Using Neighbourhoods with the Guaranteed Convergence PSO

E.S. Peer, F. van den Bergh, A.P. Engelbrecht
Department of Computer Science
University of Pretoria
South Africa
espeer@cs.up.ac.za, fvdbergh@cs.up.ac.za, engel@cs.up.ac.za

Abstract— **The standard Particle Swarm Optimiser (PSO) may prematurely converge on suboptimal solutions that are not even guaranteed to be local extrema. The guaranteed convergence modifications to the PSO algorithm ensure that the PSO at least converges on a local extremum at the expense of even faster convergence. This faster convergence means that less of the search space is explored reducing the opportunity of the swarm to find better local extrema. Various neighbourhood topologies inhibit premature convergence by preserving swarm diversity during the search. This paper investigates the performance of the Guaranteed Convergence PSO (GCPSO) using different neighbourhood topologies and compares the results with their standard PSO counterparts.**

I. Introduction

The Particle Swarm Optimiser (PSO) was first introduced by Kennedy and Eberhart [1], [2]. It can be applied to virtually any problem that can be expressed in terms of an objective function for which extrema must be found.

The PSO algorithm is iterative and involves initialising a number of vectors (called particles) randomly within the search space of the objective function. These particles are collectively known as the swarm. Each particle represents a potential solution to the problem expressed by the objective function. During each time step the objective function is evaluated to establish the fitness of each particle using its position as input. Fitness values are used to determine which positions in the search space are better than others. Particles are then made to "fly" through the search space being attracted to both their personal best position as well as the best position found by the swarm so far.

The particles are "flown" through the search space by updating the position of the i^{th} particle at time step t according to the following equation:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t) \tag{1}$$

where $\mathbf{x}_i(t)$ and $\mathbf{v}_i(t)$ are vectors representing the current position and velocity respectively. Assuming particles of dimension $j \in 1 \ldots n$, the velocity updates are governed by the following equation [3]:

$$\begin{aligned} v_{i,j}(t+1) &= wv_{i,j}(t) + \\ &\quad c_1 r_{1,j}(y_{i,j} - x_{i,j}(t)) + \\ &\quad c_2 r_{2,j}(\hat{y}_j - x_{i,j}(t)) \end{aligned} \tag{2}$$

where $0 \leq w < 1$ is an inertia weight determining how much of the particle's previous velocity is preserved, c_1 and c_2 are two positive acceleration constants, $r_{1,j}$ and $r_{2,j}$ are two uniform random sequences sampled from $U(0,1)$, $\mathbf{y_i}$ is the personal best position found by the i^{th} particle and $\hat{\mathbf{y}}$ is the best position found by the entire swarm so far.

The effect of neighbourhoods on the PSO have been studied [4], [5]. It has been found that neighbourhoods can improve performance, particularly on multimodal functions. Neighbourhoods preserve swarm diversity, inhibiting early convergence to suboptimal solutions. Despite this, the standard PSO is not guaranteed to converge on a local extremum. The Guaranteed Convergence PSO (GCPSO) [6], [7] guarantees local convergence at the expense of faster convergence which reduces swarm diversity. The purpose of this paper is to determine whether the GCPSO also benefits from neighbourhoods. Two neighbourhood topologies were applied to the GCPSO and tested on a variety of benchmark functions. This paper presents the results of the experiments conducted and compares these results with the standard PSO.

The convergence behaviour of the standard PSO is discussed in Section II. Section III provides an overview of the GCPSO. Next, an overview of Neighbourhoods is provided in Section IV. Benchmark functions to measure the performance of the PSO are given in Section V. The experiment configuration can be found in Section VI. Results of the experiments are presented in Section VII. Finally, some concluding remarks are made in Section VIII.

II. Convergence of the standard PSO

The velocity update in Equation (2) consists of three terms. The first is the inertia term. The second is known as the cognition component because it

models the particle's personal knowledge about the search space. The third term is known as the social component because it models the interaction with the other particles.

To understand the convergence behaviour of the standard PSO, consider the scenario where the objective function is unimodal. As a particle moves towards $\hat{\mathbf{y}}$ its personal best position must improve due to the unimodal landscape. Thus, on unimodal functions, $\mathbf{x}_i = \mathbf{y}_i$ giving a zero cognition component. All particles are drawn by the social component toward $\hat{\mathbf{y}}$. When a particle reaches $\hat{\mathbf{y}}$ then $\mathbf{x}_i = \hat{\mathbf{y}}$ and the social component no longer plays any significant role either. Once the cognition and social components can be ignored, it is then clear that a given particle eventually stops moving. Since $w < 1$, the velocity is reduced by a non-zero amount during each time step.

Note that the best particle moves based only on the inertia term since $\mathbf{x}_i = \mathbf{y}_i = \hat{\mathbf{y}}$ at the time step when it became the best. After this point, its position may improve, in which case $\mathbf{x}_i = \mathbf{y}_i = \hat{\mathbf{y}}$ holds again. Alternatively, its position will worsen, in which case it will be drawn back to $\mathbf{y} = \hat{\mathbf{y}}$ by the social component. This makes it likely for the best particle to stand still, particularly for values of $w \ll 1$.

Informally, when all the particles collapse with zero velocity on a given position in the search space, then the swarm has converged. If all the particles manage to catch up to the best particle before the optimum location is reached then the only component keeping the swarm moving is the inertia term. In this case, it is possible for the inertia weight to drive all velocities to zero before the swarm manages to reach a local extremum. Worse, the inertia is not even necessarily in a direction that leads to an extremum. Thus, it is possible for the standard PSO to converge prematurely without finding a local extremum.

For a more formal discussion and mathematical proofs about the convergence behaviour of the standard PSO see [7], [8]. The following relation should hold in order for the PSO to converge [7]:

$$\frac{c_1 + c_2}{2} - 1 < w \tag{3}$$

III. The GCPSO

The GCPSO was introduced by van den Bergh [6], [7] to address the issue of premature convergence to solutions that are not guaranteed to be local extrema. The modifications to the standard PSO involve replacing the velocity update equation (see Equation (2)) of only the best particle with the following (based on [9]):

$$v_{i,j}(t+1) = wv_{i,j}(t) - x_{i,j}(t) + y_{i,j} + \rho(t) r_j \tag{4}$$

where r_j is a sequence of uniform random numbers sampled from $U(-1, 1)$ and $\rho(t)$ is a scaling factor determined using:

$$\begin{aligned} \rho(0) &= 1.0 \\ \rho(t+1) &= \begin{cases} 2\rho(t) & \text{if } \#\text{successes} > \mathrm{s_c} \\ 0.5\rho(t) & \text{if } \#\text{failures} > \mathrm{f_c} \\ \rho(t) & \text{otherwise} \end{cases} \end{aligned} \tag{5}$$

where s_c and f_c are tunable threshold parameters. Whenever the best particle improves its personal best position, the success count is incremented and the failure count is set to 0 and *vise versa*. The success and failure counters are both set to 0 whenever the best particle changes.

These modifications cause the best particle to perform a directed random search in a non-zero volume around its best position in the search space. Recall that, in the standard PSO, the best particle can only move towards a better solution due to the inertia term (which may be 0). Therefore, the directed random search can dramatically speed up movement towards an extremum, particularly on unimodal functions. In addition, the GCPSO is guaranteed to at least converge on a local extremum (see [7] for a proof).

On multimodal functions, the GCPSO has the potential to find poorer solutions than the standard PSO. This is a consequence of faster convergence of the best particle towards a local extremum.

Consider a scenario where the best particle has rapidly found a local extremum due to the directed random search. This extremum would be a much better solution at any given point in time than it would have been in the case of the standard PSO. As a result, other particles approaching this position are not likely to be candidates for the best position. These particles miss opportunities to draw the swarm to their areas of the search space. The entire swarm may be drawn to a local solution earlier, limiting exploration.

IV. Neighbourhoods

Neighbourhoods model the structure of social networks [4]. Implementing neighbourhoods in the standard PSO requires replacing the velocity update (see Equation (2)) with the following:

$$\begin{aligned} v_{i,j}(t+1) = \; & wv_{i,j}(t) + \\ & c_1 r_{1,j}(y_{i,j} - x_{i,j}(t)) + \\ & c_2 r_{2,j}(\hat{y}_{i,j} - x_{i,j}(t)) \end{aligned} \tag{6}$$

where $\hat{\mathbf{y}}_i$ is the best position found so far in the neighbourhood of the i^{th} particle as opposed to the best position found by the entire swarm.

The original velocity update equation models a fully connected network structure since every particle is attracted to the best solution found by the entire swarm. This topology is known as *gbest* [10]. Other topologies differ only by which particles are considered to be in the neighbourhood of the i^{th} particle and as a consequence which particles play a role in the social component of Equation (6).

The *lbest* [10] topology includes only the k nearest neighbours, in terms of particle indices, of the i^{th} particle. This forms a ring like topology and as such is also known as the circle topology.

More recently, Kennedy [5] identified other useful neighbourhood topologies including the Von Neumann topology. The Von Neumann topology connects the particles in a grid network structure where each particle is directly connected to four neighbours: above, below, to the left and to the right.

The primary purpose of neighbourhoods is to preserve diversity within the swarm by impeding the flow of information through the network. Neighbourhood topologies such as *lbest* with $k = 2$ provide a very slow flow of information since any two randomly chosen particles communicate social information via a long indirect path. This allows particles to explore larger areas of the search space without being immediately drawn to the global best particle.

The structural layout of network connections may also have an effect on performance, particularly for multimodal functions [5].

Modifying the GCPSO to cater for neighbourhoods presents difficulties in dealing with $\rho(t)$ since there is now the possibility that more than one best particle exists. In addition, these neighbourhood best positions may or may not be shared between multiple neighbourhoods at any given point in time. Thus, the number of best particles is constantly changing. To solve this, a value for $\rho(t)$ is stored for each particle. Care must be taken to copy the previous value of $\rho(t)$ to a new best particle whenever the neighbourhood best particle changes (since a particle that is no longer the best in a given neighbourhood may still be the best in other neighbourhoods).

V. Benchmark Functions

The benchmark functions in this section provide a balance of unimodal and multimodal as well as easy and difficult functions. Subsets of these functions have been used in various particle swarm studies [4], [5], [7].

For each of these functions, the goal is to find the global minimiser. Stated formally:

Given $\mathrm{f} : \mathbb{R}^n \to \mathbb{R}$
find $\mathbf{x}^* \in \mathbb{R}^n$ for which $\mathrm{f}(\mathbf{x}^*) \leq \mathrm{f}(\mathbf{x}),\ \forall \mathbf{x} \in \mathbb{R}^n$

Spherical: $\mathbf{x}^* = \mathbf{0}$, with $f(\mathbf{x}^*) = 0$.

$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2 \tag{7}$$

Quadric: $\mathbf{x}^* = \mathbf{0}$, with $f(\mathbf{x}^*) = 0$.

$$f(\mathbf{x}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2 \tag{8}$$

Rosenbrock: $\mathbf{x}^* = (1, 1, \ldots, 1)$, with $f(\mathbf{x}^*) = 0$.

$$f(\mathbf{x}) = \sum_{i=1}^{n/2} \left(100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2 \right) \tag{9}$$

Ackley: $\mathbf{x}^* = \mathbf{0}$, with $f(\mathbf{x}^*) = 0$.

$$\begin{aligned} f(\mathbf{x}) \quad = \quad & -20 \exp\left(-0.2 \sqrt{\tfrac{1}{n} \textstyle\sum_{i=1}^{n} x_i^2} \right) \\ & - \exp\left(\tfrac{1}{n} \textstyle\sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + e \end{aligned} \tag{10}$$

Griewank: $\mathbf{x}^* = \mathbf{0}$, with $f(\mathbf{x}^*) = 0$.

$$f(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1 \tag{11}$$

Rastrigin: $\mathbf{x}^* = \mathbf{0}$, with $f(\mathbf{x}^*) = 0$.

$$f(\mathbf{x}) = \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i) + 10 \right) \tag{12}$$

Schwefel: $f(\mathbf{x}^*) = \mathbf{0}$, when
$\mathbf{x}^* = (-420.9687, -420.9687, \ldots, -420.9687)$.

$$f(\mathbf{x}) = 418.9829n + \sum_{i=1}^{n} x_i \sin\left(\sqrt{|x_i|} \right) \tag{13}$$

Spherical, Quadric and Rosenbrock are unimodal while Ackley, Griewank, Rastrigin and Schwefel are multimodal functions. The functions have been presented roughly in order of difficulty for the PSO to find the global minimiser (see Section VII).

VI. Experiment Parameters

The PSO parameters that were used are listed in Table I. The acceleration constants and inertia weight satisfy Equation (3) and are expected to provide convergent behaviour. The GCPSO specific parameters were empirically determined. The *lbest* parameter was chosen to provide a high impedance to

TABLE I
PSO PARAMETERS

Parameter	Value
#particles	20
$c_1 = c_2$	1.49618
w	0.729844
$s_c = f_c$	5
k (*lbest*)	2

communication within the swarm, since it was hypothesised that this would aid in finding the best minimiser for the multimodal functions. The benchmark function parameters that were used are listed in Table II, where d is the domain around the origin in which the PSO was initialised.

TABLE II
FUNCTION PARAMETERS

Function	n	d
Spherical	30	100
Quadric	30	100
Rosenbrock	30	2.048
Ackley	30	30
Griewank	30	600
Rastrigin	30	5.12
Schwefel	30	500

The cross product of the *gbest*, *lbest* and Von Neumann neighbourhoods with the standard PSO and GCPSO were tested on each function resulting in six experiments per function. For each experiment the simulation was run 100 times and the mean, standard deviation, median and range of error values over all the runs were recorded. Each simulation was allowed to run for 200 000 evaluations of the objective function.

VII. RESULTS

A table of results is provided for each of the benchmark functions in Section V. Each table consists of rows of experiments. The first column lists the names of the experiments where the prefixes G and P refer to the type of swarm used: GCPSO and standard PSO respectively and the subscripts g, l and v refer to the neighbourhood topology used: *gbest*, *lbest* and Von Neumann respectively. The second column lists the sample mean and standard deviation of the error. Since all the benchmark functions have $f(\mathbf{x}^*) = 0$, the error is defined to be equal to the fitness value obtained. The notation $\overline{x} \pm s$ where $\overline{x}$ is the mean and s is the standard deviation is used. The third column lists the median error values and the last gives the range of minimum and maximum error values.

For each experiment, the median error was plotted against the number of evaluations of the objective function performed. All the plots for a given benchmark function have been provided on the same system of axes. To improve the resolution, the $\log_{10}$ values for the error have been plotted instead of the actual values. These plots can be used to compare the convergence behaviour of the various PSO configurations given a particular function. The plots have a self explanatory key.

Under the assumption of normally distributed samples, statistical hypothesis testing can be used to reason about the population mean. The assumption of a normal distribution may be relaxed in the case of large samples (typically 100 samples or more). Whenever a statistically significant difference in the population mean is mentioned, it is assumed that the simulation data is normally distributed. Be aware that this is not necessarily guaranteed, even given the relatively large samples used here. All statistical tests were performed at a 1% level of significance using the standard t-test.

TABLE III
ERROR VALUES FOR SPHERICAL

	$\overline{x} \pm s$	Median	R
G_g	3e-161 ± 1e-160	4e-164	[8e-169:1e-159]
G_l	3e-93 ± 2e-92	4e-102	[2e-105:1e-91]
G_v	6e-119 ± 3e-118	3e-122	[1e-126:2e-117]
P_g	3e-96 ± 2e-95	7e-109	[4e-124:2e-94]
P_l	2e-91 ± 7e-91	1e-92	[1e-95:5e-90]
P_v	4e-112 ± 3e-111	6e-116	[5e-121:2e-110]

The results for the Spherical function are provided in Table III. Generally, for unimodal functions, the more connected the neighbourhood structure the better the results. The GCPSO did not benefit from neighbourhoods. There is little point in attempting to preserve swarm diversity for unimodal functions, except when using the standard PSO (from now, referred to as the PSO) since neighbourhoods may help prevent premature convergence. This explains why the P_v outperformed the P_g configuration. The Von Neumann architecture may slow down the rate of communication between the particles just enough to prevent all the particles collapsing on the same region in search space too soon. The social component of Equation (2) has more impact in diverse swarms. However, the P_v configuration did not perform significantly better than the P_g configuration. All of the configurations perform significantly better than the P_l configuration.

The *lbest* configurations were relatively unstable

having higher standard deviations. The high deviation for the P_g configuration may be explained by premature convergence. The G_g configuration was the most stable having the lowest deviation. Note that the mean and deviation of the two *lbest* configurations are comparable while the median for the GCPSO version is much better.

The directed random search performed by the GCPSO provided an advantage both in terms of solution quality and stability of the algorithm. The GCPSO always outperformed its PSO counterparts.

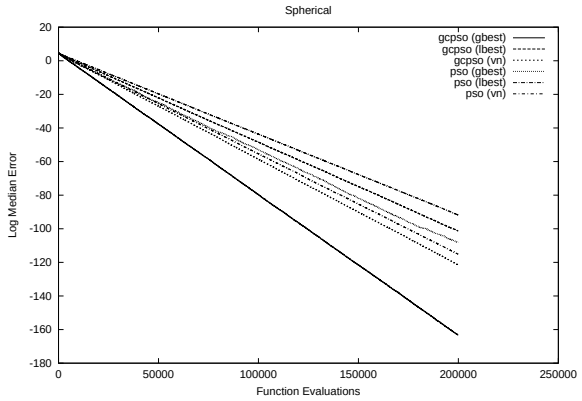

Fig. 1. Log Median Error for Spherical

Consider the plot in Fig. 1. Here the more rapid convergence of the GCPSO algorithm was evident, particularly in the case of the *gbest* topology. The GCPSO showed less of an advantage over the PSO for the neighbourhood configurations.

The directed random search performed by the GCPSO is sensitive to the value of $\rho(t)$ in Equation (4). Recall that the success and failure counts are reset each time the best position switches from one particle to the next. In addition, if the switch takes place between particles in different areas of the search space then the value of $\rho(t)$ may not be suited to different function landscapes. The GCPSO algorithm incurs a penalty each time the best particle changes since it takes time for $\rho(t)$ to re-adjust and stabilise. It is possible that the neighbourhood configurations incurred more switches than the *gbest* topology resulting in less optimal values for $\rho(t)$.

Table IV demonstrates trends in Quadric that are similar to Spherical (both are unimodal). Once again the GCPSO configurations consistently outperformed their PSO counterparts. Again, the more connected neighbourhood topologies exhibited superior performance. The P_g configuration once again performed more poorly than G_g.

The mean error for the G_g configuration was somewhat higher than the same configuration on Spherical, suggesting that Quadric is a slightly harder function to optimise. However, the neighbourhood configurations provided performance more consistent with that of Spherical.

TABLE IV
ERROR VALUES FOR QUADRIC

	$\overline{x} \pm s$	Median	R
G_g	7e-139 ± 5e-138	3e-143	[1e-149:5e:137]
G_l	7e-90 ± 3e-89	6e-92	[1e-94:2e-88]
G_v	2e-110 ± 1e-109	8e-114	[2e-116:8e-109]
P_g	2e-90 ± 2e-89	3e-108	[1e-123:2e-88]
P_l	2e-89 ± 6e-89	2e-90	[5e-93:3e-88]
P_v	5e-111 ± 3e-110	9e-114	[1e-117:2e-109]

The difference between the two Von Neumann configurations is negligible. To lesser degree, the difference between the *lbest* configurations is also very small. This suggests that the GCPSO benefited much less from neighbourhoods than the PSO on Quadric.

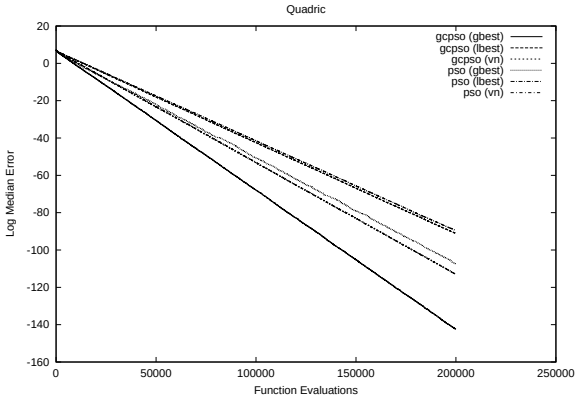

Fig. 2. Log Median Error for Quadric

Fig. 2 shows the plot for Quadric. This plot is virtually identical to Fig. 1 except for the smaller differences between the neighbourhood architectures mentioned before.

TABLE V
ERROR VALUES FOR ROSENBROCK

	$\overline{x} \pm s$	Median	Range
G_g	0.0129 ± 0.0246	0.0052	[0.0003:0.1679]
G_l	0.654 ± 0.53	0.5394	[0.168:4.3173]
G_v	0.1801 ± 0.097	0.1629	[0.0291:0.552]
P_g	0.058 ± 0.43	0.0057	[0.0004:4.3011]
P_l	0.4588 ± 0.305	0.4212	[0.1613:2.9975]
P_v	0.1617 ± 0.0936	0.1365	[0.0341:0.515]

Given Table V, it can be concluded that Rosenbrock is a significantly harder function than Spherical or Quadric. Rosenbrock breaks the trend of the GCPSO always performing better than the PSO on unimodal functions. Note that, if the s_c and s_f parameters are carefully tuned for Rosenbrock, then it is possible to get the G_g configuration to outperform the P_g configuration slightly [7]. This indicates that the sensitivity of the GCPSO to values of $\rho(t)$ is dependent on the function landscape.

For Rosenbrock the directed random search of the GCPSO did not appear to provide any advantage over the PSO. In fact, the P_l configuration was significantly better than G_l. However, the trend of more connected neighbourhoods performing better on unimodal functions still holds for both GCPSO and PSO configurations. The *gbest* configurations performed significantly better than the Von Neumann configurations and these were in turn significantly better than the *lbest* configurations.

It has already been noted that the GCPSO appears not to benefit from neighbourhoods to the same extent that the PSO does. However, the GCPSO also does not appear to be any more adversely affected by neighbourhoods than the PSO on unimodal functions.

Once again the two *lbest* configurations as well as the P_g configuration demonstrated instability as evidenced by their high standard deviations. The P_l appeared to be more stable than G_l. Once again the G_g configuration was the most stable indicating that even on Rosenbrock the directed random search is not without some advantage.

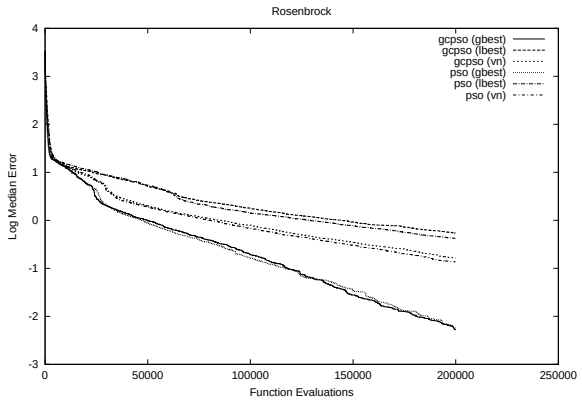

Fig. 3. Log Median Error for Rosenbrock

The plot in Fig. 3 illustrates an interesting convergence pattern for Rosenbrock. All six configurations demonstrated a similar pattern. There was a rapid decrease in error early in the search. After this, the improvement flattened out for a short while before making another smaller dip. Finally, the error value continued to decrease at a much slower pace. Since all six configurations display this characteristic, it is not accounted for by the neighbourhood topology or the type of swarm. What is noticeable is that the *gbest* configurations maintained a much steeper descent in error value after the other configurations had flattened out.

TABLE VI
Error values for Ackley

	$\overline{x} \pm s$	Median	Range
G_g	2.0181 ± 1.3102	1.9565	[1e-14:9.3975]
G_l	0.2794 ± 0.5424	7e-15	[7e-15:2.221]
G_v	0.6824 ± 0.8269	7e-15	[4e-15:2.5799]
P_g	3.6708 ± 1.5625	3.3444	[3e-14:8.9018]
P_l	0.0667 ± 0.2676	7e-15	[7e-15:1.3404]
P_v	0.7098 ± 0.845	1e-14	[4e-15:2.887]

Consider Table VI which contains the results for Ackley. As soon as the function was multimodal, the trend regarding neighbourhoods reversed. Now, less connected networks performed better. This was motivated in Section IV.

The G_g configuration was significantly better than the P_g configuration. This may again be accounted for by the premature convergence of the PSO. Therefore, the directed random search may still be useful in a multimodal context but clearly to a lesser degree. The GCPSO seemed to break down with the *lbest* topology. The P_l configuration was significantly better than the G_l configuration. The GCPSO managed to hold its ground with the Von Neumann topology but was not significantly better than the PSO.

The stability of the *gbest* topologies were low. Some *gbest* simulations became trapped in local minima. The neighbourhood topologies reduce the risk of this happening by maintaining swarm diversity for longer periods of time.

Fig. 4 shows the two *gbest* configurations getting trapped in a local minimum very early during the search. Only the neighbourhood configurations managed to maintain any momentum. The Von Neumann architectures converged more rapidly than their *lbest* counterparts. However, both *lbest* configurations managed to achieve significantly better solutions than either of the Von Neumann configurations. This indicates that faster convergence may not be desirable in multimodal landscapes.

The results in Table VII for Griewank display similar characteristics to Ackley. Again, it can be seen that for multimodal functions the less connected neighbourhoods performed better. The P_g configuration got trapped again. The G_g configuration was

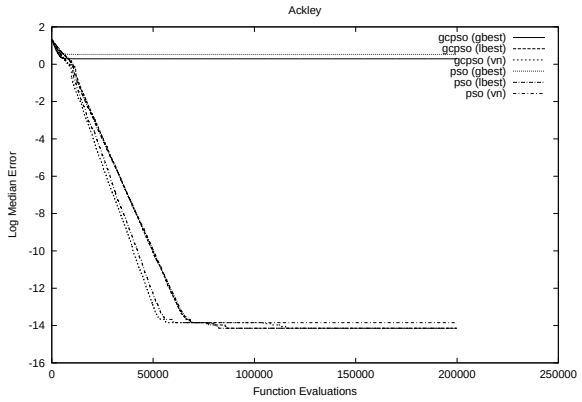

Fig. 4. Log Median Error for Ackley

TABLE VII
ERROR VALUES FOR GRIEWANK

	$\overline{x} \pm s$	Median	Range
G_g	0.0162 ± 0.0219	0.0074	[1e-19:0.1077]
G_l	0.0039 ± 0.0075	1e-19	[1e-19:0.0393]
G_v	0.0104 ± 0.0145	0.0074	[0:0.08549]
P_g	0.1353 ± 0.3154	0.0405	[2e-19:2.1642]
P_l	0.0051 ± 0.0086	1e-19	[0:0.0394]
P_v	0.0134 ± 0.019	0.0074	[0:0.0903]

once again significantly better than the P_g configuration.

Although both Von Neumann configurations are significantly better than the P_g configuration they failed to keep up with the *lbest* configurations. In fact, both *lbest* architectures are significantly better than all the other topologies but neither of them is significantly better than the other. Nevertheless, G_l has a slightly better average error and stability.

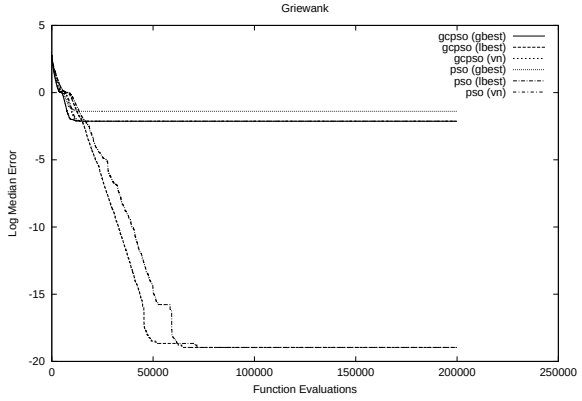

Fig. 5. Log Median Error for Griewank

Fig. 5 shows that even the Von Neumann architectures got trapped early during the search. Only the two *lbest* configurations were able to maintain diversity long enough not to get trapped in a local minimum. Notice that the G_l configuration displayed a much more rapid convergence curve than the P_l configuration. This is due to the directed random search.

TABLE VIII
ERROR VALUES FOR RASTRIGIN

	$\overline{x} \pm s$	Median	Range
G_g	71.925 ± 18.692	71.637	[31.839:139.29]
G_l	61.202 ± 15.415	60.692	[27.859:93.526]
G_v	55.837 ± 14.501	54.723	[29.849:99.496]
P_g	72.204 ± 18.678	68.652	[29.849:136.31]
P_l	64.567 ± 14.941	63.677	[31.839:112:43]
P_v	54.355 ± 15.353	51.738	[22.884:98.501]

The results for Rastrigin in Table VIII are interesting because the ranking of the neighbourhood architectures is different. For Rastrigin, the Von Neumann topologies were significantly better than the *lbest* topologies and they were in turn significantly better than the *gbest* topologies. This suggests that the structural layout of the neighbourhood may be significant. There was no significant difference between the GCPSO and PSO configurations. In fact, both algorithms have almost identical performance in terms of mean error as well as stability for a given neighbourhood topology. The only noticeable difference was the slightly higher worst case error for P_l which could have been caused by a single simulation.

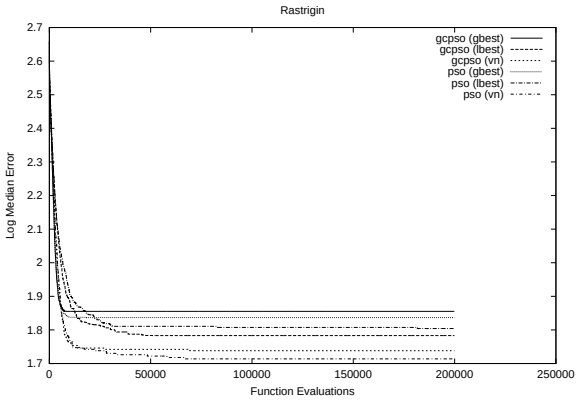

Fig. 6. Log Median Error for Rastrigin

Fig 6 shows very similar convergence characteristics between the GCPSO and PSO configurations. The medians for the PSO configurations are slightly better than the GCPSO, however, the averages for the *gbest* and *lbest* configurations show the reverse to

be true.

TABLE IX
ERROR VALUES FOR SCHWEFEL

	$\overline{x} \pm s$	Median	Range
G_g	4539 ± 706.06	4550.2	[2882.2:6534.1]
G_l	4762.1 ± 509.36	4797.8	[3395.4:5863.1]
G_v	4496.9 ± 707.83	4451.5	[2862.3:6356.5]
P_g	4535.6 ± 722.33	4510.8	[2803.1:6179.5]
P_l	4634 ± 642.22	4609.5	[2329.4:6219.5]
P_v	4273.4 ± 565.86	4333.1	[2664.9:5449.1]

The results for Schwefel are presented in Table IX for completeness. Schwefel has an extremely difficult landscape with many local minima. In addition, the global minimum is located in one corner of the search space, making it even more difficult for the PSO to locate it.

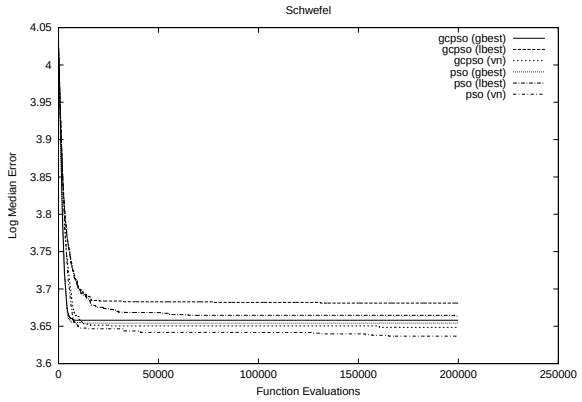

Fig. 7. Log Median Error for Schwefel

Fig. 7 shows that the *lbest* topologies performed worse than the *gbest* topologies. In fact, both *gbest* topologies performed significantly better than the G_l configuration. This behaviour is not easily explained and should be investigated further.

VIII. CONCLUSION

The very nature of neighbourhoods impedes information flow. This is not a desirable property for unimodal optimisation as it slows convergence when there is no need to preserve diversity. The premature convergence of the standard PSO before reaching a local extremum is better solved by using the guaranteed convergence modifications. Stability of the *lbest* topologies was also poor for unimodal functions.

On the other hand, for multimodal optimisation problems, neighbourhoods are extremely desirable. In this case, slowing down convergence preserves diversity in the swarm. This results in a larger area of the search space being explored. It makes intuitive sense that sparsely connected network structures should display better performance on multimodal functions. This was found to be true for some functions with Rastrigin and Schwefel being exceptions to the rule.

The benefits of the directed random search in multimodal functions is greatly reduced but not eliminated. Although the GCPSO does not seem to benefit as much from neighbourhoods as the standard PSO does, it does not seem to be adversely affected by them either.

REFERENCES

[1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, vol. IV, (Perth, Australia), pp. 1942–1948, 1995.

[2] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, (Nagoya, Japan), pp. 39–43, 1995.

[3] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE Congress on Evolutionary Computation*, (Anchorage, Alaska), pp. 69–73, May 1998.

[4] J. Kennedy, "Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance," in *Proceedings of IEEE Congress on Evolutionary Computation*, (Washington D.C, USA), pp. 1931–1938, July 1999.

[5] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proceedings of the IEEE Congress on Evolutionary Computation*, (Honolulu, Hawaii USA), May 2002.

[6] F. van den Bergh and A. Engelbrecht, "A new locally convergent particle swarm optimizer," in *Proceedings of IEEE Conference on Systems, Man and Cybernetics*, (Hammamet, Tunisia), Oct. 2002.

[7] F. van den Bergh, *An Analysis of Particle Swarm Optimizers*. PhD thesis, Department of Computer Science, University of Pretoria, South Africa, 2002.

[8] M. Clerc and J. Kennedy, "The Particle Swarm: Explosion, Stability and Convergence in a Multi-Dimensional Complex Space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[9] F. Solis and R. Wets, "Minimization by random search techniques," *Mathematics of Operations Research*, vol. 6, pp. 19–30, 1981.

[10] R. C. Eberhart, P. Simpson, and R. Dobbins, *Computational Intelligence PC Tools*, chapter 6, pp. 212–226. Academic Press Professional, 1996.