Introduction

Particle Swarm Optimization is a method in which to optimize particles to a solution: it is non-deterministic, and population based, inspired by flocks or swarms of birds and other flying animals. A collection of particles is used to emulate a swarm and iteratively move through a search space looking for a solution. It is apt for static optimization problems.

Particle Swarm Optimization Algorithm

Through implementation and experimentation of a vanilla particle swarm optimization algorithm optimizing the Rastrigin function, the performance of such an algorithm can be compared to uniformly random searches. The Rastrigin function is defined as below:

$$f(x) = 10n_x + \sum_{i=1}^{n_x}(x_i^2 - 10\cos(2\pi x_i))$$

Where n_x is the dimensionality of the problem and x_i is a measure of position of a particle in a given dimension, constrained within the search space of axes $[-5.12, 5.12]$ for all axes. An optimal solution is 0, or when all values of x_i are 0.

The PSO algorithm generates particles at random throughout the search space with their positions stored. They are synchronously iterated over, updating their positions based on a velocity vector, which is a function of its fitness, the swarm best fitness, and a few PSO parameters. The new location for the particle is updated with the below equation:

$$l_i(t+1) = l_i(t) + v_i(t+1)$$

Or simply, the sum of the current location and the current velocity. The velocity is updated, however, based on an iterative step which can be described by the below equation:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1\big(y(t) - x_i(t)\big) + c_2 r_2(\hat{y}(t) - x_i(t))$$

Where $v_i(t)$ is the particle velocity at a given time t, ω is the inertial weight, c_1, c_2 are cognitive and social coefficients respectively, r_1, r_2 are stochastic components, and $y(t)$ and $\hat{y}(t)$ are the personal best and swarm best solutions respectively. This equation generates a new velocity vector for the particle, whereby its location is changed based on its trajectory through the search space.

PSO Parameters

Of note are ω, c_1, and c_2 as PSO parameters. The above equation can be roughly broken into three parts: an inertial weight component, a cognitive component, and a social component. Note the three terms each have one of these parameters as multiplicand.

The inertial weight parameter is the measure of resistance to change in velocity or how difficult it is to change velocity. A higher inertial weight means the above equation favors that term and will more closely resemble the original velocity. Typically, $\omega \in [0, 1]$.

c_1, or the cognitive coefficient, governs how much emphasis the particle's personal best has on the new velocity. A higher cognitive coefficient means more global exploration and local exploitation. c_2, or the social coefficient, however, governs how much the swarm's personal best has on the new velocity. A higher social coefficient means more global exploitation and local exploration. In a general sense, what this means is these two coefficients balance whether a particle is independent or relies on the swarm to determine its new velocity. Usually $c_1, c_2 \in [0, 2]$.

This iterative process allows each particle to "move" around the search space based on the PSO parameters.

Testing Methodology

This study is meant to explore the effect these parameters have on efficacy of the PSO algorithm using five base tests:

	Test 1	Test 2	Test 3	Test 4	Test 5
Inertial Weight	0.729844	0.4	1.0	-1.0	N/A
Cognitive Coefficient	1.496180	1.2	2.0	2.0	N/A
Social Coefficient	1.496180	1.2	2.0	2.0	N/A
Number of Iterations	500	500	500	500	0
Swarm Size	50	50	50	50	25000

Each test is performed five times using differing random seeds, with the results gathered taken from the mean of all tests performed.

Note the last test is different from the others: this is a random search instead performing no iterative steps and instead determining fitness based on uniformly randomized particles in the search space. To be a fair test, the number of particles is increased to match the number of particles multiplied by the number of iterations of any other test.

Experimental Results

For Test 1, the swarm slowly converges to an approximate solution and does not get trapped in a local optimum. For most of the iterations, the swarm best and swarm average fitness differ by some margin, however both remain well below a random search. This test provided the most fruitful results with the closest approximate solution to the Rastrigin function with a best fitness of 47.24 (0 is optimal) for both swarm best and swarm average.

Comparatively, for Test 2, a solution is quickly converged upon within the first 100 iterations which is much faster than in the prior test. Likewise, both the swarm best and swarm average fitness converged to a close solution, but not as closely as Test 1. The best fitness found was 111.22. For this test, all particles fell into a local optimum and failed to converge further to a more fit solution.

Both Tests 3 and 4 provide almost identical results with an immediate convergence to a low-fitness solution, becoming trapped in a local optimum immediately. In these tests, however, both still outperformed a random search despite failing to find a suitable solution.

Interpretation

Concerning Test 1, a relatively high inertial weight contributed to a resistance to change velocity (or maintain a new velocity closely resembling the original velocity): this lead to slower convergence. The limitation of 500 iterations meant the algorithm was still approximating a solution and perhaps with more iterations, a better solution could have been found.

With Test 2, however, with a lower inertial weight, the velocity is more apt to differ from prior velocity: in other words, the particles moved “faster”, leading to a much quicker convergence. Owing to the lower cognitive and social coefficients, however, paired with the stochastic elements ascribed to those terms, meant less “discovery” in the search space, leading to a less fit solution.

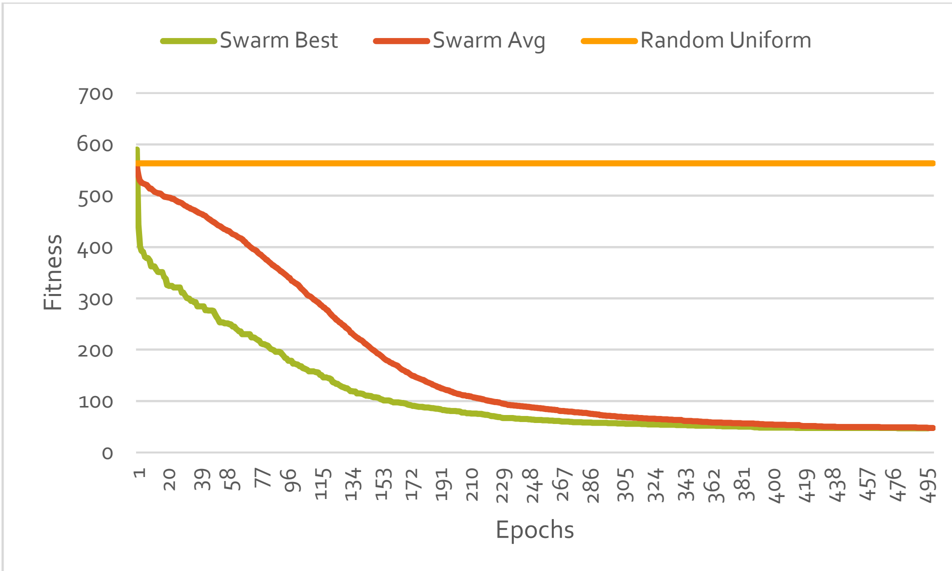
The remaining two tests were too resistant to change due to their high magnitude inertial weights. Despite their large cognitive and social coefficients, particles failed to traverse the search space and instead remained still or regressed further from the solution. As personal bests are not updated in the event of a particle moving further away from the solution, the graphs for both Test 3 and 4 resemble each other. Were this contraindication removed, however, you might see fitness decrease as further iterative steps are made, due to the negative inertial weight in Test 4.

Conclusion

On comparing to a randomized search, the efficacy of a PSO algorithm in finding a solution to static problems such as the Rastrigin function is immediately apparent. However, only with careful PSO parameter selection are results viable: inappropriate selection leads to results which are only minutely better than random and could be considered within the realm of within margin of error.

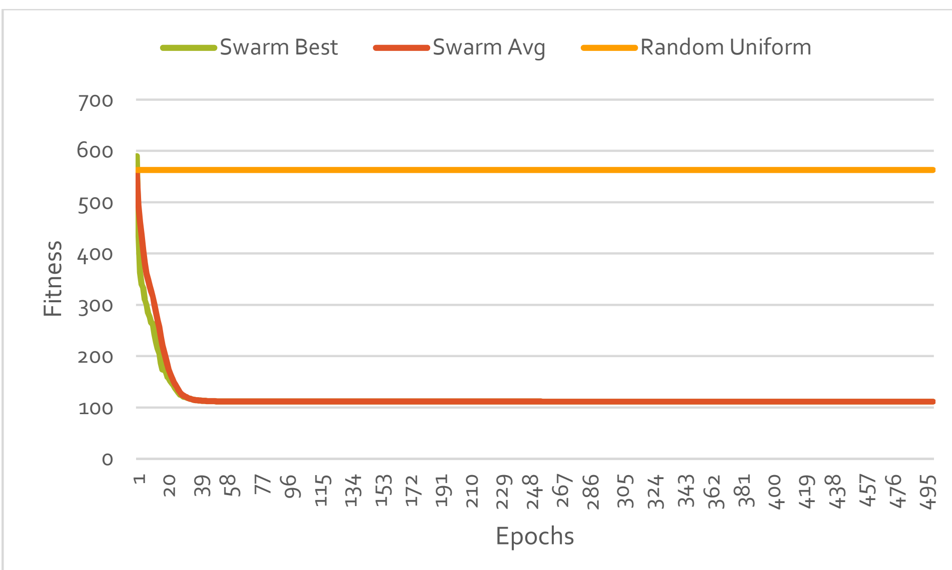
Graphs and Tables

Test 1



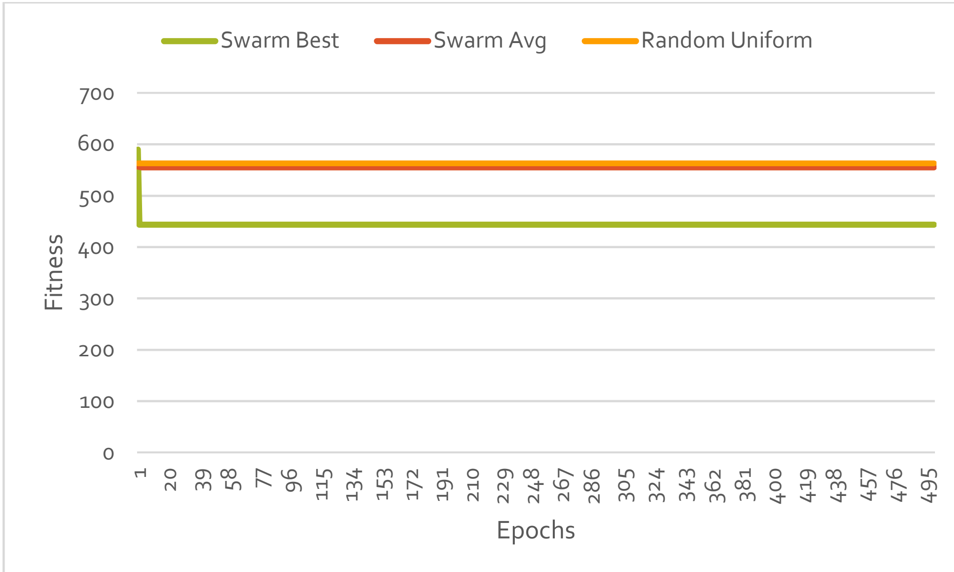
	Swarm Best	Swarm Average
Top	47.23694	47.75822
Mean	110.766	169.3123
Median	63.3205	87.10366
Stdev	91.7258	151.556

Test 2



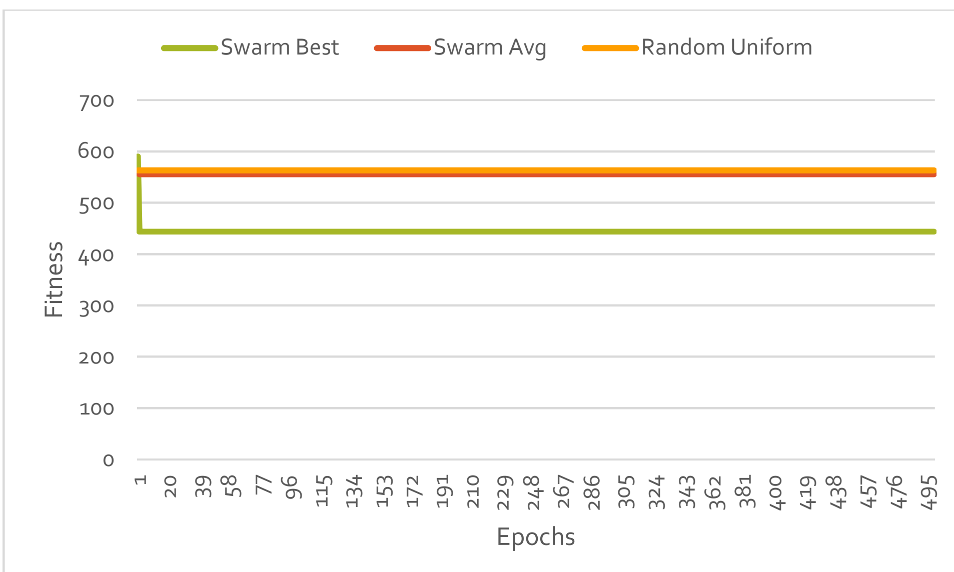
	Swarm Best	Swarm Average
Top	111.2222	111.2222
Mean	118.4522	120.7435
Median	111.2226	111.2226
Stdev	38.30103	47.34845

Test 3



	Swarm Best	Swarm Average
Top	443.6582	555.3408
Mean	443.9499	555.344
Median	443.6582	555.3408
Stdev	6.52282	0.060433

Test 4



	Swarm Best	Swarm Average
Top	443.6582	555.59
Mean	443.9499	555.5922
Median	443.6582	555.59
Stdev	6.52282	0.048301