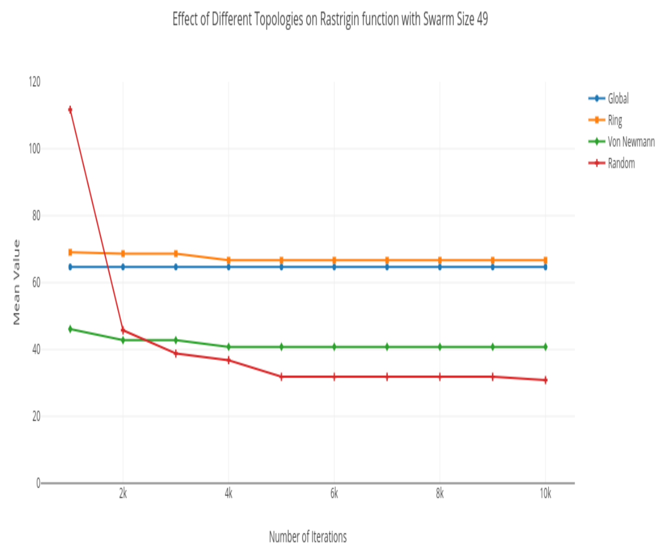

Fig. 20. Simulation results of varying neighborhood topologies on the Rastrigin evaluation function. Parameters are controlled at $swarmSize = 49$, $Iterations = 10000$ and $numDimensions = 30$.

Figures 18-20 display the different topologies across three different swarm sizes. In all three figures the global topology shows no variation between its best value at iteration 1,000 and 10,000. This comes from the exploitative nature of the global topology, by which particles gets stuck in local minimums that they cannot escape. In Figures 18 and 19 the ring topology shows some variation from the iteration 1,000 to 5,000 but then flatlines. The von Neumann topology shows a similar variation in the early iterations before it also flattens, due to the local minimum present in Rastrigin. The static neighborhoods in order of effectiveness on the Rastrigin function are global, ring, and then von Neumann. This order matches the order of the topologies from most to least exploitation (global exploits more than ring, who in turn exploits more than von Neumann). This makes sense since the Rastrigin function prioritizes early exploration over exploitation. The random topology performs the best independent of swarm size. This is again due to its focus on exploration. However, while Rosenbrock is mainly exploitative in nature and Ackley is mainly explorative, Rastrigin requires a solid balance of the two (with a slight lean towards exploration). This is shown by the random topology in Figures 19 and 20, where the initial values of the random topology are extremely poor before they drop around iteration 2,000. These poor early values are a result of large swarm sizes and dynamic topologies focusing too much on exploration.

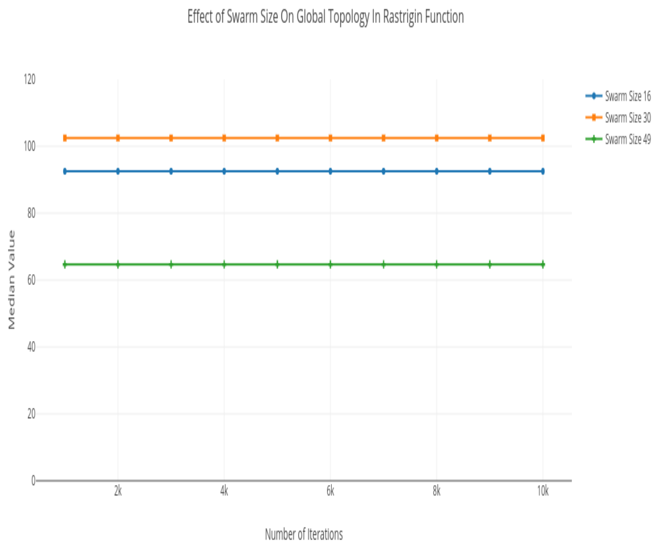

Fig. 21. Simulation results of varying swarm sizes on the Rastrigin evaluation function. Parameters are controlled at $Topology = global$, $Iterations = 10000$ and $numDimensions = 30$.

G. *Effect of Swarm Size on the Rastrigin Function*

In the experiments that dealt with varying swarm sizes for each of the topologies on the Ackley function, we noticed an increase in swarm size led to a general decrease in median values per 1,000 iterations for global, ring and von Neumann topologies. The random topology benefited most by the increased swarm size due to the random topologies nature to prioritize exploration. The random topologies ability to overcome local minima and explore the search space leads to its improved performance. The four figures in this section outline the effect these swarm sizes had on the four topologies.

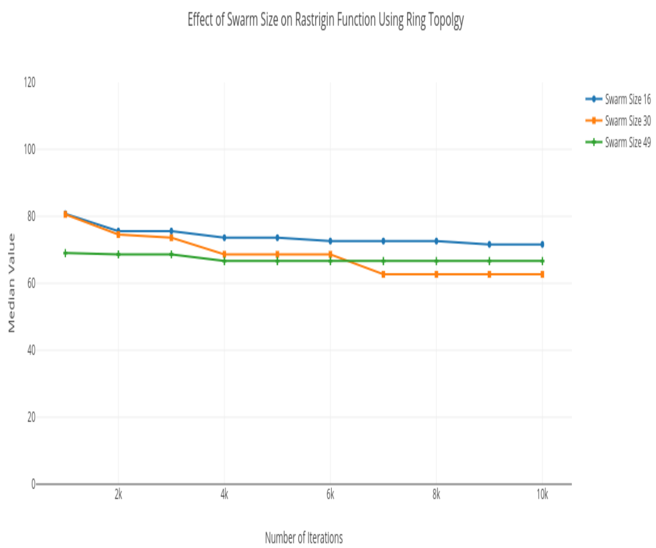

Fig. 22. Simulation results of varying swarm sizes on the Rastrigin evaluation function. Parameters are controlled at $Topology = ring$, $Iterations = 10000$ and $numDimensions = 30$.

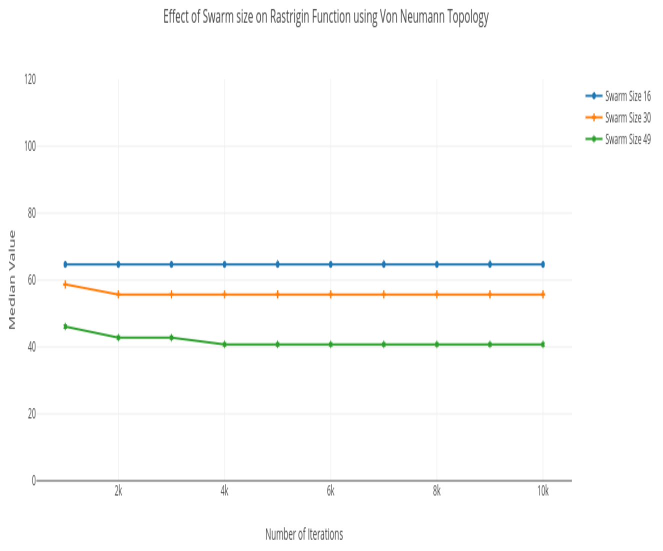

Fig. 23. Simulation results of varying swarm sizes on the Rastrigin evaluation function. Parameters are controlled at $Topology = vonNeumann$, $Iterations = 10000$ and $numDimensions = 30$.

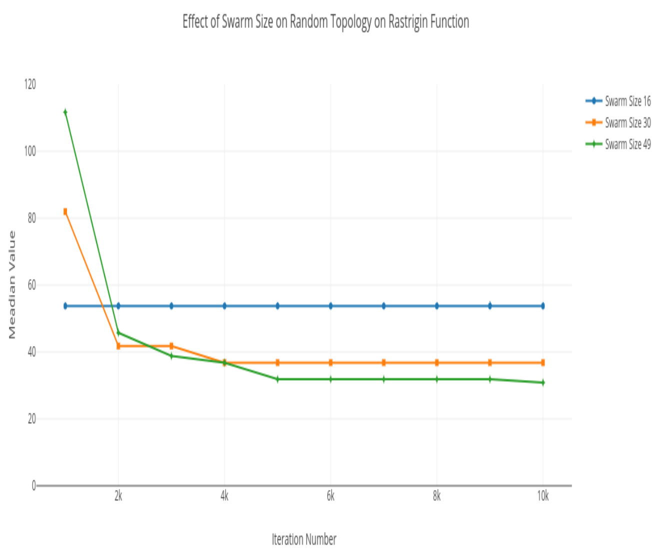

Fig. 24. Simulation results of varying swarm sizes on the Rastrigin evaluation function. Parameters are controlled at $Topology = random$, $Iterations = 10000$ and $numDimensions = 30$.

In figure 21 the graph compares how increasing swarm size affects the performance of the global neighborhood. The figure shows that there is no variation between iteration 1,000 and 10,000 for any of the swarm sizes. However, the swarm size of 49 has the best final value. This suggests that global neighborhood gets particles stuck in local minimum early, but an increase in the swarm size has a higer tendency to find a better location early that all the particles converge to.

In figure 22, the graph shows the correlation between the increase in swarm sizes and the ring topology. Although there is slight variation among each of the swarm sizes from iteration 1,000 to 10,000, this change is not large. Again, bigger swarms generally produce better results. The reasons for this lack of variation and the benefit of larger swarms is parallel to the global topology. The exploitative nature of the topology gets particles stuck in local minima. With more particles there is a greater chance of a particle finding a better location early on.

In figure 23, the graph shows the correlation between an increase in the swarm size and the von Neumann topology. Similar to our findings in global and ring, the increase in swarm size leads to better performance of the algorithm. This is again due to the probability that an increased swarm covers more locations and finds a better location early on.

In figure 24 the graph compares the effect of swarm size on random neighborhoods. The smallest swarm size seems to converge within the first 1,000 iterations, while the increased swarm size seems to give the particle more search space to explore leading to a significant decrease in the median over 10,000 iterations. This could be due to the fact that the small swarm size does not lead to enough exploration in the search space. An increased swarm size with a random topology heavily focuses on that exploration.

VII. Further Work

In further work, one goal would be to see the effect the constant increase of swarm size and how its whether there is a boundary where the increase in swarm size does not affect the outcome in any significant way. In out current model we only have three different variations on swam size and while this might seem like a lot it is not excessive and the variability could be increased. If we had more time to do the project another thing that might have been useful was if we tested the effect and correlation of the variability in dimensions and the variability in swarm size. To see this effect you would run the test of each of the original swarm sizes and have three distinct dimensional sizes that you would run over 20 times on each different swarm size This would amount to about 9 tests without changing the neighborhood. You could then do this for each neighborhood topology.

Further it would be interesting to start applying the random neighborhood topology to more functions that have a lot of local minima to see if the general performance of the dynamic nature of random neighborhoods is replicated.

Another thing that could be interesting to try would be to increase the k constant of the random neighborhood creation to see if having random neighborhoods that are bigger than 5 values would increase the exploration in tough optimization problems. Another idea to play with besides modifying the k constant would be to increase the rate at which random neighborhoods generate past the first generation. We tested random neighborhoods with a chance for generation for each subsequent iteration with a probability of 0.2. It would be interesting to see if there is an optimum number for that chance.

The last possibility for further work would be to increase the iteration size and see if the neighborhoods topologies that would get stuck in local minima on the Ackley function and the Rastrigin function would overcome the boundary of local minima and find a more optimal solution.

VIII. Conclusions

Through our experiments with Particle Swarm Optimization we determined that performance is dependent on both the swarm size as well as neighborhood topology. Each evaluation function requires a different balance between exploitation and exploration. Along these lines, neighborhood topologies are the biggest factor for balancing the two.

For our Rosenbrock function experiments, 16 particles and a global neighborhood topology produced the best results, with a median value of 0.0124. Global neighborhoods, which promote exploitation, work best as a result of Rosenbrock's valley which require local search.

Within our Ackley function experiments, 16 particles and a random neighborhood topology produced the best results, with a median value of 10.5426. Random neighborhoods, which promote exploration, work best as a result of Ackleys numerous local minima. The Ackley functions local minima trap exploitative topologies within. Random topologies dynamicism pushes past local minima and continually explores for the global optimum.

For our Rastrigin function experiments, 49 particles and a random neighborhood topology produced the best results, with a median value of 30.8437. Similar to Ackley, random neighborhoods, which promote exploration, work best as a result of Rastrigins numerous local minima and large search space. However, the increased swarm size of 49 from Ackleys optimal 16 stems from the fact that Rastrigin prioritizes maximum exploration over a balance of the two as in Ackley.

In the end, we determined that the type of topology, static vs. dynamic, lead to the largest gains in topology largely dependent on the modality of the problem. Uni-modal problems without local minimums do not trap particles, thus exploitation allows the swarm to find and do local search around good locations early on. Multi-modal problems have many local minimums and thus require a large amount of exploration to reach the optimum. Neighborhoods that are recreated during the PSO process allow for this increased exploration and thus are the most effective topologies for these type of problems.

IX. References

1) C. Monson and K. Seppi (2005). Exposing Origin-Seeking Bias in PSO. Proceedings of the 2005 Genetic and Evolutionary Computatio Conference, pp 241-248.
2) Virtual Library of Simulation Experiments:. (n.d.). Retrieved from https://www.sfu.ca/ ssurjano/ackley.html