Université Libre de Bruxelles

Institut de Recherches Interdisciplinaires
et de Développements en Intelligence Artificielle

Convergence Behavior of the Fully Informed Particle Swarm Optimization Algorithm

Marco A. Montes de Oca and Thomas Stützle

IRIDIA – Technical Report Series

Technical Report No.
TR/IRIDIA/2008-011

April 2008

To appear in the Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2008)

IRIDIA – Technical Report Series
ISSN 1781-3794

Published by:

IRIDIA, *Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle*
Université Libre de Bruxelles
Av F. D. Roosevelt 50, CP 194/6
1050 Bruxelles, Belgium

Technical report number TR/IRIDIA/2008-011

Revision history:

TR/IRIDIA/2008-011.001 April 2008

Convergence Behavior of the Fully Informed Particle Swarm Optimization Algorithm

Marco A. Montes de Oca mmontes@ulb.ac.be
Thomas Stützle stuetzle@ulb.ac.be
IRIDIA, Université Libre de Bruxelles, Brussels, Belgium

April 2008

Abstract

The fully informed particle swarm optimization algorithm (FIPS) is very sensitive to changes in the population topology. The velocity update rule used in FIPS considers all the neighbors of a particle to update its velocity instead of just the best one as it is done in most variants. It has been argued that this rule induces a random behavior of the particle swarm when a fully connected topology is used. This argument could explain the often observed poor performance of the algorithm under that circumstance.

In this paper we study experimentally the convergence behavior of the particles in FIPS when using topologies with different levels of connectivity. We show that the particles tend to search a region whose size decreases as the connectivity of the population topology increases. We therefore put forward the idea that spatial convergence, and not a random behavior, is the cause of the poor performance of FIPS with a fully connected topology. The practical implications of this result are explored.

1 Introduction

Particle swarm optimization (PSO) is a swarm intelligence optimization technique that was inspired by the behavior of flocks of birds [5]. In a PSO algorithm, particles (i.e., potential solutions to an optimization problem) move in the search space with a velocity that is updated every iteration. The performance of the algorithm depends on the way the particles move in the search space and a large body of research in the field has been devoted to the analysis and proposal of different movement rules (see [2, 10] for recent accounts of PSO research).

In the traditional PSO algorithm, a particle is attracted toward the best position it has visited (with respect to an objective function) and toward the best position found by the particles in its neighborhood (we will refer to this strategy as *best-of-neighborhood*). Neighborhood relations are usually defined in advance through a population topology which can be defined by a graph $G = \{V, E\}$, where each vertex in V corresponds to a particle in the swarm and each edge in E establishes a neighbor relation between a pair of particles.

A prominent alternative to the best-of-neighborhood velocity update strategy is the one used in the fully informed particle swarm optimization algorithm (FIPS) [8]. In FIPS, a particle is attracted to the best positions of all the particles in its neighborhood, not only to the best one. A study on the performance attained by algorithms using both strategies and different population topologies was carried out by Mendes [7, 8]. In these and subsequent studies, FIPS with a fully connected topology, i.e., when each particle has all the particles in the swarm as neighbors, has exhibited a particularly bad performance in comparison with the one obtained with other topologies. It has been argued that this happens because the simultaneous attraction to multiple points "confound" the particles, provoking a random behavior of the particle swarm [6, 10]. A random behavior of the particles could explain FIPS's performance with a fully connected topology, specially in high-dimensional search spaces.

In this paper, we conduct a series of experiments in order to test the argument mentioned above. Our results show that the cloud of particles tends to explore a region whose size decreases as the topology connectivity increases. If a fully connected topology is used, the search region is located near the swarm's centroid. This behavior, which is intensified when large population sizes are used, can impair the algorithm's exploratory capabilities. The reason for FIPS's poor performance with fully connected topologies is thus not a random behavior, but spatial convergence. These results complement some of the findings derived from a theoretical analysis of the sampling distribution of some PSO algorithms [9]. The implications of these results, from a practical perspective, are explored in a second series of experiments.

2 Fully Informed Particle Swarm Optimization Algorithm

The fully informed particle swarm optimization algorithm (FIPS) was proposed by Mendes et al. [8] as an alternative to PSO algorithms that use the best-of-neighborhood velocity update strategy. In FIPS, the update of the velocity and position of a particle i over dimension j is as follows

$$v_{i,j}^{t+1} = \chi \left[v_{i,j}^{t} + \frac{1}{K_i} \sum_{n=1}^{K_i} \mathrm{U}(0,\varphi)(p_{\mathrm{N}_i(n),j}^{t} - x_{i,j}^{t}) \right] , \qquad (1)$$

where χ is a constriction factor, K_i is the number of particles in the neighborhood of particle i, $\mathrm{U}(0,\varphi)$ is a uniformly distributed random number in the range $[0,\varphi)$ where φ is called an acceleration coefficient, $\mathrm{N}_i(n)$ is a function that returns the index of the n-th neighbor of particle i, and $p_{\mathrm{N}_i(n),j}^{t}$ is the j-th component of the previous best position of the n-th neighbor of i.

The constriction factor χ is used in order to avoid an "explosion" of the particles' velocity. Clerc and Kennedy [1] found the relation

$$\chi = \frac{2k}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|} ,$$

where $k \in [0,1]$, and $\varphi > 4$, to compute it.

3 Expected Behavior

Previous theoretical studies of the behavior of particle swarm algorithms have made some simplifying assumptions in order to facilitate mathematical analyses. These assumptions are usually stagnation (i.e., no solution improvement over time) and the absence of stochasticity. In this section, we make the same assumptions in order to derive the expected behavior of the particles in FIPS. In the following section we evaluate the extent to which the conclusions derived from our analysis hold in the fully stochastic version of the algorithm.

It is possible to derive the expected behavior of a particle using FIPS's velocity and position update rules by substituting $\mathrm{U}(0,\varphi)$ by its expected value $\varphi/2$ in Equation 1. By doing so, we obtain

$$v_{i,j}^{t+1} = \chi \left[v_{i,j}^{t} + \frac{\varphi}{2K_i} \sum_{n=1}^{K_i} (p_{\mathrm{N}_i(n),j}^{t} - x_{i,j}^{t}) \right] . \tag{2}$$

In a stagnation phase, $p_{\mathrm{N}_i(n),j}^{t} = p_{\mathrm{N}_i(n),j}\ \forall t$. Thus we obtain

$$v_{i,j}^{t+1} = \chi \left[v_{i,j}^{t} + \frac{\varphi}{2K_i} \sum_{n=1}^{K_i} (p_{\mathrm{N}_i(n),j} - x_{i,j}^{t}) \right] , \tag{3}$$

which can be simplified to

$$v_{i,j}^{t+1} = \chi \left[v_{i,j}^{t} + \frac{\varphi}{2} (\bar{p}_{\mathrm{N}_i(*),j} - x_{i,j}^{t}) \right] , \tag{4}$$

where

$$\bar{p}_{\mathrm{N}_i(*),j} = \frac{1}{K_i} \sum_{n=1}^{K_i} p_{\mathrm{N}_i(n),j} . \tag{5}$$

For convenience, let us define the swarm's centroid in a componentwise fashion as

$$c_j = \frac{1}{N} \sum_{i=1}^{N} p_{i,j} , \tag{6}$$

where N is the size of the particle swarm.

Equations 4 and 5 tell us that in the absence of solution improvement it is expected that a particle is attracted toward the centroid of its neighbors' previous best positions. It is also apparent that the size of a particle's neighborhood has a direct influence on the expected behavior of a particle. The larger the neighborhood size, that is, as $K_i \rightarrow N$, the closer $\bar{p}_{\mathrm{N}_i(*),j}$ will be to the corresponding component of the swarm's centroid c_j. With a fully connected topology all the particles are then expected to move toward the swarm's centroid which effectively makes the algorithm to search around a single point. This behavior results in a bias that, depending on the topography of the objective function, could render the algorithm ineffective. Additionally, if a swarm has a large and highly connected population, the bias toward the neighborhood centroid effectively becomes a bias toward the center of the initialization region.

4 Validation Experiments

Two series of experiments are designed to test the validity of the analysis with the fully stochastic FIPS algorithm. The first set of experiments consists in observing, through some auxiliary measures, the particles' behavior on a flat objective function. These experiments are aimed at observing the behavior of the algorithm under forced stagnation. In the second set of experiments, an inverted parabola over a constrained range is used as a minimization objective function so that the particles' previous best positions move, presumably, toward one of the two local optima, that is, stagnation is no longer forced.

In all the experiments, FIPS is run with its most commonly used parameter settings, that is, $\varphi = 4.1$ and $\chi = 0.7298$. The behavior of FIPS with different swarm and neighborhood sizes is investigated. Swarms of 10, 100 and 1000 particles are used. Each particle in these swarms had {10, 5, 3}, {100, 51, 25, 13, 7, 3}, and {1000, 501, 251, 125, 63, 31, 15, 7, 3} particles as neighbors respectively. The neighborhood size is expressed as the number of particles to which a given particle is connected (including itself). Our data is based on 100 independent runs of up to 1000 iterations each.

The auxiliary measures used in our experiments are the following:

Average distance between particles. This measure is computed using the expression $\mathrm{D} = \frac{2}{(S+1)S} \sum_{j=i}^{S} \sum_{i=1}^{S} |\boldsymbol{x_i} - \boldsymbol{x_j}|$, where $|\cdot|$ is the Euclidean norm and S is the size of the swarm. The average distance between particles is used as a measure of spatial convergence of the particles.

Average particle speed. Computed as $\mathrm{AS} = \frac{1}{S} \sum_{j=i}^{S} |\boldsymbol{v_i}|$, where $\boldsymbol{v_i}$ is the velocity vector of particle i. The average speed is used as a measure of the length of the average step size of the particles in the swarm.

Swarm's Centroid to the Origin. Computed as $\mathrm{C2O} = \left| \frac{1}{S} \sum_{i=1}^{S} \boldsymbol{x_i} \right|$. This measure is useful for tracking the movement of the particle swarm relative to the origin.

4.1 Experiments with a flat objective function

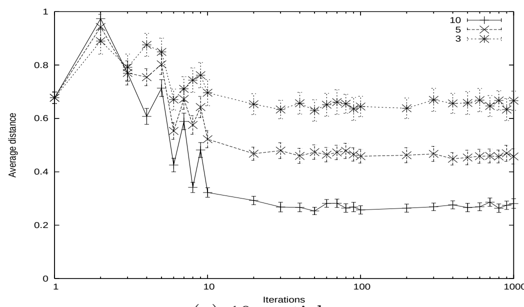
For this series of experiments, the objective function is

$$f(x) = 0, \tag{7}$$

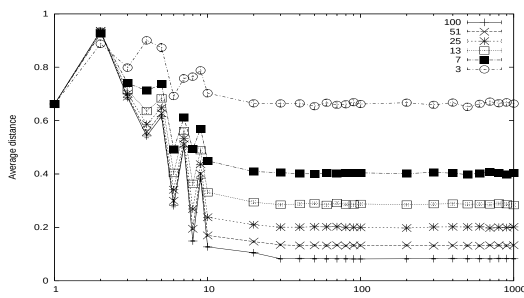
where $x \in \mathbb{R}$. The particles are initialized randomly in the range $[-1, 1]$ with a random initial velocity within the same range. The condition for accepting a new best position is a strict improvement; therefore, the particles' previous best positions remain where they are initialized.

Figure 1 shows the development of the average distance between particles over time with swarms and neighborhoods of different size. The size of the confidence intervals in the experiments with 100 and 1000 particles are close to zero.

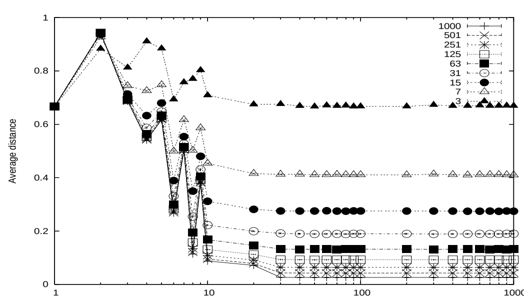
The distance between particles reaches a stable value after some iterations. This means that the particles end up congregating in a region of constant size. The size of this region decreases as the size of the particles' neighborhood increases. The final size of this region depends on both the particles' neighborhood



(a) 10 particles



(b) 100 particles



(c) 1000 particles

Figure 1: Development of the average distance between particles over time on a one-dimensional flat objective function. The plots show the sample mean with 95% confidence intervals. Each line corresponds to a different neighborhood size.

size and the swarm size, although the influence of the swarm size decreases when a particle's neighborhood is small. In particular, the average distance between particles, with neighborhood size of 3 is the same regardless of the size of the swarm. Note also that this stable value is very close to the one obtained after initialization, which corresponds to the expected distance between any two uniformly distributed random points in a segment of length equal to 2 (i.e., the length of our initialization range) [3]. This suggests that, in this case, the algorithm does not lose its exploration capabilities.

Figure 1 tells us that the particles end up *on average* at a certain distance from each other, but it does not tell us whether they are moving or not; however, since the particles' velocity ultimately depends on their separation, distance and velocity are correlated. Thus the larger the particles' neighborhoods, the lower the average final velocity. The distance of the particles' centroid to the origin (the plots are not shown for the sake of conciseness) is constant across iterations, meaning that the interparticle distances oscillate around a stable point with constant amplitude, the magnitude of which decreases as the particles' neighborhoods increase in size.

The experimental results shown in this section confirm that the fully stochastic FIPS algorithm in a stagnation phase behaves as explained in the previous section. The particles explore near the centroid of the particles' previous best positions and the population topology determines the size of the exploration area. The larger a particle's neighborhood, the smaller the region in which the particles oscillate.

4.2 Experiments with an inverted parabola

The objective function used in the following set of experiments is

$$f(x) = -x^2, \tag{8}$$

where $x \in [-10, 10]$. As in the previous experiments, the particles are initialized randomly in the range $[-1, 1]$ with a random initial velocity within the same range. If a particle's next position falls outside the boundaries, the velocity value is left unchanged but the particle's position is set to the boundary value. This objective function permits the observation of the bias toward the centroid of the previous best positions when these can move according to the underlying objective function topography (see Fig. 2). If both the distance between particles and the distance of the swarm's centroid to the origin are small (e.g., within the initialization range), it is possible to conclude that a bias is present. Furthermore, it is possible to measure the severity of this bias by estimating the probability of this phenomenon to happen.

Figure 3 shows the development of the average distance between particles over time on the inverted parabola problem. The average distance between particles decreases as the neighborhood size increases. In other words, the more connected the particle swarm, the more likely it is that the particles remain close to each other. In fact, when the fully connected topology is used, the swarm collapses into a single point. These results agree with the ones obtained with the flat objective function.

The ratio between the number of runs in which the swarm's centroid remains within the initialization range and the total number of runs is used as an estimate

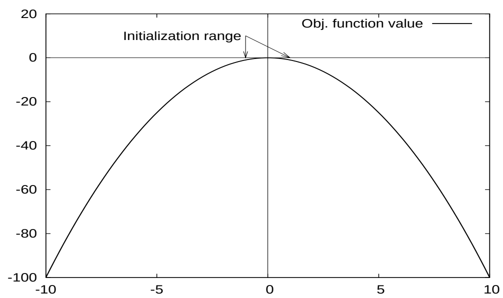


Figure 2: The inverted parabola provides two local optima at the extremes of the range $[-10, 10]$. The particles' initialization range is shown. If a bias exists, the particle swarm should remain tight with its centroid close to the origin.

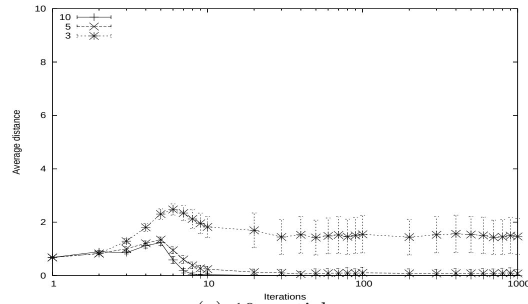
of the probability with which the swarm can get trapped. The development of this estimate over time is shown in Figure 4. The graph shows only the results obtained with fully connected topologies which are the ones that make the algorithm exhibit the strongest bias.

With 10 particles, the particle swarm moves away from the initialization range. In this case, the swarm remains compact but it moves toward one of the two local optima. With 100 particles, the probability of being trapped within the initialization range is 0.03. With 1000 particles, this probability increases to 0.12. Strikingly, a swarm of 1000 particles is not capable of finding a local optimum of a constrained inverted parabola with a probability of 0.12! These results highlight the effect of the swarm size on the severity of the bias toward the swarm's centroid. This result follows from the fact that increasing the number of particles strengthens the attraction of the particles toward the swarm's centroid.

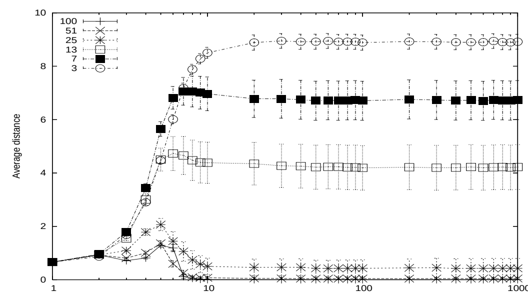
These results clearly show that the number of particles considered for computing a particle's velocity have a major impact on the exploratory capabilities of FIPS. The larger the neighborhood, the greater the attraction of a particle to explore on a small region close to the centroid of its neighbors' previous best positions. If this region happens to be of a lower quality than that of the particles' previous best positions (as in the inverted parabola problem), the probability of stagnation increases.

5 Optimization Experiments

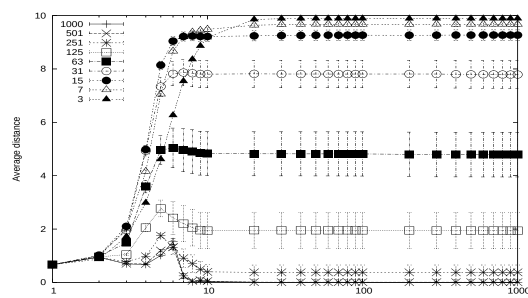
In this section, we explore the implications of our results on the performance of FIPS as an optimization algorithm. Based on the results obtained so far, it is possible to anticipate that FIPS using a fully connected population topology will have a better performance than with less connected topologies during the first iterations before the particles converge, on problems or settings where the population is uniformly distributed over a "funnel" of the objective function topography. The word funnel is used as a metaphor to refer to a region in the search space where a number of local optima are clustered together and where a trajectory that moves from one local optimum to another in a strictly descending way ends up necessarily at the local optimum of minimum value [11].



(a) 10 particles



(b) 100 particles



(c) 1000 particles

Figure 3: Development of the average distance between particles over time on an inverted parabola objective function. The plots show the sample mean and 95% confidence intervals. Each line corresponds to a different neighborhood size.

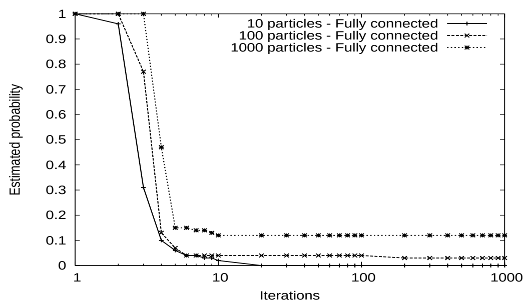


Figure 4: Estimated probability of a swarm being trapped within its initialization range due to a bias toward the swarm's centroid.

We expect such a difference in performance due to the bias toward the swarm's centroid that a fully connected topology induces. In a single-funnel problem, where the global optimum is located at its bottom, the bias will drive the particles close to the optimum before converging. In a multiple-funnel problem, the same bias will hinder the exploratory capabilities of the algorithm because no possibility of escaping from a low quality funnel is possible. In this case, the performance of the algorithm will greatly depend on the initialization conditions.

5.1 Single-funnel problems

Two single-funnel benchmark problems are used to test whether FIPS with a fully connected topology could take advantage of the bias toward the swarm's centroid. The problems are the sphere and Rastrigin functions. Both functions are shifted in order to discard the effects of any possible bias toward the origin of the coordinate system. Their definitions are:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} (x_i - 100)^2, \tag{9}$$

and

$$f(\boldsymbol{x}) = 10n + \sum_{i=1}^{n} ((x_i - 5.12)^2 - 10\cos(2\pi(x_i - 5.12))). \tag{10}$$

In the first case, the initialization range is $x_i \in [0, 200]$ and in the second case, it is $x_i \in [0, 10.24]$. The problems are ten-dimensional, that is, $n = 10$. The first problem is a unimodal problem while the second one is multimodal.

Tables 1 and 2 show the mean solution value with 95% confidence intervals[1] obtained on the sphere and Rastrigin problems respectively. The data is organized by population size, iteration number and neighborhood size.

Regardless of whether the problem is unimodal or multimodal, the results share a number of common characteristics. After 10 iterations, the best results are obtained with swarms with highly connected topologies. It is important to note that in the 1000 particles case, the difference between the best results obtained with large neighborhood sizes (1000, 501 and 251 particles) are not

[1] Numbers that are less than or equal to 1×10^{-20} are indicated with 0.

Table 1: Mean solution quality over time with 95% confidence intervals on the shifted 10-dimensional sphere problem using different swarm and neighborhood sizes.

Pop. Size	Iteration	Neighbors	Solution
10	10	10	**3825.18 (±289.52)**
		5	5403.56 (±371.16)
		3	8278.63 (±526.83)
	100	10	39.598 (±15.540)
		5	**3.906 (±1.949)**
		3	9.487 (±2.699)
	1000	10	3.803 (±3.728)
		5	1.1e-4 (±2e-4)
		3	**1.7e-12 (0)**
100	10	100	**338.39 (±21.94)**
		51	417.37 (±23.43)
		25	682.89 (±38.26)
		13	1240.99 (±67.75)
		7	2158.17 (±121.45)
		3	4468.14 (±256.06)
	100	100	0.001 (±0.002)
		51	0.001 (±6e-4)
		25	6e-5 (±3e-5)
		13	**7e-6 (±1e-6)**
		7	3e-4 (±3e-5)
		3	1.686 (±0.13)
	1000	100	0.001 (±0.002)
		51	3e-8 (±5e-8)
		25	**0 (0)**
		13	**0 (0)**
		7	**0 (0)**
		3	**0 (0)**
1000	10	1000	122.53 (±6.700)
		501	**116.73 (±5.781)**
		251	123.40 (±5.854)
		125	148.48 (±7.491)
		63	201.79 (±10.573)
		31	314.22 (±15.337)
		15	586.36 (±30.903)
		7	1263.00 (±67.451)
		3	2526.23 (±131.71)
	100	1000	**8e-11 (±0)**
		501	6e-10 (±7e-11)
		251	1e-8 (±6e-9)
		125	6e-8 (±1e-8)
		63	4e-7 (±8e-8)
		31	5e-7 (±7e-8)
		15	1e-6 (±7e-8)
		7	1e-4 (±9e-6)
		3	0.784 (±0.053)
	1000	1000	**0 (±0)**
		501	**0 (±0)**
		251	**0 (±0)**
		125	**0 (±0)**
		63	**0 (±0)**
		31	**0 (±0)**
		15	**0 (±0)**
		7	**0 (±0)**
		3	**0 (±0)**

Table 2: Mean solution quality over time with 95% confidence intervals on the shifted 10-dimensional Rastrigin problem using different swarm and neighborhood sizes.

Pop. Size	Iteration	Neighbors	Solution
10	10	10	**81.528 (±2.279)**
		5	91.448 (±2.400)
		3	102.40 (±2.861)
	100	10	**19.307 (±2.769)**
		5	28.838 (±2.301)
		3	40.735 (±1.561)
	1000	10	11.640 (±1.035)
		5	7.418 (±0.689)
		3	**6.437 (±0.752)**
100	10	100	**45.585 (±1.512)**
		51	46.648 (±1.544)
		25	50.846 (±1.489)
		13	56.878 (±1.541)
		7	63.667 (±1.711)
		3	78.371 (±1.792)
	100	100	19.402 (±3.257)
		51	14.936 (±3.027)
		25	15.826 (±1.536)
		13	**13.305 (±1.145)**
		7	13.389 (±1.004)
		3	27.500 (±0.974)
	1000	100	15.251 (±2.855)
		51	8.807 (±2.585)
		25	10.359 (±1.147)
		13	4.457 (±0.485)
		7	2.501 (±0.221)
		3	**2.115 (±0.210)**
1000	10	1000	31.363 (±1.113)
		501	**30.491 (±1.061)**
		251	30.562 (±1.005)
		125	31.939 (±1.144)
		63	32.935 (±0.949)
		31	35.573 (±1.157)
		15	42.431 (±1.189)
		7	49.273 (±1.488)
		3	60.557 (±1.487)
	100	1000	**0.002 (±0.005)**
		501	0.009 (±0.019)
		251	0.005 (±0.007)
		125	0.040 (±0.029)
		63	0.647 (±0.235)
		31	4.050 (±0.559)
		15	4.780 (±0.406)
		7	6.670 (±0.383)
		3	19.193 (±0.730)
	1000	1000	**0.002 (±0.004)**
		501	0.009 (±0.019)
		251	0.005 (±0.007)
		125	0.039 (±0.029)
		63	0.420 (±0.146)
		31	2.355 (±0.427)
		15	1.536 (±0.193)
		7	0.721 (±0.103)
		3	0.482 (±0.095)

statistically significant (the confidence intervals overlap). With 10 and 100 particles and after 100 iterations, the best results are not obtained with the fully connected topology but with less connected ones. After 1000 iterations, the best results are obtained with the least connected topologies. With 1000 particles and a fully connected topology very good results are obtained after 1000 iterations.

The results obtained in these experiments agree with our expectations. A highly connected topology coupled with a single-funnel topography (regardless of whether there are local minima or not) provides the algorithm the opportunity of finding good solutions during the first iterations. The solution improvement then stagnates due to the spatial convergence of the particles. With large populations, highly connected topologies obtain very good results even in long runs. We think this is the case because the swarm's centroid gets closer to the center of the initialization range as the size of the neighborhood increases. In our experiments, being close to the center of the initialization range is equivalent to being close to the location of the global optima of the test problems used.

5.2 Multiple-funnel problems

A multiple-funnel benchmark problem is used for this set of experiments. It is the Schwefel's sine root function, the mathematical definition of which is

$$f(\boldsymbol{x}) = 418.9829n - \sum_{i=1}^{n} \left(x_i \sin(\sqrt{|x_i|}) \right) , \tag{11}$$

with $x_i \in [-500, 500]$ and $n = 10$. If the boundary constraint is violated, the maximum value of a single-precision floating-point number is returned as the value of the objective function. This makes particles exploring outside the search range of interest to go back to it. In this case, the function is not shifted because the global optimum is not located in the origin of the coordinate system.

Table 3 shows the mean solution value with 95% confidence intervals obtained on the sine root problem. After 10 iterations the best performance is obtained with a fully connected topology. At 100 and 1000 iterations, the best performance is obtained with the least connected topology. With large swarms (100 and 1000 particles), the solution quality obtained when the algorithm uses a highly connected topology does not improve over time. The best solution found during the very first iterations of the algorithm remains the best even after 1000 iterations. This suggests that, as expected, the performance of the algorithm with highly connected topologies greatly depends on the initialization conditions. The strong attraction between the particles with highly connected topologies hinders the exploratory capabilities of the algorithm, effectively stagnating the solution improvement process.

5.3 Restarts

A simple strategy to deal with solution improvement stagnation is to restart the optimization algorithm from new initial conditions [4]. If premature spatial convergence is the cause of the solution improvement stagnation observed when FIPS uses a fully connected topology, then using a dynamic restart mechanism, whereby the particles are reinitialized every time they concentrate in a small

Table 3: Mean solution quality over time with 95% confidence intervals on the 10-dimensional Schwefel's sine root problem using different swarm and neighborhood sizes.

Pop. Size	Iteration	Neighbors	Solution
10	10	10	**3022.23 (±57.732)**
		5	3027.68 (±59.462)
		3	3034.44 (±62.204)
	100	10	2708.88 (±74.850)
		5	2558.02 (±58.187)
		3	**2339.65 (±45.720)**
	1000	10	2488.50 (±55.161)
		5	1630.11 (±61.721)
		3	**544.21 (±49.640)**
100	10	100	**2559.40 (±38.665)**
		51	2570.95 (±43.054)
		25	2581.18 (±43.172)
		13	2577.70 (±41.942)
		7	2571.18 (±42.598)
		3	2569.61 (±45.356)
	100	100	2559.40 (±38.665)
		51	2570.95 (±43.054)
		25	2581.18 (±43.172)
		13	2575.19 (±41.486)
		7	2410.92 (±35.819)
		3	**1837.44 (±37.650)**
	1000	100	2559.40 (±38.665)
		51	2570.95 (±43.054)
		25	2578.47 (±42.704)
		13	2522.70 (±35.426)
		7	1596.58 (±46.349)
		3	**113.81 (±16.275)**
1000	10	1000	**2035.64 (±53.400)**
		501	2071.30 (±44.807)
		251	2098.28 (±38.063)
		125	2124.09 (±41.101)
		63	2137.04 (±41.938)
		31	2150.55 (±42.482)
		15	2175.96 (±41.881)
		7	2179.08 (±41.113)
		3	2156.53 (±39.736)
	100	1000	2035.64 (±53.400)
		501	2071.30 (±44.807)
		251	2098.28 (±38.063)
		125	2124.09 (±41.101)
		63	2137.04 (±41.938)
		31	2150.55 (±42.482)
		15	2175.96 (±41.881)
		7	2107.84 (±36.238)
		3	**1544.90 (±29.845)**
	1000	1000	2035.64 (±53.400)
		501	2071.30 (±44.807)
		251	2098.28 (±38.063)
		125	2124.09 (±41.101)
		63	2137.04 (±41.938)
		31	2150.55 (±42.482)
		15	2175.96 (±41.881)
		7	1207.13 (±28.137)
		3	**4.563 (±3.428)**

region, should improve the performance of the algorithm in the long run. However, the effectiveness of using spatial convergence as a restart criterion will depend on the selected distance threshold. In cases in which finding very high quality solutions requires an algorithm to make very small steps in the search space, a large threshold will provoke premature restarts. In general, the restart criteria should be tuned for the specific problem at hand.

A series of experiments are carried out in order to verify the effectiveness of a dynamic restart mechanism on the performance of FIPS with fully connected topologies. The experimental setup is the same that was used in the previous sections. Two extra parameters are used in these series of experiments: the restart criterion and whether the best solution before a restart is carried over to the next run to bias the search or not. The restart criterion is set to be when the average distance between particles drops below 1% and 10% of the expected distance between randomly generated points in a 10-dimensional space (which was estimated using a Monte Carlo method). The experiments were run with and without the best solution being carried over restarts. When a solution is carried over to the next run, it is used as the previous best position of one particle of the swarm.

Table 4 shows the results obtained with the Rastrigin (single-funnel) and Schwefel's sine root (multiple-funnel) functions[2]. The reported results correspond to the solution value after 100 and 1000 iterations only. The results obtained after 10 iterations are not shown because they do not show any difference with respect to the results obtained without restarts. This is due to the fact that after 10 iterations the restart criteria are never met. The symbols $<$, $>$, $=$ are used to indicate whether the results are less than, greater than, or equal to (not necessarily in a statistical sense) the ones obtained without restarts.

[2]The results obtained with the sphere function are not shown due to space constraints. Nevertheless they are commented in the text.

Table 4: Mean solution quality over time with 95% confidence intervals obtained with FIPS and fully connected topologies with restarts. The best result for different restart criteria is shown in boldface.

Problem	Pop. Size	Iteration	Restart Criterion	Solution Quality: Without solution carried over		Solution Quality: With solution carried over	
Rastrigin	10	100	1%	**20.133 (±2.709)**	¿	**20.406 (±2.731)**	¿
			10%	38.322 (±1.841)	¿	37.837 (±2.070)	¿
		1000	1%	**7.179 (±0.451)**	¡	**6.530 (±0.428)**	¡
			10%	20.374 (±0.919)	¿	19.997 (±0.869)	¿
	100	100	1%	18.386 (±3.289)	¡	18.247 (±3.305)	¡
			10%	**7.671 (±1.753)**	¡	**8.865 (±2.067)**	¡
		1000	1%	11.963 (±3.099)	¡	11.591 (±3.114)	¡
			10%	**1.485 (±0.118)**	¡	**1.552 (±0.108)**	¡
	1000	100	1%	**0.002 (±1e-4)**	=	**0.001 (±1e-4)**	¡
			10%	0.209 (±0.016)	¿	0.204 (±0.014)	¿
		1000	1%	**0.001 (±5e-5)**	¡	**0.001 (±5e-5)**	¡
			10%	0.106 (±0.006)	¿	0.107 (±0.005)	¿
Schwefel	10	100	1%	**2710.24 (±74.464)**	¿	**2703.32 (±74.905)**	¡
			10%	2738.53 (±72.355)	¿	2748.71 (±66.240)	¿
		1000	1%	**2242.07 (±63.576)**	¡	**2280.60 (±62.145)**	¡
			10%	2273.13 (±63.167)	¡	2281.43 (±56.363)	¡
	100	100	1%	2559.40 (±38.665)	=	2559.40 (±38.665)	=
			10%	**2399.58 (±41.582)**	¡	**2404.77 (±42.176)**	¡
		1000	1%	2559.40 (±38.665)	=	2559.40 (±38.665)	=
			10%	**2059.08 (±34.477)**	¡	**2075.46 (±32.067)**	¡
	1000	100	1%	2035.64 (±53.400)	=	2035.64 (±53.400)	=
			10%	**1877.89 (±46.237)**	¡	**1873.50 (±45.933)**	¡
		1000	1%	2035.64 (±53.400)	=	2035.64 (±53.400)	=
			10%	**1628.35 (±36.083)**	¡	**1627.99 (±35.642)**	¡

In all cases, the differences between the results obtained with and without the best solution carried over restarts are not statistically significant. Particles in FIPS are not only attracted toward the best particle, so carrying over the best solution does not provide any significant benefit.

On the sphere function, the use of restarts is beneficial only in the case of a swarm of 10 particles and a restart criterion of 1%. In all other cases, the use of restarts is detrimental. In the sphere problem, finding good solutions requires the algorithm to make very small steps near the optimum. These results constitute an example of the detrimental effects of using restart criteria that make the algorithm restart prematurely. On the Rastrigin problem, restarting with a criterion of 1% always improves the solution quality obtained after 1000 iterations. On the Schwefel problem, improvement is always attained with a criterion of 10% after 1000 iterations. It is worth noting that the best results obtained with restarts are still far from the results obtained with other population topologies. These results further confirm the hypothesis that the spatial convergence of the particles is the reason for the solution improvement stagnation observed in FIPS with a fully connected topology.

6 Conclusions

In the fully informed particle swarm optimization algorithm (FIPS), each particle uses the information from all its neighbors to update its velocity. The structure of the population topology has, therefore, a critical impact on the behavior of the algorithm which in turn affects its performance as an optimizer. Previous studies have found that when a fully connected topology is used, the performance of FIPS is considerably reduced. It has been argued that this happens because the simultaneous influence of all the particles in the swarm "confounds" the particle that is updating its velocity, provoking a random behavior of the particle swarm [6, 10].

In this paper, we carried out an analysis of the expected behavior of a particle in FIPS under stagnation and tested empirically the validity of the conclusions drawn from this analysis. It turns out that the observed performance of the algorithm with fully connected topologies is a consequence of the spatial convergence of the particles during the search, rather than a random behavior. With highly connected topologies, the particles explore in a region close to the centroid of the swarm. These results complement the findings derived from an analysis of the sampling distribution of some PSO algorithms [9], where it has been shown that FIPS becomes more and more stable (with respect to the first four moments of its sampling distribution under stagnation) as the size of the particles' neighborhoods increases.

Not surprisingly, the final performance of the algorithm depends on the objective function topography. If the population is evenly distributed around a "funnel" in the landscape, the bias will produce good results, especially during the first iterations of the algorithm. When the region where the particles explore happens to be of lower quality than the particles' previous best positions, the algorithm is in high risk of becoming trapped and being unable to improve any further. In this case, increasing the diversity of the population by making it larger, does not work because the larger the population, the stronger is the bias toward the centroid of the swarm. Enhancing the exploratory capabilities of the algorithm by using dynamic restarts provides some benefits but these are problem-dependent.

7 Acknowledgments

Marco A. Montes de Oca is funded by the Programme Alβan, the European Union Programme of High Level Scholarships for Latin America, scholarship No. E05D054889MX, and by the ANTS project, an *Action de Recherche Concertée* funded by the Scientific Research Directorate of the French Community of Belgium. Thomas Stützle acknowledges support from the F.R.S-FNRS of the French Community of Belgium of which he is a Research Associate.

References

[1] M. Clerc and J. Kennedy. The particle swarm–explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, 2002.

[2] A. P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, Chichester, West Sussex, England, 2005.

[3] B. Gaboune, G. Laporte, and F. Soumis. Expected distances between two uniformly distributed random points in renctangles and rectangluar parallelpipeds. *The Journal of the Operational Research Society*, 44(5):513–519, 1993.

[4] H. H. Hoos and T. Stützle. *Stochastic Local Search: Foundations and Applications*. Morgan Kaufmann, San Francisco, CA, USA, 2004.

[5] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, pages 1942–1948, Piscataway, NJ, USA, 1995. IEEE Press.

[6] J. Kennedy and R. Mendes. Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 36(4):515–519, 2006.

[7] R. Mendes. *Population Topologies and Their Influence in Particle Swarm Performance*. PhD thesis, Escola de Engenharia, Universidade do Minho, 2004.

[8] R. Mendes, J. Kennedy, and J. Neves. The fully informed particle swarm: Simpler, maybe better. *IEEE Transactions on Evolutionary Computation*, 8(3):204–210, 2004.

[9] R. Poli. On the moments of the sampling distribution of particle swarm optimisers. In *Proceedings of the workshop on particle swarm optimization: the second decade. Genetic and Evolutionary Computation Conference (GECCO)*, pages 2907–2914, New York, NY, USA, 2007. ACM Press.

[10] R. Poli, J. Kennedy, and T. Blackwell. Particle swarm optimization. An overview. *Swarm Intelligence*, 1(1):33–57, 2007.

[11] A. M. Sutton, D. Whitley, M. Lunacek, and A. Howe. PSO and Multi-Funnel Landscapes: How cooperation might limit exploration. In M. Cattolico, editor, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 75–82, New York, NY, USA, 2006. ACM Press.