Particle Swarm Optimization: Global Best or Local Best?

AP Engelbrecht
Department of Computer Science
University of Pretoria
Pretoria, South Africa
Email: engel@cs.up.ac.za

***Abstract*—A number of empirical studies have compared the two extreme neighborhood topologies used in particle swarm optimization (PSO) algorithms, namely the star and the ring topologies. Based on these empirical studies, and also based on intuitive understanding of these neighborhood topologies, there is a faction within the PSO research community that advocates the use of the local best (lbest) PSO due to its better exploration abilities, diminished susceptibility to being trapped in local minima, and because it does not suffer from premature convergence as is the case with the global best (gbest) PSO. However, the opinions that emanated from these studies were based on a very limited benchmark suite containing only a few benchmark functions. This paper conducts a very elaborate empirical comparison of the gbest and lbest PSO algorithms on a benchmark suite of 60 boundary constrained minimization problems of varying complexities. The statistical analysis conducted shows that the general statements made about premature convergence, exploration ability, and even solution accuracy are not correct, and shows that neither of the two algorithms can be considered outright as the best, not even for specific problem classes.**

I. Introduction

The first two versions of the particle swarm optimizer (PSO) [1] differed only in the neighborhood topology used: the global best (gbest) PSO used the star topology where all particles are attracted towards the best position found by the swarm, while the local best (lbest) PSO used the ring topology where each particle is attracted towards the best position found by that particle's neighborhood [2]. Much has been said about the advantages and disadvantages of these topologies, and others [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17]. The general message from these publications is against the use of gbest PSO in general, and if used, gbest PSO should be limited to unimodal problems. This opinion is strongly expressed by Kennedy on his blog, Great Swarm Speaks, as quoted in [4]:

> "...and it might be time to mount a sword-swinging crusade against any use of the gbest topology. How did this happen? It is the worst way to do it, completely unnecessary. I will not tolerate anybody who uses that topology complaining about 'premature convergence'."

Clerc [4] stated that the above sentiment is valid for single-objective PSO algorithms, but that the gbest PSO is sometimes still acceptable for multi-objective optimization. Clerc also emphasized the fact that gbest is really only suited for unimodal problems. Note that the published empirical results thus far, upon which these sentiments were based, considered benchmark suites of very limited size.

This paper accepts the invitation to a sword-swinging crusade, by providing statistically sound evidence that these damning statements about gbest PSO are unfounded, based on an extensive empirical analysis of gbest PSO and lbest PSO on a set of 60 boundary constrained optimization problems of varying difficulty. The analysis is conducted by comparing performance with respect to solution accuracy, success rate, efficiency, and consistency. At the same time, the diversity profiles of the two algorithms are also compared. The objective of this paper is by no means to provide evidence against the use of lbest PSO, or any other neighborhood topology for that matter, but rather to show that the choice of which topology to use (here considering only the two rivals, the star and the ring topology) is very function dependent. To strongly emphasize this point, the choice is not even function class (i.e. unimodal or multimodal, or separable or non-separable) dependent, but function dependent. The empirical analysis of this paper provides convincing support in favor of gbest PSO even for multimodal and non-separable problems. As an outcome of this paper, it is hoped that any such general statements against gbest PSO no longer be made.

The rest of this paper is organized as follows: Section II provides a summary of the PSO as used in this study, followed by a review of studies about neighborhood topologies and statements made about these topologies in Section III. The empirical procedure is discussed in Section IV, and the results presented and discussed. A list of the benchmark functions used is given in Appendix A.

II. Basic Particle Swarm Optimizer

This section summarizes the very basic PSO as used in this study, assuming the inertia weight model as developed by Shi and Eberhart [18]. Each particle, i, has its position, $\mathbf{x}_i$, updated using

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

978-1-4799-3194-1/13 $31.00 © 2013 IEEE
DOI 10.1109/BRICS-CCI-CBIC.2013.31

with $\mathbf{x}_i(0) \sim U(\mathbf{x}_{min}, \mathbf{x}_{max})$. The step size, or velocity, $\mathbf{v}_i$ is computed using

$$\begin{aligned} v_{ij}(t+1) &= wv_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] \\ &+ c_2 r_{2j}(t)[\hat{y}_{ij}(t) - x_{ij}(t)] \end{aligned} \quad (2)$$

where $v_{ij}(t)$ is the velocity of particle i in dimension $j = 1, \ldots, n_x$ at time step t, $x_{ij}(t)$ is the position of particle i in dimension j at time step t, c_1 and c_2 are positive acceleration constants, $r_{1j}(t), r_{2j}(t) \sim U(0,1)$ are random values in the range $[0,1]$, sampled from a uniform distribution, and n_x is the dimension of the search space.

The personal best position, $\mathbf{y}_i$, is computed as

$$\mathbf{y}_i(t+1) = \begin{cases} \mathbf{y}_i(t) & \text{if } f(\mathbf{x}_i(t+1)) \geq f(\mathbf{y}_i(t)) \\ & \text{and } \mathbf{x}_i(t+1) \in [\mathbf{x}_{min}, \mathbf{x}_{max}]^{n_x} \\ \mathbf{x}_i(t+1) & \text{if } f(\mathbf{x}_i(t+1)) < f(\mathbf{y}_i(t)) \end{cases} \quad (3)$$

where $f : \mathbb{R}^{n_x} \to \mathbb{R}$ is the objective function.

The neighborhood best position, $\hat{\mathbf{y}}_i$, is computed as the best solution found within the neighborhood, $\mathcal{N}_i$, of particle i; that is,

$$\hat{\mathbf{y}}_i(t+1) \in \{\mathcal{N}_i | f(\hat{\mathbf{y}}_i(t+1)) = \min\{f(\mathbf{x})\}, \quad \forall \mathbf{x} \in \mathcal{N}_i\} \quad (4)$$

with the neighborhood defined as

$$\begin{aligned} \mathcal{N}_i &= \{\mathbf{y}_{i-n_{\mathcal{N}_i}}(t), \mathbf{y}_{i-n_{\mathcal{N}_i}+1}(t), \ldots, \\ &\quad \mathbf{y}_{i-1}(t), \mathbf{y}_i(t), \mathbf{y}_{i+1}(t), \ldots, \mathbf{y}_{i+n_{\mathcal{N}_i}}(t)\} \end{aligned} \quad (5)$$

for neighborhoods of size $n_{\mathcal{N}_i}$.

The initial velocities, $\mathbf{v}_i(0)$, are preferably initialized to zero vectors [19].

What distinguishes the global best (gbest) PSO from the local best (lbest) PSO is the neighborhood topology used, and thus how the neighborhood best position is determined:

- The gbest PSO uses a star neighborhood topology, where each particle has the entire swarm as its neighborhood. Therefore, $\hat{\mathbf{y}}_i = \hat{\mathbf{y}}$ for all particles $i = 1, \ldots, n_s$, where n_s is the size of the swarm. The consequence is that all particles are attracted to one global best position.
- The lbest PSO uses a ring topology, where each particle's neighborhood consists of itself and its immediate two neighbours. Note that neighborhoods overlap. The consequence is that each particle is attracted to a (usually) different neighborhood best position.

III. Gbest PSO Versus Lbest PSO

Since the very first publications on PSO, up to very recently, much has been published about the two extreme neighborhood topologies used to transfer information about best positions throughout the swarm. The general sentiment portrayed from these publications is that gbest PSO should not be used, and that a sparsely connected topology should be preferred. This section summarizes some of these opinions:

- Gbest PSO should not be used due to premature convergence to local optima as observed for a number of optimization problems [20], [10], [13], [16], [21]. While it is the case that gbest PSO may prematurely converge, it is important to note that convergence is not necessarily to a local optimum as pointed out in [22], and formally proved in [23]. A reason for such premature convergence to a stable state [24], [25], [26] is due to the fact that the position of a particle can be the same as both its personal best and global best positions [22]. If this happens for a number of iterations, the inertia component is driven to zero and so are the velocities, resulting in no changes in particle positions. Knowing this, and addressing the problem, similar to what is proposed in [22] can effectively address the problem of premature convergence to an arbitrary point. The same can occur in lbest PSO where a particle's position can be the same as its personal best and neighborhood best positions, resulting in the same stagnating behavior. Such stagnation behavior is also hinted at in [27]. Section IV-E provides some empirical evidence to support the statement that premature convergence also occurs in lbest PSO. The issue that there is no guaranteed convergence to a local optimum can be addressed by making use of some form of mutation [23]. Peer *et al* [16] also noted that the premature convergence problem is best addressed considering changes to the velocity update equation instead of changes in the neighborhood topology.
- Gbest PSO converges fast due to the faster transfer of the best position throughout the swarm, and therefore the strong attraction to one best position [3], [5], [6], [20], [10], [11], [13], [14], [15]. Inversely, lbest PSO converges more slowly, and therefore explores more as it maintains diversity for longer. Intuitively, this makes sense. However, as is pointed out in Section IV-E, gbest PSO converged faster than lbest PSO for less than half of the benchmark functions. Furthermore, evidence is given in Section IV-E that lbest PSO converges faster than gbest PSO for a number of the functions used, and actually reduced diversity faster than gbest PSO for some functions.
- Gbest PSO is more susceptible to being trapped in local minima than lbest PSO [5], [6], [11]. This statement should be considered together with the formal proofs that there is no guarantee that the original forms of PSO will even find a local minimum [23], and the reason for premature stagnation as described above. Results presented in Section IV-E show that gbest PSO provided more accurate solutions for just as many functions than did lbest PSO provide more accurate results. This fact alone indicates that both algorithms are equally good at finding good solutions and equally bad at finding worse solutions, depending on the function. It also indicates that lbest PSO can just as well become trapped in a local minimum. However, confirmation of this is needed by looking at the derivative of the objective function at the point to which particles converge.
- Gbest PSO is best suited to unimodal problems and should not be used for multimodal problems [3], [4], [7],

[10], [16]. This statement is simply not true as is pointed out in Section IV-E. While gbest PSO did provide better results on most of the unimodal problems, gbest PSO did also perform better than lbest PSO for a number of multimodal problems.
- Gbest PSO does not perform well for non-separable problems [11]. This statement is also not true. Section IV-E shows that gbest PSO performed better than lbest PSO for a number of non-separable problems. Based on this, and the finding with reference to function modality, it is simply not possible to suggest that any one of the two algorithms performs best for any of these broad classes of optimization functions.
- Increasing the size of neighborhoods deteriorates performance, and lbest PSO is superior to gbest PSO in terms of solution accuracy for the majority of problems [3], [20], [15]. This is simply not the case, as shown in Section IV-E.

The above lists only the main opinions about the two neighborhood topologies. The next section provides empirical support for the message of this paper, i.e. that gbest PSO is not as bad as it is being portrayed, and that it is not possible to identify a clear cut winner for any class of optimization functions.

IV. Empirical Analysis

The main purpose of this study was to provide an elaborate empirical comparison of the gbest and lbest PSO algorithms. A summary of the algorithms and their control parameters as used in this study are given in Section IV-A, and the benchmark functions used are described in Section IV-B. The performance measures are described in Section IV-C, followed by a description of the experimental procedure in Section IV-D. The results are then presented and discussed in Section IV-E.

A. Algorithms

The performance of the gbest PSO and that of the lbest PSO is compared in this study. For this purpose, the two algorithms were implemented[1] to differ only in the topology used. Both algorithms used synchronous updates, used 30 particles, and the control parameters were set as $w = 0.729844$ and $c_1 = c_2 = 1.49618$, shown to lead to convergent behavior [28]. Velocity clamping was not used. In order to ensure that best positions are feasible (i.e. do not leave the boundaries of the search space), personal best positions were updated only if the resulting position is feasible. Neighborhood (global) best positions were selected as the best personal best position in a partcle's neighborhood.

Each algorithm was executed for 5000 iterations on each of the benchmark functions, and 50 independent samples of each algorithm on each function were obtained.

[1]CIlib, available at http://www.cilib.net, was used to implement all algorithms.

B. Benchmark Functions

Each of the algorithms was applied to a benchmark suite of 60 problems, which includes unimodal, multi-modal, separable and non-separable, shifted, rotated, noisy, and composition functions. All of the problems were evaluated in 30 dimensions. The functions and their domains are given in Appendix A, and Table I summarizes the characteristics of these functions. A function, f_l, that is shifted, rotated, or shifted and rotated, is respectively referred to as f_l^{Sh}, f_l^{R} and f_l^{ShR}, where l is the index of the function.

A function, f_l, was shifted using

$$f_l^{Sh}(\mathbf{x}) = f_l(\mathbf{z}) + \beta$$

where $\mathbf{z} = \mathbf{x} - \gamma$; γ and β are constants. Table I lists the values by which each function was shifted.

Two approaches to rotation were implemented: Where an entry in the rotation column of Table I indicates "ortho", the function f_l was rotated by a randomly generated orthonormal rotation matrix. The second approach, indicated by "linear" rotates the function using a linear transformation matrix. For both approaches the condition number is given in the table in parentheses, and rotation was done using Salomon's method [29]. A new rotation matrix was computed for each of the 50 independent runs of the algorithm. The rotated function, referred to as f_l^R, was computed by multiplying the decision vector $\mathbf{x}$ with the transpose of the rotation matrix.

Some functions were both shifted and rotated, with the resulting function referred to as f_l^{ShR}. Noisy functions were generated by multiplying each decision variable, x_j, by zero-mean noise sampled from a Gaussian distribution with deviation of one. These functions are referred to as f_l^N. Functions that are shifted and noisy are referred to as f_l^{ShN}.

Note that f_{27} is indicated as a non-separable function, even though it is separable near the optimum.

Rotations for functions f_{26} to f_{37} and function f_{15}^{ShRE} were by linear transformation matrix, with condition numbers that differ for each component function. Also the severity of shifts differ for the different component functions. For the detail of these parameters, the reader is referred to [30]. For these functions, the entry "yes" simply indicates if a transformation of the expanded or composition function was done. An entry of "CEC05" refers the reader to the definition of the function as in [30].

In total, 60 different functions were used. However, note that the actual number of functions is significantly larger, since a new rotation matrix was calculated for each of the 50 independent runs for rotated functions, each resulting in a different function.

C. Performance Measures

Three performance measures were used to compare the two algorithms:
- **Accuracy**, computed as the best objective function value obtained at the end of the 5000 iterations.
- **Success rate**, computed as the percentage of the 50 independent runs that reached specified accuracy levels. For

TABLE I
CHARACTERISTICS OF BENCHMARK FUNCTIONS (F: FUNCTION NUMBER, EQ: EQUATION NUMBER, M: MODALITY, S: SEPARABILITY, SH: SHIFTED, R: ROTATED, SHR: SHIFTED AND ROTATED, N: NOISY, E: EXPANDED FUNCTION, C: COMPOSITION FUNCTION).

F	**Eq**	**M**	**S**	**Sh**	**R**	**ShR**	**N**	**E**	**C**
f_1	6	Uni	Yes						
f_2	7	Multi	No	$\beta = -140$ $\gamma = 10$	ortho (1)	$\beta = -140$ $\gamma = -32$ linear (100)			
f_3	8	Multi	Yes						
f_4	9	Multi	No						
f_5	10	Uni	Yes	$\beta = -450$ $\gamma = 10$	ortho (1)	$\beta = -450$ $\gamma = 10$ ortho (1)			
f_6	11	Multi	No	$\beta = -180$ $\gamma = 10$	ortho (1)	$\beta = -180$ $\gamma = -60$ linear (3)			
f_7	12	Uni	Yes						
f_8	13	Multi	Yes						
f_9	14	Multi	No						
f_{10}	15	Uni	No						
f_{11}	16	Uni	Yes				N(0,1)		
f_{12}	18	Multi	Yes	$\beta = -330$ $\gamma = 2$	ortho (1)	$\beta = -330$ $\gamma = 1$ linear (2)			
f_{13}	19	Multi	No	$\beta = 390$ $\gamma = 10$	ortho (1)				
f_{14}	20	Multi	No						
f_{15}	21	Multi	No			$\beta = -300$ $\gamma = 20$ linear (3)		Yes	
f_{16}	22	Uni	No	$\beta = -450$ $\gamma = 10$	ortho (1)		$\|N(0, 0.4)$ $\beta = -450$ $\gamma = 10$		
f_{17}	24	Uni	No	$\beta = -310$					
f_{18}	25	Multi	No	$\beta = -460$					
f_{19}	26	Uni	Yes						
f_{20}	27	Uni	Yes						
f_{21}	28	Multi	No						
f_{22}	29	Uni	Yes	$\beta = -450$ $\gamma = 10$					
f_{23}	30	Multi	Yes						
f_{24}	31	Multi	Yes						
f_{25}	32	Multi	Yes			$\beta = 90$ $\gamma = 0.1$ linear (5)			
f_{26}	CEC05	Multi	No	$\beta = -130$ $\gamma = 1$				Yes	
f_{27}	CEC05	Multi	No						Yes
f_{28}	CEC05	Multi	No		Yes				Yes
f_{29}	CEC05	Multi	No				Yes		Yes
f_{30}	CEC05	Multi	No		Yes				Yes
f_{31}	CEC05	Multi	No		Yes				Yes
f_{32}	CEC05	Multi	No		Yes				Yes
f_{33}	CEC05	Multi	No		Yes				Yes
f_{34}	CEC05	Multi	No		Yes				Yes
f_{35}	CEC05	Multi	No		Yes				Yes
f_{36}	CEC05	Multi	No		Yes				Yes
f_{37}	CEC05	Multi	No		Yes				Yes

the purpose of this study, a total of 1000 accuracy levels have been considered, with the accuracy levels starting at the best accuracy obtained by the two algorithms, logarithmically increasing towards the worst accuracy.

- **Efficiency**, computed as the average number of iterations over the 50 independent runs to reach different accuracy levels.
- **Consistency**, computed as the deviation from the average best function value obtained over the 50 independent runs.

In addition to these performance measures, the average diversity of the swarms were computed as the average distance from the center of the swarm [31].

D. Experimental Procedure

For each function, the Mann-Whitney U test was used to indicate difference in performance with respect to a specific performance measure at a significance level of 0.05. For each function class, the total number of wins for each algorithm was calculated as well as the number of functions for which there was no statistically significant difference in performance. These scores are reported in the tables to follow. With reference to the success rate, the samples to which the Mann-Whitney U test was applied consisted of the success rates for each of the accuracy levels. Therefore, the scores indicated for the success rate performance measure indicate how successful each algorithm was over the entire range of accuracy levels. With reference to the efficiency performance measure, the samples consisted of the average number of iterations over the 50 independent runs for each of the accuracy levels. A win based on the Mann-Whitney U test therefore indicates that the corresponding algorithm converged faster to most of the accuracy levels. The same process was followed with respect to the diversity measurement, where a win indicates that the corresponding algorithm had the highest diversity over the 5000 iterations.

E. Results

Table II summarizes the wins and losses obtained over all functions of the gbest PSO with respect to the lbest PSO, for the accuracy, success rate and efficiency measures. Results for the diversity measure are also given. When referring to the overall scores at the bottom of the table, it is clear that neither the gbest PSO, nor the lbest PSO can be claimed to be the best with respect to any of the performance measures. This immediately negates any published claim and statement that the lbest PSO is better than the gbest PSO.

With reference to the overall total scores, the following observations are noted:

- **Accuracy**: Gbest PSO obtained statistically significant better accuracies for 33.9% of the functions, while lbest PSO obtained better accuracies for only 37.3% of the functions. For the remaining 22.8% of the functions, there was no statistical significant difference in accuracy. Therefore, there is no statistical grounds upon which a general statement can be made that lbest PSO provides more accurate results than gbest PSO.
- **Success rate**: Gbest PSO showed better success rates over all the accuracy levels for 39.0% of the functions, while lbest PSO obtained success rates better than gbest PSO for only 22.0% of the functions. The two algorithms showed no difference in performance with respect to success rate for 39.0% of the functions. Notable is that gbest PSO achieved better success rates for nearly twice more of the functions that lbest PSO did. This raises questions about statements that gbest PSO should not be used due to the issue of premature stagnation. While premature stagnation is a problem with gbest PSO, the success rate results do show that this is no worse than lbest PSO, and hints towards the same problem of premature stagnation occuring in lbest PSO. The results in [27] also hints towards stagnation occuring in lbest PSO.
- **Efficiency**: For 44.1% of the problems gbest PSO did converge significantly faster to the accuracy levels than lbest PSO. For 27.1% of the simulations did lbest PSO converge faster to the different accuracy levels. While gbest PSO did converge faster to the given accuracy levels than lbest PSO, statements to the effect that gbest PSO always converge faster are therefore not true, as this was the case for less than half of the functions.
- **Diversity**: For none of the functions did the two algorithms show the same diversity profile. While the gbest PSO did reduce diversity faster than lbest PSO for 39% of the functions, lbest PSO maintained a lower diversity than gbest PSO over the 5000 iterations for 81.4% of the functions. In fact, for 59.3% of the functions did lbest PSO show a lower diversity than gbest PSO over all iterations. Statements that lbest PSO maintains diversity for longer are therefore simply not true based on the findings of this empirical study.

Drilling down to the unimodal and multimodal problem classes, it is seen that:

- **Accuracy**: Gbest PSO provided better results for more unimodal functions than did lbest PSO. For 61.1% of the unimodal functions did gbest PSO provide significantly better results, while lbest PSO performed better than gbest PSO for 16.7% of the functions. For the remaining 22.2% of the functions, there were no significant difference in performance. While gbest PSO did not perform better than lbest PSO for all the unimodal problems as is intuitively expected, gbest PSO did perform better for most of the functions. For the multimodal functions, lbest PSO provided the most accurate results for 46.3% of the functions, and gbest PSO for 22.0% of the functions. For the remaining 31.7% functions there was no significant difference in performance. It is therefore simply not the case that lbest PSO performs better than gbest PSO on multimodal functions, as this was shown to be the case for less than half of the problems. The findings from this study are also supported by empirical evidence from [3].
- **Success rate**: Gbest PSO obtained better success rates than lbest PSO for 72.2% of the unimodal functions, while lbest PSO performed best with respect to success rate for 27.8% functions. For the multimodal functions, performance was just about equal, with lbest PSO providing best success rates for 29.3% of the functions, and gbest PSO for 26.8% of the functions.
- **Efficiency**: Gbest PSO did not converge faster to the different accuracy levels for most of the problems. Only for 44.4% did gbest PSO converge faster, with lbest PSO converging faster for 27.8% of the problems. For the remainder of the problems, there was no significant difference in convergence speed. Convergence to difference accuracy levels for the multimodal functions was

TABLE II

OUTCOMES OF STATISTICAL ANALYSIS COMPARING GBEST WITH LBEST PSO; '>' INDICATES THE NUMBER OF FUNCTIONS FOR WHICH GBEST PSO WAS BETTER THAN LBEST PSO, '<' INDICATES THE NUMBER OF FUNCTIONS FOR WHICH GBEST PSO WAS WORSE THAN LBEST PSO, AND '=' INDICATES THE NUMBER OF FUNCTIONS FOR WHICH THERE WAS NO STATISTICALLY SIGNIFICANT DIFFERENCE IN PERFORMANCE.

Function Class		**Number of Functions**	**Accuracy**			**Success Rate**			**Efficiency**			**Diversity**		
			>	=	<	>	=	<	>	=	<	>	=	<
Unimodal	Seperable	7	5	0	2	6	0	1	2	0	5	5	0	2
	Non-separable	3	2	1	0	2	1	0	2	1	0	2	0	1
	Noisy	2	1	0	1	1	1	0	2	0	0	1	0	1
	Shifted	5	2	3	0	2	3	0	2	3	0	1	0	4
	Rotated	1	1	0	0	1	0	0	0	1	0	0	0	1
	Total	18	11	4	3	12	5	1	8	5	5	9	0	9
Multimodual	Seperable	6	1	2	3	2	2	2	3	1	2	6	0	0
	Non-seperable	9	4	1	4	3	4	2	4	3	2	1	0	8
	Shifted	10	3	4	3	5	5	0	8	1	1	1	0	9
	Rotated	4	0	3	1	1	2	1	2	1	1	0	0	4
	Noisy	1	0	1	0	0	1	0	0	1	0	0	0	1
	Composition	11	1	2	8	0	4	7	1	5	5	0	0	11
	Total	41	9	13	19	11	18	12	18	12	11	8	0	33
	Overall Total	59	20	17	22	23	23	13	26	17	16	11	0	48
	Overall Unimodal	18	11	4	3	12	5	1	8	5	5	9	0	9
	Overall Multimodal	41	13	19	14	11	18	12	18	12	11	2	0	39
	Overall Seperable	17	7	4	6	9	5	3	12	1	4	5	0	12
	Overall Non-seperable	42	13	13	16	14	18	10	11	16	9	6	0	36

  faster using the gbest PSO for 43.9% of the functions, compared to 26.8% for the lbest PSO. Note that these accuracy levels included more lower accuracies than higher accuracies.

Despite the fact that gbest PSO was shown to converge faster to the different accuracy levels for many functions, this was not at the cost of worst performance in terms of success rate nor accuracy.

With reference to separability, note the following:

- **Accuracy**: Gbest PSO and lbest PSO were just about equally good at minimizing the separable functions, with gbest PSO providing better results for 41.2% of the functions compared to the 35.3% of lbest PSO. The same is observed for the non-separable functions, where lbest PSO performed better for 38.1% of the functions and gbest PSO for 31.0%. It can therefore not be stated that lbest PSO is a better choice to optimize non-separable problems.
- **Success rate**: gbest PSO obtained better success rates for more separable and non-separable functions than lbest PSO. For the separable functions, gbest PSO was better for 52.9% of the functions and for the non-separable functions better for 33.3%, compared to the 17.6% and 23.8% of lbest PSO respectively for the two function classes.
- **Efficiency**: Gbest PSO again converged faster to the different accuracy levels for more functions for both function classes than lbest PSO. For the separable problems, gbest PSO was faster than lbest PSO for 70.6% of the functions compared to lbest PSO's 23.5%. For the non-separable functions, gbest PSO was faster for 26.2% functions and lbest PSO for 21.4%. For most of the non-separable functions (52.4%), there was no statistically significant difference in the convergence speed.

Table III summarizes the average accuracy and standard deviation over the 50 independent runs obtained at the last iteration. This table is provided to consider the consistency of the two algorithms, which is indicated by the standard deviation. The smaller the standard deviation, the more consistent the algorithm is in producing results close to the average. The number of functions for which there was an order of magnitude difference in the standard deviation in favor of each algorithm was counted. For gbest PSO, 21.7% of the functions showed a significantly smaller standard deviation compared to that of lbest PSO. On the other hand, lbest PSO had a significantly smaller standard deviation for 31.6% of the functions. Therefore, no one of the algorithms can be labeled as the most consistent over all, or even most of the functions.

What remains to be shown, is that lbest PSO may also be trapped in local minima, that lbest PSO does not always converge slower than gbest PSO, and that lbest PSO does not always maintain a higher diversity than gbest PSO. Figures 1 and 2 respectively illustrates the fitness and diversity profiles of gbest PSO and lbest PSO for selected functions. With reference to the fitness profiles in Figure 1, the following observations are made:

- For the elliptic function, lbest PSO initially improved accuracy faster than gbest PSO, providing evidence of faster convergence for lbest PSO.
- For the shifted schaffer 6 function, the rate at which accuracy is improved was about the same for the two algorithms, but after 1000 iterations, lbest PSO improved accuracy at a much slower rate than gbest PSO, failing to reach the same accuracy than gbest PSO after 5000 iterations. This may indicate convergence of lbest PSO to a worse local minimum than gbest PSO.
- While gbest PSO did converge faster than lbest PSO, lbest PSO did not succeed in obtaining accuracies better

TABLE III

AVERAGE ACCURACY AND STANDARD DEVIATION AT FINAL ITERATION

Function	gbest PSO	lbest PSO	Function	gbest PSO	lbest PSO
f_1	1.07E-012 ± 6.69E-012	1.34E-025 ± 1.13E-025	f_{15}	7.49E+000 ± 1.10E+000	8.88E+000 ± 1.17E+000
f_2	2.15E+000 ± 1.08E+000	7.98E-015 ± 1.85E-015	f_{15}^{Sh}	-2.88E+002 ± 4.95E-001	-2.88E+002 ± 2.94E-001
f_2^{Sh}	-1.37E+002 ± 1.35E+000	-1.40E+002 ± 1.32E-001	f_{16}	1.33E-008 ± 2.60E-008	2.06E+000 ± 1.97E+000
f_2^{R}	3.11E+000 ± 1.20E+000	1.28E+000 ± 7.74E-001	f_{16}^{Sh}	-4.50E+002 ± 1.35E-008	-4.48E+002 ± 2.14E+000
f_2^{ShR}	-1.19E+002 ± 6.04E-002	-1.19E+002 ± 6.28E-002	f_{16}^{ShN}	9.48E+002 ± 1.15E+003	6.51E+003 ± 3.38E+003
f_3	1.06E-010 ± 5.29E-010	1.42E-005 ± 8.04E-005	f_{16}^{R}	4.95E-007 ± 1.57E-006	4.04E+000 ± 3.45E+000
f_4	-1.30E+004 ± 1.12E+003	-1.26E+004 ± 9.46E+002	f_{17}	4.15E+004 ± 1.06E+004	4.38E+004 ± 1.15E+004
f_5	1.41E-072 ± 9.98E-072	3.10E-042 ± 5.37E-042	f_{17}^{Sh}	4.44E+004 ± 1.29E+004	4.61E+004 ± 9.99E+003
f_5^{Sh}	-4.50E+002 ± 4.87E-014	-4.50E+002 ± 8.12E-015	f_{18}	3.74E+004 ± 3.36E+004	2.02E+004 ± 1.59E+004
f_5^{ShR}	4.97E+003 ± 3.24E+003	3.17E+004 ± 6.81E+003	f_{18}^{Sh}	3.48E+004 ± 3.64E+004	2.02E+004 ± 1.45E+004
f_6	2.86E-002 ± 3.48E-002	3.06E-003 ± 4.97E-003	f_{19}	4.20E-004 ± 7.01E-004	2.25E-001 ± 2.15E-001
f_6^{Sh}	-1.80E+002 ± 5.49E-001	-1.80E+002 ± 1.18E-001	f_{20}	5.23E-010 ± 2.79E-009	4.60E-026 ± 5.84E-026
f_6^{R}	2.33E-001 ± 3.02E-001	1.02E-001 ± 1.65E-001	f_{21}	-3.30E+032 ± 7.22E+032	-1.96E+030 ± 4.86E+030
f_6^{ShR}	-8.23E+001 ± 1.65E+001	-7.77E+001 ± 1.86E+001	f_{22}	4.31E-077 ± 2.73E-076	2.64E-046 ± 4.62E-046
f_7	1.04E-072 ± 7.36E-072	4.74E-045 ± 9.46E-045	f_{22}^{Sh}	-4.50E+002 ± 1.84E-013	-4.50E+002 ± 8.12E-015
f_8	-2.40E+001 ± 1.42E+000	-2.45E+001 ± 7.45E-001	f_{23}	6.08E+000 ± 1.05E+001	0.00E+000 ± 0.00E+000
f_9	-7.85E-001 ± 7.98E-003	-7.88E-001 ± 5.68E-003	f_{24}	-3.10E+001 ± 4.72E-014	-3.10E+001 ± 1.06E-009
f_{10}	1.19E-008 ± 2.03E-008	1.90E+000 ± 1.96E+000	f_{25}	7.79E+000 ± 3.25E+000	1.38E-001 ± 5.09E-001
f_{11}	9.69E-131 ± 6.69E-130	2.33E-065 ± 1.26E-064	f_{25}^{ShR}	1.20E+002 ± 3.09E+000	1.21E+002 ± 1.91E+000
f_{11}^{N}	-3.21E+000 ± 4.57E-001	-3.47E+000 ± 3.40E-001	f_{26}	-1.22E+002 ± 2.76E+000	-1.23E+002 ± 1.89E+000
f_{12}	7.18E+001 ± 1.83E+001	7.76E+001 ± 1.75E+001	f_{27}	5.80E+002 ± 4.27E+001	6.04E+002 ± 3.12E+001
f_{12}^{Sh}	-2.53E+002 ± 1.99E+001	-2.42E+002 ± 2.32E+001	f_{28}	8.40E+002 ± 6.74E+001	6.73E+002 ± 2.73E+002
f_{12}^{R}	1.05E+002 ± 2.91E+001	9.76E+001 ± 2.26E+001	f_{29}	8.63E+002 ± 5.84E+001	8.43E+002 ± 9.07E+001
f_{12}^{ShR}	-1.96E+002 ± 4.05E+001	-2.28E+002 ± 2.56E+001	f_{30}	7.81E+002 ± 3.81E+002	2.98E+002 ± 1.15E+002
f_{13}	1.32E+001 ± 1.72E+001	2.08E+001 ± 2.23E+001	f_{31}	8.37E+002 ± 3.00E+002	3.54E+002 ± 8.47E+001
f_{13}^{R}	8.23E+002 ± 1.90E+003	2.39E+002 ± 1.06E+003	f_{32}	9.84E+002 ± 5.55E+002	5.16E+002 ± 1.41E+001
f_{14}	5.96E-001 ± 2.12E-001	3.30E-001 ± 6.47E-002	f_{33}	9.84E+002 ± 5.55E+002	5.16E+002 ± 1.41E+001
f_{33}	1.29E+003 ± 4.38E+002	7.60E+002 ± 4.27E+002	f_{35}	8.90E+002 ± 5.09E+002	5.14E+002 ± 2.02E+000
f_{36}	4.84E+002 ± 4.02E+002	3.32E+002 ± 1.47E+002	f_{37}	3.97E+002 ± 4.23E+001	3.71E+002 ± 1.02E-001

than gbest PSO. Again, this might point to lbest PSO converging on a worse local minimum, or premature convergence to some arbitrary point in the search space. The same behavior is seen for the noisy shifted schwefel 1.2 and shifted rotated weierstrass functions.

- Note how lbest PSO converges at the same rate as gbest PSO on the unimodal schwefel 2.22 function.
- The Shubert function provides clear evidence of lbest PSO stagnating on a very bad solution, failing to improve accuracy.

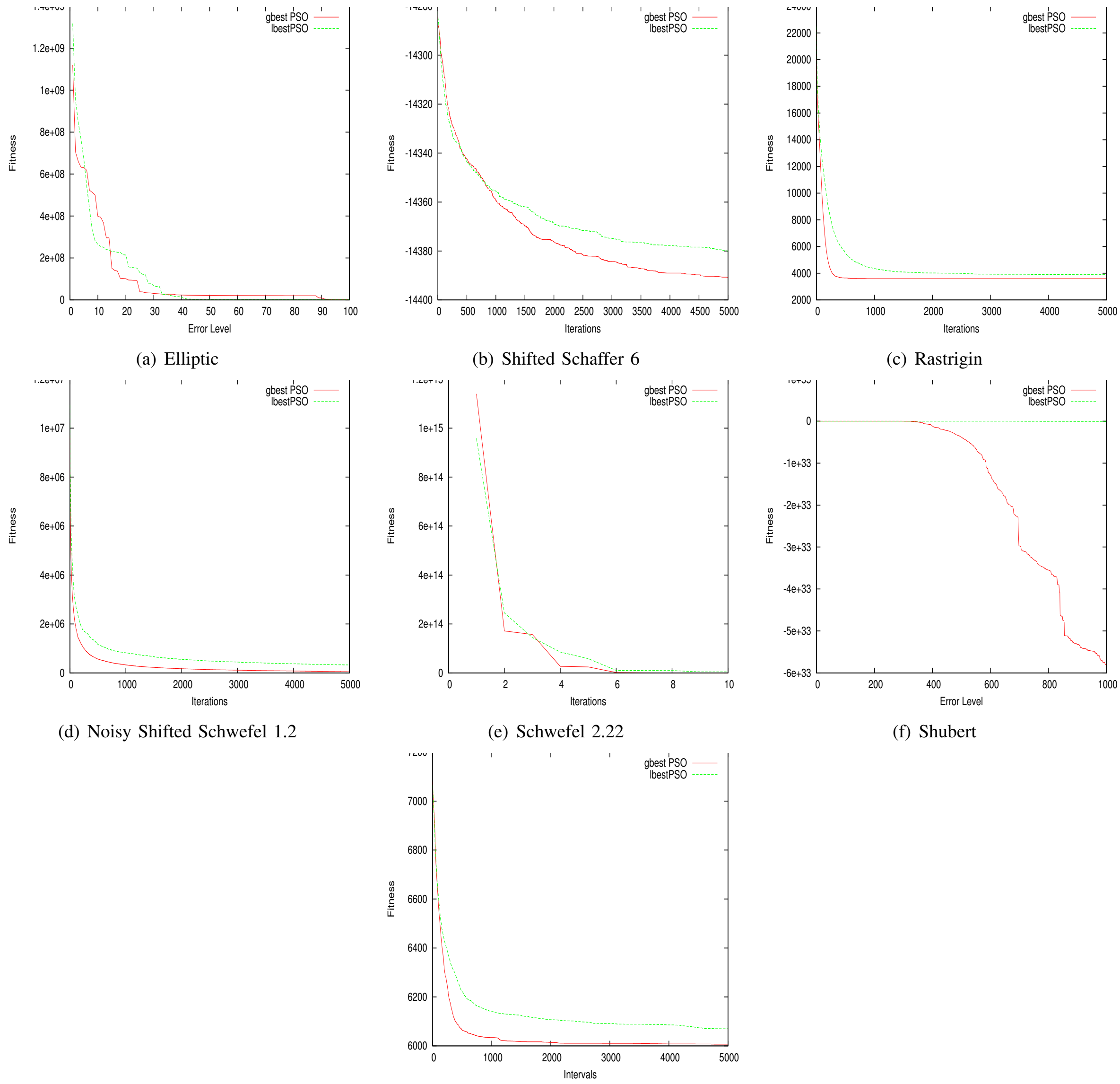
Figure 2 provides evidence that lbest PSO does not always maintain a higher diversity than gbest PSO. For the rastrigin and De Jong F4 functions, both algorithms fail to reduce diversity, with gbest PSO maintaining a significantly larger diversity than lbest PSO. The same trend is shown for the shifted rotated Ackley, shifted schaffer 6, and vincent functions, with the additional behavior that diversities increase over time. The reason for the increase in diversities has to be investigated, but can be due to particles moving outside of the defined boundaries [19]. While lbest PSO did have a larger diversity than gbest PSO for the griewank function after 80 iterations, lbest PSO started off with a lower diversity than gbest PSO.

In summary, the results above have clearly shown that neither of the two algorithms can be cosidered the outright best for any of the function classes (these conclusions are also supported by the study in [3]). Previously published statements to this effect should therefore not be considered as a ground truth. Rather, what the results above show, without any doubt, is that the choice between gbest PSO or lbest PSO is definitely function dependent, not function class dependent (considering the function classes as used in this study). In order to determine for which class of functions one algorithm is better than another, more characteristics of the function landscape [32] have to be considered, not just the modality and separability. Such a study is left for future research.

V. CONCLUSIONS

A number of opinions have been raised about which of the global best (gbest) particle swarm optimization (PSO) and local best (lbest) PSO algorithms is best to use. The general trend of these opinions is that gbest PSO should not be used, citing premature convergence, fast convergence, and quick reduction in diversity as reasons. Instead, lbest PSO should be preferred due to its slower convergence, better exploration, and better solutions being found. However, these opinions were based on intuitive expectations about the performance of the algorithms and empirical analysis on a very limited set of benchmark functions.

This paper has conducted an elaborate empirical analysis to compare the performance of gbest PSO and lbest PSO on a large set of 60 benchmark functions, with respect to solution accuracy, success rate, efficiency and consistency. The diversity profiles of the algorithms were also compared on these functions. Based on the statistical analysis done, the main conclusion of this study is that neither of the two algorithms can be considered the preferred algorithm for any



(a) Elliptic (b) Shifted Schaffer 6 (c) Rastrigin

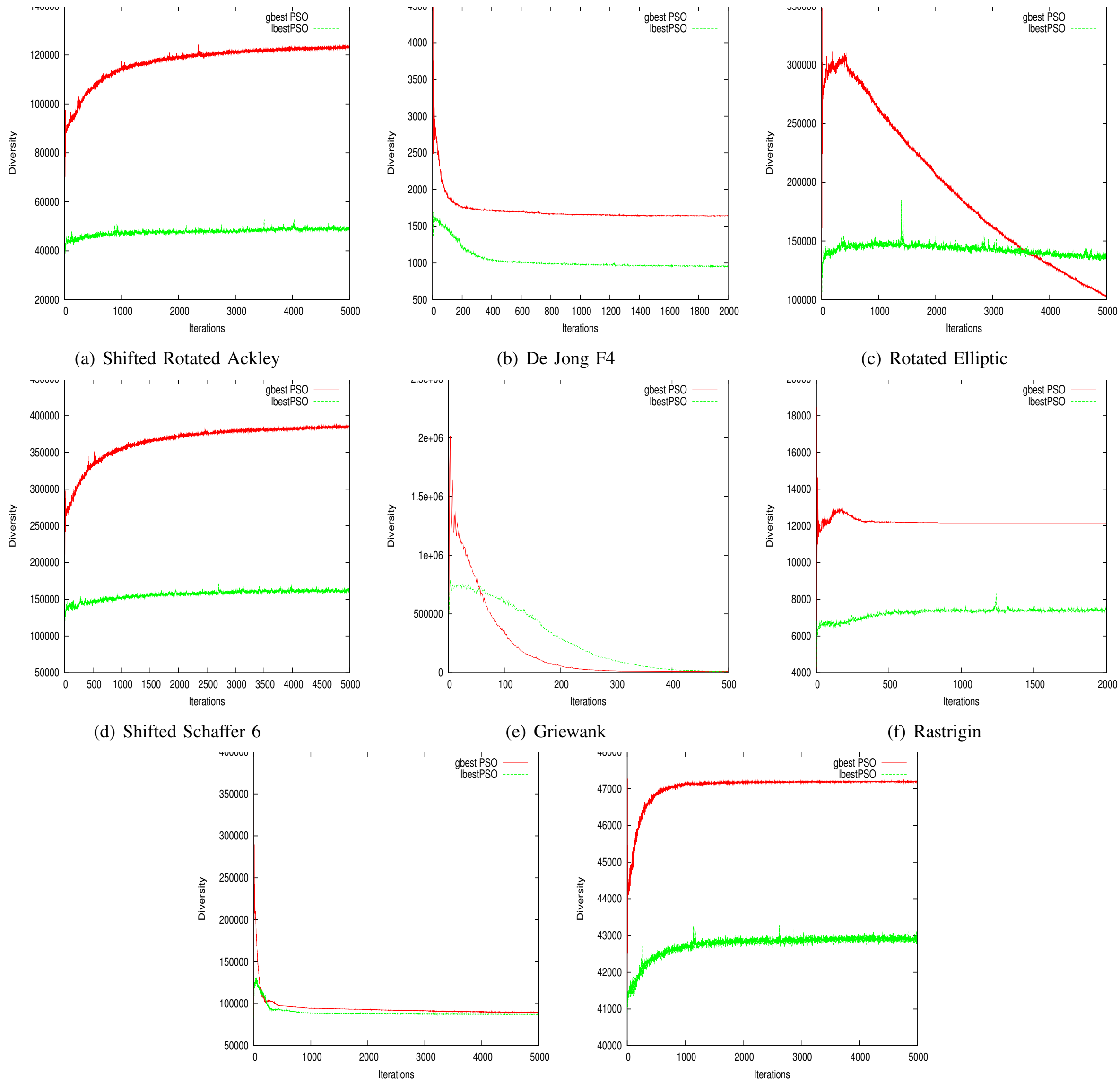
(d) Noisy Shifted Schwefel 1.2 (e) Schwefel 2.22 (f) Shubert

(g) Shifted Rotated Weierstrass

Fig. 1. Fitness Profiles for Selected Functions

of the main classes of functions considered, and statements to the contrary should be considered with a grain of salt. Over all of the functions, the two algorithms performed very similarly with respect to solution accuracy, while gbest PSO performed slightly better than lbest PSO with respect to success rate and efficiency. With respect to consistency, lbest PSO performed slightly better than gbest PSO.

The main point that this study makes is that, if the objective is to find the most optimal algorithm for a specific optimization problem, then the neighborhood topology used has to be included as one of the parameters tuned. However, when the objective is to determine if a specific change to PSO provides improved performance, then it is arbitrary whether gbest PSO or lbest PSO is used. The important aspect of such a study is that the base algorithm used in the comparison should be the same, with the only difference the new change. As an example, if the objective is to evaluate if Gaussian mutation of particle positions improve on performance, it does not matter if a gbest PSO or a lbest PSO is used. Assuming that a gbest PSO is used, then what does matter is that if the performance is compared with other mutation-based PSO algorithms, then these algorithms must also use the star neighborhood topology. If not, one will not know if differences in performance are due to the difference in mutation operators, or in the neighborhood



(a) Shifted Rotated Ackley

(b) De Jong F4

(c) Rotated Elliptic

(d) Shifted Schaffer 6

(e) Griewank

(f) Rastrigin

(g) Shifted Rosenbrock

(h) Vincent

Fig. 2. Diversity Profiles for Selected Functions

topology used.

References

[1] R. Eberhart and J. Kennedy, "A New Optimizer using Particle Swarm Theory," in *Proceedings of the Sixth International Symposium on Micromachine and Human Science*, 1995, pp. 39–43.

[2] R. Eberhart, P. Simpson, and R. Dobbins, *Computational Intelligence PC Tools*, 1st ed. Academic Press Professional, 1996.

[3] D. Bratton and J. Kennedy, "Defining a Standard for Particle Swarm Optimization," in *Proceedings of the IEEE Swarm Intelligence Symposium*, 2007, pp. 120–127.

[4] M. Clerc, "From Theory to Practice in Particle Swarm Optimization," in *Handbook of Swarm Intelligence: Adaptation, Learning, and Optimization*. Springer, 2010, vol. 8, pp. 3–36.

[5] W. Elshamy, H. Emara, and A. Bahgat, "Clubs-based Particle Swarm Optimization," in *Proceedings of the IEEE Swarm Intelligence Symposium*, 2007.

[6] A. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, 1st ed. Wiley, 2005.

[7] J. Kennedy, "Small Worlds and Mega-Minds: Effects of Neighborhood Topology on Particle Swarm Performance," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 1999, pp. 1931–1938.

[8] ——, "Stereotyping: Improving Particle Swarm Performance with Cluster Analysis," in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 2, July 2000, pp. 1507–1512.

[9] J. Kennedy, R. Eberhart, and Y. Shi, *Swarm Intelligence*. Morgan Kaufmann, 2001.

[10] J. Kennedy and R. Mendes, "Population Structure and Particle Performance," in *Proceedings of the IEEE Congress on Evolutionary Computation*. IEEE Press, 2002, pp. 1671–1676.

[11] ——, "Neighborhood Topologies in Fully-Informed and Best-of-Neighborhood Particle Swarms," in *Proceedings of the IEEE International Workshop on Soft Computing in Industrial Applications*, June 2003, pp. 45–50.

[12] J. Lane, A. Engelbrecht, and J. Gain, "Particle Swarm Optimization with Spatially Meaningful Neighbours," in *Proceedings of the IEEE Swarm Intelligence Symposium*, 2008.
[13] Y. Marinakis and M. Marinaki, "Particle Swarm Optimization with Expanding Neighborhood Topology for the Permutation Flowshop Scheduling Problem," *Soft Computing*, 2013.
[14] R. Mendes, J. Kennedy, and J. Neves, "Watch thy Neighbor or How the Swarm can Learn from its Environment," in *Proceedings of the IEEE Swarm Intelligence Symposium*, April 2003, pp. 88–94.
[15] ——, "The Fully Informed Particle Swarm: Simpler, Maybe Better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204–210, 2004.
[16] E. Peer, F. van den Bergh, and A. Engelbrecht, "Using Neighborhoods with the Guaranteed Convergence PSO," in *Proceedings of the IEEE Swarm Intelligence Symposium*. IEEE Press, 2003, pp. 235–242.
[17] T.-P. G, A. Reyes-Medina, and J. Ramirez-Tores, "A Statistical Study of the Effects of Neighborhood Topologies in Particle Swarm Optimization," in *Studies in Computational Intelligence*. Springer, 2011, vol. 343, pp. 179–192.
[18] Y. Shi and R. Eberhart, "A Modified Particle Swarm Optimizer," in *Proceedings of the IEEE Congress on Evolutionary Computation*, May 1998, pp. 69–73.
[19] A. Engelbrecht, "Particle Swarm Optimization: Velocity Initialization," in *Proceedings of the IEEE Congress on Evolutionary Computation*. IEEE Press, 2012.
[20] T. Huang and A. Mohan, "Significance of Neighborhood Topologies for the Reconstruction of Microwave Images using Particle Swarm Optimization," in *Proceedings of the Asia-Pacific Microwave Conference*, vol. 1, 2005.
[21] M. Richards and D. Ventura, "Dynamic Sociemery in Particle Swarm Optimization," in *Proceedings of the Sixth International Conference on Computational Intelligence and Natural Computing*, 2003.
[22] F. van den Bergh and A. Engelbrecht, "A New Locally Convergent Particle Swarm Optimizer," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 2002, pp. 96–101.
[23]
[24] M. Clerc and J. Kennedy, "The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.
[25] I. Trelea, "The Particle Swarm Optimization Algorithm: Convergence Analysis and Parameter Selection," *Information Processing Letters*, vol. 85, no. 6, pp. 317–325, 2003.
[26] F. van den Bergh and A. Engelbrecht, "A Study of Particle Swarm Optimization Particle Trajectories," *Information Sciences*, vol. 176, no. 8, pp. 937–971, 2005.
[27] A. Engelbrecht, B. Masiye, and G. Pampara, "Niching Ability of Basic Particle Swarm Optimization Algorithms," in *Proceedings of the IEEE Swarm Intelligence Symposium*, 2005.
[28] R. Eberhart and Y. Shi, "Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 1, July 2000, pp. 84–88.
[29] R. Salomon, "Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions," *BioSystems*, vol. 39, pp. 263–278, 1996.
[30] P. Suganthan, N. Hansen, J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," Tech. Rep., 2005.
[31] "Measuring Exploration/Exploitation in Particle Swarms using Swarm Diversity," in *Proceedings of the IEEE Congress on Evolutionary Computation*. IEEE Press, 2008.
[32] K. Malan and A. Engelbrecht, "Review of Fitness Landscapes and A Way Forward," *accepted for Information Sciences*, 2013.

APPENDIX

The definitions of the benchmark functions used in this study is provided in this appendix, together with detail on the domain of each function. Many of the functions below are equivalent to functions defined in the CEC 2005 benchmark set [30]. Such equivalencies are indicated by giving the CEC 2005 benchmark number with the superscript CEC.

f_1, the absolute value function, defined as

$$f_1(\mathbf{x}) = \sum_{j=1}^{n_x} |x_i| \tag{6}$$

with each $x_j \in [-100, 100]$.

f_2, the ackley function, defined as

$$f_2(\mathbf{x}) = -20e^{-0.2\sqrt{\frac{1}{n}\sum_{j=1}^{n_x} x_j^2}} - e^{\frac{1}{n}\sum_{j=1}^{n_x}\cos(2\pi x_j)} + 20 + e \tag{7}$$

with each $x_j \in [-32.768, 32.768]$. Shifted, rotated, and rotated and shifted verions of the ackley function were used, respectively referred to as f_2^{Sh}, f_2^{R} and f_2^{ShR}. Function f_2^{ShR} is equivalent to function f_8^{CEC}.

f_3, the alpine function, defined as

$$f_3(\mathbf{x}) = \left(\prod_{j=1}^{n_x} \sin(x_j)\right) \sqrt{\prod_{j=1}^{n_x} x_j} \tag{8}$$

with each $x_j \in [-10, 10]$.

f_4, the egg holder function, defined as

$$\begin{aligned} f_4(\mathbf{x}) \;\; = \;\; & \sum_{j=1}^{n_x-1} \Bigg(-(x_{j+1}+47)\sin(\sqrt{|x_{j+1}+x_j/2+47|}) \\ + \;\; & \sin(\sqrt{|x_j-(x_{j+1}+47)|})(-x_j)\Bigg) \end{aligned} \tag{9}$$

with each $x_j \in [-512, 512]$.

f_5, the elliptic function, defined as

$$f_5(\mathbf{x}) = \sum_{j=1}^{n_x} (10^6)^{\frac{j-1}{n_x-1}} \tag{10}$$

with each $x_j \in [-100, 100]$. Shifted, rotated, and rotated and shifted verions of the elliptic function were used, respectively referred to as f_5^{Sh}, f_5^{R} and f_5^{ShR}. Function f_5^{ShR} is equivalent to function f_3^{CEC}.

f_6, the griewank function, defined as

$$f_6(\mathbf{x}) = 1 + \frac{1}{4000}\sum_{j=1}^{n_x} x_j^2 - \prod_{j=1}^{n_x}\cos\left(\frac{x_j}{\sqrt{j}}\right) \tag{11}$$

with each $x_j \in [-600, 600]$. Shifted, rotated, and rotated and shifted verions of the elliptic function were used, respectively referred to as f_6^{Sh}, f_6^{R} and f_6^{ShR}. For function f_6^{ShR}, each $x_j \in [0, 600]$, which means that the global minimum is outside of the bounds. Function f_6^{ShR} is equivalent to function f_7^{CEC}. with bounds as given above.

f_7, the hyperellipsoid function, defined as

$$f_7(\mathbf{x}) = \sum_{j=1}^{n_x} j x_j^2 \tag{12}$$

with each $x_j \in [-5.12, 5.12]$.

f_8, the michalewicz function, defined as

$$f_8(\mathbf{x}) = -\sum_{j=1}^{n_x} \sin(x_j)\left(\sin\left(\frac{j x_j^2}{\pi}\right)\right)^{2m} \tag{13}$$

with each $x_j \in [0, \pi]$ and $m = 10$.

f_9, the norwegian function, defined as

$$f_9(\mathbf{x}) = \prod_{j=1}^{n_x} \left(\cos(\pi x_j^3) \left(\frac{99 + x_j}{100} \right) \right) \quad (14)$$

with each $x_j \in [-1.1, 1.1]$.

f_{10}, the quadric function, defined as

$$f_{10}(\mathbf{x}) = \sum_{i=1}^{n_x} \left(\sum_{j=1}^{i} x_j \right)^2 \quad (15)$$

with each $x_j \in [-100, 100]$.

f_{11}, the quartic function, defined as

$$f_{11}(\mathbf{x}) = \sum_{j=1}^{n_x} j x_j^4 \quad (16)$$

with each $x_j \in [-1.28, 1.28]$. A noisy version of the quartic function, referred to as de jong's f4 function, was generated as follows:

$$f_{11}^N(\mathbf{x}) = \sum_{j=1}^{n_x} (j x_j^4 + N(0, 1)) \quad (17)$$

The domain was the same as that of the quartic function.

f_{12}, the rastrigin function, defined as

$$f_{12}(\mathbf{x}) = 10 n_x + \sum_{j=1}^{n_x} \left(x_j^2 - 10 \cos(2\pi x_j) \right) \quad (18)$$

with $x_j \in [-5.12, 5.12]$. Shifted, rotated, and rotated and shifted verions of the rastrigin function were used, respectively referred to as f_{12}^{Sh}, f_{12}^{R} and f_{12}^{ShR}. Function f_{12}^{ShR} is equivalent to function f_{10}^{CEC}.

f_{13}, the rosenbrock function, defined as

$$f_{13}(\mathbf{x}) = \sum_{j=1}^{n_x - 1} \left(100(x_{j+1} - x_j^2)^2 + (x_j - 1)^2 \right) \quad (19)$$

with $x_j \in [-30, 30]$. Shifted, rotated, and rotated and shifted verions of the rastrigin function were used, respectively referred to as f_{13}^{Sh}, f_{13}^{R} and f_{13}^{ShR}. Function f_{13}^{Sh}'s domain was $[-100, 100]$ to be equivalent to f_6^{CEC}. Function f_{13}^{R}'s domain was also set to $[-100, 100]$. f_{10}^{CEC}.

f_{14}, the salomon function, defined as

$$f_{14}(\mathbf{x}) = -\cos(2\pi \sum_{j=1}^{n_x} x_j^2) + 0.1 \sqrt{\sum_{j=1}^{n_x} x_j^2} + 1 \quad (20)$$

with $x_j \in [-100, 100]$.

f_{15}, the schaffer 6 function, defined as

$$f_{15}(\mathbf{x}) = \sum_{j=1}^{n_x - 1} \left(0.5 + \frac{sin^2(x_j^2 + x_{j+1}^2) - 0.5}{(1 + 0.001(x_j^2 + x_{j+1}^2))^2} \right) \quad (21)$$

with each $x_j \in [-100, 100]$. A shifted and rotated, expanded version of the schaffer 6 function was used, referred to as f_{15}^{ShRE}, which is equivalent to f_{14}^{CEC}.

f_{16}, the schwefel 1.2 function, defined as

$$f_{16}(\mathbf{x}) = \sum_{j=1}^{n_x} \left(\sum_{k=1}^{j} x_k \right)^2 \quad (22)$$

with $x_j \in [-100, 100]$. Shifted, rotated, and noisy shifted verions of the schwefel 1.2 function were used, respectively referred to as f_{16}^{Sh}, f_{16}^{R} and f_{16}^{ShN}. Function f_{16}^{Sh} is equivalent to f_2^{CEC}, and f_{16}^{ShN} is equivalent to f_4^{CEC}, defined as

$$f_{16}(\mathbf{x}) = \sum_{j=1}^{n_x} \left(\sum_{k=1}^{j} x_k \right)^2 (1 + 0.4|N(0, 1)|) \quad (23)$$

f_{17}, the schwefel 2.6 function, defined as

$$f_{17}(\mathbf{x}) = \max_j \{ |\mathbf{A}_j \mathbf{x} - \mathbf{B}_j| \} \quad (24)$$

with $x_j \in [-100, 100]$, $a_{ji} \in \mathbf{A}$ is uniformly sampled from $U(-500, 500)$ such that $\det(\mathbf{A}) \neq 0$, and each $\mathbf{B}_j = \mathbf{A}_j \mathbf{x}$. Function f_{17} is equivalent to f_5^{CEC}. A shifted version of schwefel 2.6 was also implemented, referred to as f_{17}^{Sh}.

f_{18}, the schwefel 2.13 function, defined as

$$f_{18}(\mathbf{x}) = \sum_{j=1}^{n_x} (\mathbf{A}_j - \mathbf{B}_j(\mathbf{x}))^2 \quad (25)$$

with each $x_j \in [-\pi, \pi]$, and

$$\mathbf{A}_j = \sum_{i=1}^{n_x} (a_{ji} \sin \alpha_i + b_{ji} \cos \alpha_i)$$

$$\mathbf{B}_j(\mathbf{x}) = \sum_{i=1}^{n_x} (a_{ji} \sin x_i + b_{ji} \cos x_i$$

where $a_{ji} \in \mathbf{A}$ and $b_{ji} \in \mathbf{B}$ with $a_{ji}, b_{ji} \sim U(-100, 100)$, and $\alpha_i \sim U(-\pi, \pi)$. This function is equivalent to f_{12}^{CEC}. A shifted version of schwefel 2.13 was also implemented, referred to as f_{18}^{Sh}.

f_{19}, the schwefel 2.21 function, defined as

$$f_{19}(\mathbf{x}) = \max_j \{ |x_j|, 1 \leq j \leq n_x \} \quad (26)$$

with each $x_j \in [-100, 100]$.

f_{20}, the schwefel 2.22 function, defined as

$$f_{20}(\mathbf{x}) = \sum_{j=1}^{n_x} |x_j| + \prod_{j=1}^{n_x} |x_j| \quad (27)$$

with each $x_j \in [-10, 10]$.

f_{21}, the shubert function, defined as

$$f_{21}(\mathbf{x}) = \prod_{j=1}^{n_x} \left(\sum_{i=1}^{5} (i \cos((i+1) x_j + i) \right) \quad (28)$$

with each $x_j \in [-10, 10]$.

f_{22}, the spherical function, defined as

$$f_{22}(\mathbf{x}) = \sum_{j=1}^{n_x} x_i^2 \quad (29)$$

with each $x_j \in [-5.12, 5.12]$. A shifted version was implemented, referred to as f_{22}^{Sh} and equivalent to f_1^{CEC}.

f_{23}, the step function, defined as

$$f_{23}(\mathbf{x}) = \sum_{j=1}^{n_x} (\lfloor x_j + 0.5 \rfloor)^2 \quad (30)$$

with each $x_j \in [-100, 100]$.

f_{24}, the vincent function, defined as

$$f_{24}(\mathbf{x}) = -\left(1 + \sum_{j=1}^{n_x} \sin(10\sqrt{x_j})\right) f_{27} - \quad (31)$$

with each $x_j \in [0.25, 10]$.

f_{25}, the weierstrass function, defined as

$$\begin{aligned} f_{25}(\mathbf{x}) &= \sum_{j=1}^{n_x}\left(\sum_{i}^{20}(a^i \cos(2\pi b^i(x_j + 0.5)))\right) \\ &- n_x \sum_{i=1}^{20}(a^i \cos(\pi b^i)) \end{aligned} \quad (32)$$

with each $x_j \in [-0.5, 0.5]$, $a = 0.5$ and $b = 3$. A shifted and rotated version of the weierstrass function was implemented, referred to as f_{25}^{ShR}, equivalent to f_{11}^{CEC}.

f_{26}, a shifted expansion of the griewank and rosenbrock functions (f_6 and f_{13} respectively), equivalent to f_{13}^{CEC}. Each $x_j \in [-3, 1]$.

f_{37}, which are all composition functions, respectively equivalent to f_{15}^{CEC} to f_{25}^{CEC}. All functions have $x_j \in [-5, 5]$, except f_{37} for which $x_j \in [2, 5]$.