Identifying and Exploiting the Scale of a Search Space in Particle Swarm Optimization

Yasser Gonzalez-Fernandez
School of Information Technology
York University
Toronto, Canada
ygf@yorku.ca

Stephen Chen
School of Information Technology
York University
Toronto, Canada
sychen@yorku.ca

ABSTRACT

Multi-modal optimization involves two distinct tasks: identifying promising attraction basins and finding the local optima in these basins. Unfortunately, the second task can interfere with the first task if they are performed simultaneously. Specifically, the promise of an attraction basin is often estimated by the fitness of a single sample solution, so an attraction basin represented by a random sample solution can appear to be less promising than an attraction basin represented by its local optimum. The goal of thresheld convergence is to prevent these biased comparisons by disallowing local search while global search is still in progress. Ideally, thresheld convergence achieves this goal by using a distance threshold that is correlated to the size of the attraction basins in the search space. In this paper, a clustering-based method is developed to identify the scale of the search space which thresheld convergence can then exploit. The proposed method employed in the context of a multi-start particle swarm optimization algorithm has led to large improvements across a broad range of multi-modal problems.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

Keywords

Metaheuristics; particle swarm optimization; fitness landscape; thresheld convergence; multi-start methods

1. INTRODUCTION

An effective optimization algorithm must consider two essentially conflicting tasks, namely exploration and exploitation [21]. The first task entails the global exploration of the search space to locate regions with high-quality solutions, and the second task requires a local exploitation around the best solutions in order to improve them to an optimal level. The ratio at which these two task should be balanced is closely related to the characteristics of the optimization problem. In multi-modal search spaces, it is expected that a considerable search effort – i.e. function evaluations (FEs) – should be expended in finding the most promising areas of the search space before concentrating the search in these regions. Nonetheless, many search heuristics do not explicitly separate these processes [6, 17].

Particle swarm optimization (PSO) [14] is an example of such a heuristic with no explicit distinction between global and local search. Each particle is attracted towards its personal best position and the best position of a neighbouring member of the swarm. Even with the use of a local communication topology, these attractors quickly concentrate the search around a few regions of the search space. This level of convergence can work well in unimodal search spaces, but it can lead to poor performance on multi-modal problems.

"Thresheld convergence" [6, 17] is a technique specially designed to separate the processes of exploration and exploitation in search heuristics. It is based on the hypothesis that concurrent exploration and exploitation can interfere with the correct assessment of the quality of an "attraction basin" (regions in which greedy descent will lead to the same local optimum). If the promise of an attraction basin is estimated by the fitness of a single solution, an attraction basin represented by a random solution will often appear to be less promising than an attraction basin represented by its local optimum – even if the optimum in the first attraction basin is fitter than the optimum of the second. In order to prevent these biased comparisons, local search is "held back" during the initial phases of the search heuristic by means of a distance threshold.

Ideally, thresheld convergence should use a threshold value that is correlated to the size of the attraction basins in the search space. However, earlier uses of thresheld convergence in PSO [6, 7] have not tried to estimate this value, but they have instead set the initial threshold to a fraction of the main diagonal of the search space and then used simple mechanisms to reduce the threshold over time. In this paper, we develop a clustering-based method to explicitly identify the scale of the search space which thresheld convergence can then exploit. The proposed method is employed in the context of a multi-start particle swarm optimization algorithm where it achieves positive results on a broad range of multi-modal problems.

The development of this result begins with the necessary background on PSO and thresheld convergence in Section 2 before illustrating our research motivation in Section 3. Sec-

GECCO'14, July 12–16, Vancouver, BC, Canada.

ACM 978-1-4503-2662-9/14/07 ...$15.00.
http://dx.doi.org/10.1145/2576768.2598280.

tion 4 then presents the clustering approach for the identification of the scale of the search space and the newly-proposed multi-start PSO with thresheld convergence. The performance of the new technique is then empirically studied in Section 5 before Section 6 gives a summary of the main results and the conclusions of the paper.

2. BACKGROUND

The first part of this section briefly describes the variant of PSO considered as the baseline of our study. Thresheld convergence and its integration with PSO is then presented.

2.1 Particle Swarm Optimization

The initial development of PSO includes inspirations from "bird flocking, fish schooling, and swarming theory" [14]. PSO keeps a group of candidate solutions to the optimization problem – deemed as particles in a swarm – and performs updates to their positions until a termination criterion is satisfied. In addition to its position, each particle maintains a velocity vector that represents a displacement with respect to its current position which is used to move the particle in the search space. The velocity is updated at each iteration by attracting each particle to the best position it has found so far and towards the best position achieved by any neighbouring particle. For each individual particle, a neighbourhood is defined as a group of particles in the swarm that it is able to communicate with.

A particle identified with the index i among a swarm of size S is represented in PSO by three n-dimensional real vectors: 1) its position in the search space x_i; 2) the best position it has individually found $pbest_i$; and 3) its velocity v_i. Additionally, the neighbourhood notion allows us to identify by $lbest_i \in \mathbb{R}^n$ the best position achieved by any neighbouring particle, and consequently the best position globally in the swarm by $gbest \in \mathbb{R}^n$.

The initial positions are sampled from a continuous uniform distribution within the bounds of the search space, while the velocities are initialized to zero [9]. At each iteration, PSO updates the swarm by updating the velocity and the position of each particle $i = 1, \ldots, S$ in every dimension $j = 1, \ldots, n$ according to

$$v_{i,j} = \chi \left(v_{i,j} + c_1 \epsilon_{1j} (pbest_{i,j} - x_{i,j}) + c_2 \epsilon_{2j} (lbest_{i,j} - x_{i,j})\right)$$
$$x_{i,j} = x_{i,j} + v_{i,j};$$

where $\epsilon_1, \epsilon_2 \in (0,1)^n$ are uniform random vectors simulated independently for each particle that perturb the differences $(pbest_{i,j} - x_{i,j})$ and $(lbest_{i,j} - x_{i,j})$, respectively; c_1 and c_2 are constants that scale the influence of the cognitive component $(pbest_{i,j} - x_{i,j})$ and the social component $(lbest_{i,j} - x_{i,j})$; and χ is a parameter known as constriction factor that balances the global and local search capabilities of the update rule. If after updating a particle it goes out of the search space boundaries, its position and velocity will be reset following the Reflect-Z technique [10].

The use of a swarm of 50 particles ($S = 50$) in a ring topology (every particle connected to two other particles in a circular network) with $c_1 = c_2 = 2.05$ and $\chi = 0.72984$ corresponds to our implementation of the standard PSO 2007 [4]. This configuration is used as the baseline of the study presented in the paper.

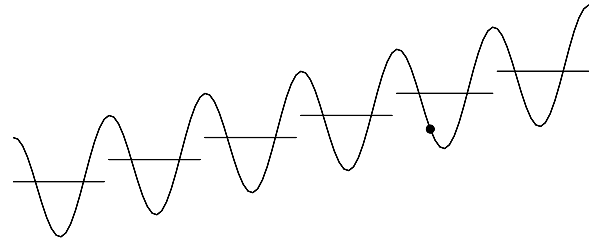

Figure 1: A search space with attraction basins of the same size and shape. The horizontal lines represent the average fitness of a random solution taken from each attraction basin. A better-than-random solution – represented by the dot – may be fitter than random solutions drawn from better basins.

2.2 Thresheld Convergence

Thresheld convergence [6, 17] is based on the hypothesis that concurrent exploitation and exploration can interfere with the effectiveness of a search technique's processes for exploration. For heuristic search techniques to exhibit a good behaviour on multi-modal fitness landscapes, they require a combination of both exploratory and exploitative moves. The exploratory part of the search process entails finding the regions of the search space where the best local optima are located, while the exploitative part must find the best possible solution – i.e. a particular local optimum – within a given region.

In general, heuristic optimization techniques estimate the promise of a region of the search space based on the evaluation of random solutions. PSO maintains a population of *pbests* and, in order to redirect the focus of the search to another (more promising) region of the search space, a sample solution from the new region that is better than the existing *pbest* representative(s) of the current region must be found. However, simultaneously combining exploration and exploitation can interfere with the process of correctly assessing the quality of a newly sampled region.

Figure 1 presents a search space with attraction basins that are all of the same size and shape – e.g. a sinusoid super-imposed over a linear slope. The average fitness of random solutions from an attraction basin is correlated to the fitness of the local optimum within that basin. Thus, given two random solutions from different attraction basins, there is a reasonable expectation that the fitter solution will be from the basin with the fitter local optimum. However, the same cannot be said for the comparison of a random solution from one attraction basin with a better-than-random solution from a second basin.

Starting from an initial solution, let us define any change that leads to a solution in a new attraction basin as an exploratory/global search step and any change that leads to a solution in the same basin as an exploitative/local search step. Without any other information, the first solution from an attraction basin can be considered to be a random solution. Subsequently, a second solution in the same basin that is better than the first solution can be considered to be a better-than-random solution. Concurrent exploitation, which can cause existing attraction basins to become represented by better-than-random solutions, can interfere with a search technique's exploratory processes by biasing its assessment of attraction basins to favour currently discovered (and partially exploited) basins.

The goal of thresheld convergence is to delay exploitation/local search and thus prevent inaccurate comparisons of attraction basins. Convergence is delayed by disallowing local search steps that are less than a distance threshold. As the threshold value decays to zero, greedier local search steps become possible. Conversely, until the threshold is sufficiently small, the search technique is forced to focus on the global search aspect of finding the best attraction basin in the search space.

The positive effects of thresheld convergence have been demonstrated on multiple metaheuristics (see e.g. [2, 3, 7]). The use of thresheld convergence in these techniques is based on the addition of a minimum step-size to the mechanisms that control the exploration/exploitation balance. In PSO, the threshold function is used during the update rule for the *pbests*. The following pseudo-code shows the original *pbest* update condition:

> **if** $f(x_i) < f(pbest_i)$ **then**
> $pbest_i \leftarrow x_i$
> **end if**

while the code fragment below shows the update condition for PSO with thresheld convergence proposed in [7]:

> **if** $f(x_i) < f(pbest_i)$ **and**
>   distance(x_i, $pbest_i$) $\geq$ *threshold* **and**
>   distance(x_i, $lbest_i$) $\geq$ *threshold* **then**
> $pbest_i \leftarrow x_i$
> **end if**

With thresheld convergence, a *pbest* is updated only if the new position is located a minimum distance away from both the old *pbest* and *lbest* positions. Ideally, each of these three solutions will be in different attraction basins, and the elimination of local search can remove a source of bias from the comparison of the potential fitness of these attraction basins.

3. MOTIVATION

In order to illustrate the research motivation, we have chosen the well-known Rastrigin's function as defined in [19] (hereafter simply Rastrigin). This function is a non-convex minimization problem frequently used as a performance test for numerical algorithms. It constitutes a typical example of a non-linear and multi-modal function, defined as

$$f(\mathbf{x}) = An + \sum_{i=1}^{n} \left[x_i^2 - A\cos(2\pi x_i)\right] ,$$

where $A = 10$ and $x_i \in [-5.12, 5.12]$. The function has a single global minimum at $\mathbf{x} = \mathbf{0}$ with $f(\mathbf{x}) = 0$, but it is a fairly difficult optimization problem due the large number of local minima. The local optima are located at the integer values of a regular grid of size 1 – thus, there are n^{11} local optima in the n-dimensional search space defined above.

By analyzing the function definition, we know that the minimum distance between adjacent local optima is 1 and the maximum distance is $\sqrt{n}$. These values represent moving from one local optimum to another only through one dimension or through the diagonal, respectively. Also, given a search point, the nearest local optima in the same basin can be obtained by rounding each component to its nearest integer value. As will be shown, this information is very useful for understanding the results of PSO on this problem and to design a strategy to improve its performance.

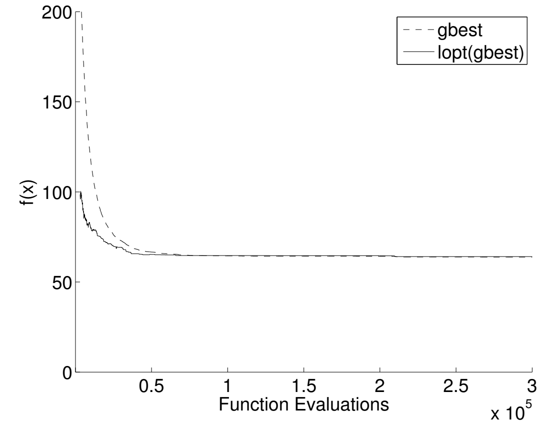

Figure 2: Results of PSO on Rastrigin for $n = 30$. The plot shows the average performance over 51 trials: mean $f(x)$ of *gbest* is 63.8.

To provide a baseline, standard PSO was run 51 times on Rastrigin for $n = 30$ dimensions. Figure 2 illustrates the progress of the algorithm in terms of the FEs up to a maximum allocation of 300,000. The dashed line represents the fitness of the best solution found so far, and the solid line the fitness of the local optimum corresponding to that solution (i.e. the local optimum in the same attraction basin). Therefore, changes in the solid line represent global search, and the convergence of the dashed line onto the solid line is evidence of local search.

From the plot, it can be seen that the solid line quickly plateaus after the first 10–20% of the FEs. This suggests that PSO uses 10–20% of its running time looking for a promising area in the search space and 80–90% performing local search in this area to find the corresponding local optimum. Since only the best generated solutions are stored as *pbests*, PSO eventually becomes highly exploitative around the regions of the search space containing these *pbest* positions as the velocities slow down to zero [8], even with the (local) ring topology.

This strong convergence is a property that makes PSO a very effective method for unimodal search spaces, but it is not an appropriate strategy for multi-modal functions. In order to obtain better results on highly multi-modal problems, PSO needs better global search capabilities – e.g. the ratio of global and local search should be inverted. Several PSO extensions tackling this drawback have been proposed in the literature, including niching techniques (see e.g. [1, 5]), multi-swarm systems (see e.g. [11, 22]), thresheld convergence (covered in Section 2.2), and many others.

One of the most simple alternatives (studied in detail in [20]) is running PSO multiple times from different initial solutions in order to increase the chance of convergence to the global optimum. Figure 3 shows the results of a simple multi-start strategy consisting of five completely independent restarts with a static allocation of 60,000 FEs each. The use of multiple restarts improves the performance on Rastrigin by 21.6% due to the additional exploration of the search space introduced by the resampled initial solutions. Nonetheless, going back to solutions sampled uniformly within the bounds of the search space wastes most of

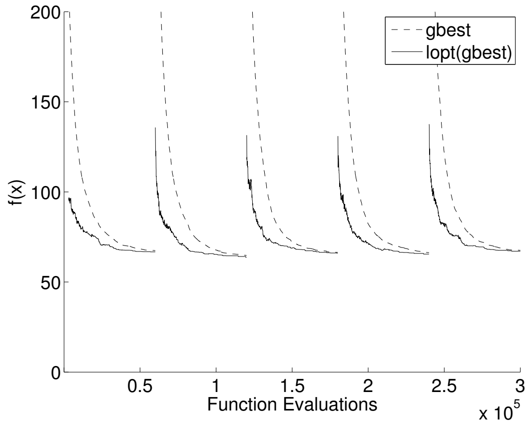

Figure 3: Results of five restarts of PSO – with 1/5 of the allotted FEs each – on Rastrigin for $n = 30$. The plot shows the average performance over 51 trials: mean $f(x)$ of *gbest* is 50.0. The discordance between this value and the chart is caused by the line plot averaging the individual restarts and not the best global result of each trial.

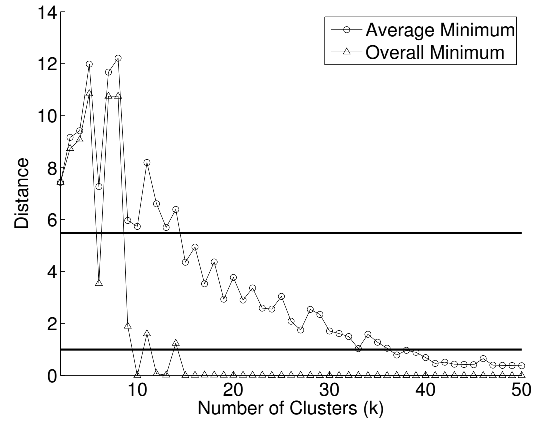

Figure 4: Average and overall minimum distance between the centroids of every pair of the $k \in [2, 50]$ clusters obtained from the population of *pbests*. The horizontal lines show the minimum and maximum distance between adjacent local optima in Rastrigin for $n = 30$.

the FEs and the algorithm produces, on average, five replicas of the behaviour of the single-run version.

Resetting the particles' positions – and in particular the *pbests* that lead the swarm – can increase the probability of convergence to the global optimum. However, relying solely on restarts does not provide an explicit mechanism to influence the exploration/exploitation balance. Thresheld convergence – by means of the threshold parameter – can control the amount of local search performed: large threshold values (in particular greater than the distance between local optima in adjacent basins) prevent the algorithm from doing any local search, while setting the threshold to zero allows PSO to converge freely. In the next section, we propose a strategy to identify and exploit the scale of the search space as the foundations for designing a PSO extension that combines thresheld convergence and multiple restarts.

4. IDENTIFYING AND EXPLOITING THE SCALE OF THE SEARCH SPACE

As illustrated in Section 3, in order to obtain better results with PSO on multi-modal problems, the ratio of global vs. local search needs to be increased. To fulfil this aim, we combine thresheld convergence and multiple restarts. Thresheld convergence allows us to have explicit control over the exploration/exploitation balance in PSO, while restarts introduce new solutions into the swarm. This section focuses on thresheld convergence, the main component around which the multi-start strategy is built.

Earlier algorithms using thresheld convergence have set the initial value of the threshold to a fraction of the main diagonal of the search space and have used a formula in terms of the total FEs (see e.g. [2, 6, 17]) or an adaptive approach (see e.g. [3, 7]) to reduce the threshold as the algorithm proceeds. A different method is proposed in the context of multi-starts: appropriate threshold values are selected automatically from an analysis of the search space and used to run different restarts of PSO with thresheld convergence, followed by a final local search step to exploit the discovered attraction basins. The goal is to make PSO search at different scales of the search space in order to focus only on a subset of the attraction basins at a given time.

The social component of the swarm update rule presented in Section 2.1 attracts connected particles towards the best positions in their neighbourhood. Due to the limited global communication of the (local) ring topology, it takes several communication cycles for this information to spread throughout the entire swarm. This situation favours the formation of groups of nearby solutions in the population of *pbests*, which constitute a promising set of solutions for the identification of threshold values. Accordingly, a method to get information about the scale of the search space expressed as the distance among neighbouring (groups of) *pbests* is required. We recur to cluster analysis and k-means in particular (see [12] for an introduction), which has been used successfully for similar purposes in the context of multi-swarm systems [13]. k-means splits the points into k groups that minimize the within-cluster sum of squared differences with respect to the centroid of the cluster, which is in concordance with the purpose of identifying groups of nearby solutions that are hopefully in different attraction basins.

Motivated by the behaviour of PSO observed in Rastrigin, the population of *pbests* after the first 10% of the total FEs is considered as the input set of solutions for k-means. Figure 4 illustrates the feasibility of this approach with a representative trial. The plot shows the average and overall minimum distance between the cluster centroids obtained with k-means for k between 2 and 50 (the swarm size). The horizontal lines correspond to the values 1 and $\sqrt{n}$ – i.e. the minimum and maximum distance between adjacent local optima, respectively.

A minimum cluster distance near zero indicates that k is so large that points which belong in the same cluster are forced to be in separate clusters. Thus, as k decreases, there should be a value at which the minimum cluster distance "spikes up" from zero. Across all fifty-one trials, there were visible spikes and these spikes often matched the known minimum distance between local optima of 1. Further, at the value of k with the spike, the average minimum distance for clusters was often close to the known maximum distance between adjacent local optima of $\sqrt{n}$.

The following procedure automates the identification of a value of k corresponding to these reference threshold values by targeting the observed spike. From $k = 50$ and decreasing iteratively towards $k = 2$, we start collecting the sample of the overall minimum distances. We select the first value of $k \leq 35$ in which the deviation of the overall minimum distance with respect to the mean of the collected sample differs by more than 10 standard deviations. For example, in Figure 4, the overall minimum distance for $k = 14$ is around 200 standard deviations away from the mean of the values for $k \in [15, 50]$. The condition is only applied for $k \leq 35$ in order to collect a reasonable sample of the overall minimum distance. If $k = 2$ is reached and the condition is never satisfied, the function is considered unimodal and the reference threshold values are both set to zero. This relatively simple heuristic performed consistently well on extensive experiments with different input sets of solutions.

Based on this heuristic, we propose the following multi-start PSO strategy with thresheld convergence. It is divided into three phases:

1. First, PSO is run for 10% of the allotted FEs. The final population of *pbests* is processed using the heuristic described above in order to obtain an estimation of appropriate threshold values for the search space. For further reference, let a and b denote the values of the average and overall minimum distance selected with the heuristic, respectively. This phase is responsible for the identification of the scale of the search space.
2. Next, the information collected during the first phase is exploited using thresheld convergence. Four restarts of PSO with thresheld convergence and different constant threshold values are run, each one for 20% of the total FEs. The thresholds for each restart are: a, $(a + b)/2$, $2b$ and b (a and b were defined in the description of the first phase). In the first restart of PSO with thresheld convergence, the information of the particle corresponding to the *gbest* of the initial phase is kept and the others are reset uniformly within the search space boundaries. In the other restarts only the threshold value changes.
3. Finally, standard PSO is run for the remaining 10% of the total FEs, starting from the *pbests* at the end of the fourth restart of PSO with thresheld convergence. The purpose of this phase is to perform local search which will find the (local) optima in the attraction basins discovered during the second phase.

Notice that at this point we have minimized the use of resets – i.e. particles are only reset after the first phase. Consequently, the use of thresheld convergence in phase two corresponds to a step threshold function. We have prioritized presenting a solid foundation to solve highly multi-modal problems based on an appropriate balance between global/local search. In the next section, a computational study of the proposed strategy on a wider range of benchmark problems is presented, and we then build an improved heuristic by incrementally adjusting phases two and three for better performance.

5. COMPUTATIONAL STUDY

This section presents an extensive computational study of the multi-start PSO with thresheld convergence introduced in the previous section. The test suite from the CEC 2013 Special Session on Real-Parameter Optimization [16] (CEC'13 benchmark for short) is considered for the evaluation of the performance of the heuristic in $n = 30$ dimensions. It includes 28 minimization problems with known global optimum, divided into three groups: unimodal functions (1–5), basic multi-modal functions (6–20), and composition functions (21–28). The second group is of particular interest to this study, given that the proposed technique was specially designed for such problems.

The experimental setup closely follows the guidelines of the CEC'13 benchmark. Fifty-one randomized trials with a maximum of 300,000 FEs were performed on each function, reporting the mean and standard deviation of the fitness errors achieved by the algorithms. In addition, the relative difference on the mean errors (%-diff) is given when two algorithms are compared. The %-diff of the mean error m_1 of a first algorithm with respect to the mean m_2 of a second algorithm is given by $100(m_1 - m_2)/\max(m_1, m_2)$. Hence, positive %-diff values indicate that the first algorithm performs better than the second one, and *vice versa*. A t-test between the two samples is also performed to focus the comparison on statistically significant differences at the 5% level.

5.1 Initial Results

Table 1 shows the results on the CEC'13 benchmark of standard PSO and the initial version of the multi-start PSO with thresheld convergence (as presented in Section 4). The proposed heuristic obtains average improvements over 5% on both the multi-modal test problems and the composite functions. On the multi-modal functions, nine results are significantly better, three are worse, and the remaining three are indistinguishable – which evidences that the clustering approach designed in terms of the observed behaviour on Rastrigin is also applicable to other multi-modal functions. In order to build a better-performing technique, the next section presents a first extension to the basic algorithm that resets the particles' positions and *pbests* during the second phase of the heuristic.

5.2 Resetting Positions and pbests

The multi-start technique described in Section 4 only resets particles during the transition between phases one and two – i.e. as part of the initialization of the first restart of PSO with thresheld convergence. In this section, resets are also introduced during the initialization of the second, third, and fourth restarts of PSO with thresheld convergence. However, to reuse earlier-obtained information about the search space implicitly encoded in the swarm, only partial resets are considered.

In the first restart of PSO with thresheld convergence, the *gbest* of the first phase is kept and the other particles are discarded, since it would not be profitable applying thresheld convergence to a swarm that has converged beyond the radius of the threshold size. Conversely, the other restarts offer more flexibility to reuse a greater number of particles because of the decreasing threshold values. Specifically, we propose keeping the positions and *pbests* of the $s > 1$ best particles from the previous restart, and completing the S particles in the initial swarm with reinitialized solutions.

Table 1: Results of PSO and the initial version of multi-start PSO with thresheld convergence (PSO+TC) – including only the step threshold function – on the CEC'13 benchmark. Bold %-diff values indicate statistical significance.

func.	PSO		Multi-Start PSO+TC (step threshold func.)		%-diff	t-test
	mean	std dev	mean	std dev		
1	0.00e+0	0.00e+0	2.81e−9	2.01e−8	-100.0%	0.32
2	2.07e+6	7.82e+5	8.00e+6	3.67e+6	**-78.3%**	0.00
3	6.75e+7	7.06e+7	9.15e+7	1.31e+8	-46.1%	0.29
4	1.80e+4	4.55e+3	2.70e+4	1.75e+4	**-47.0%**	0.00
5	0.00e+0	0.00e+0	7.44e−7	2.35e−6	**-100.0%**	0.02
1–5					-74.3%	
6	1.67e+1	4.34e+0	3.03e+1	1.44e+1	**-53.0%**	0.00
7	6.37e+1	1.74e+1	3.39e+1	9.54e+0	**50.0%**	0.00
8	2.09e+1	5.51e−2	2.09e+1	5.76e−2	0.0%	
9	2.86e+1	1.95e+0	2.39e+1	2.35e+0	**16.5%**	0.00
10	1.32e−1	5.53e−2	4.80e+0	2.72e+0	**-97.9%**	0.00
11	6.47e+1	1.51e+1	3.81e+1	9.90e+0	**43.7%**	0.00
12	7.85e+1	1.72e+1	6.52e+1	1.95e+1	**16.1%**	0.00
13	1.44e+2	2.21e+1	1.23e+2	2.63e+1	**14.%**	0.00
14	2.63e+3	3.80e+2	1.91e+3	4.66e+2	**28.3%**	0.00
15	3.95e+3	6.25e+2	4.59e+3	6.64e+2	**-14.6%**	0.00
16	1.66e+0	3.51e−1	1.58e+0	3.53e−1	3.8%	0.27
17	1.01e+2	1.55e+1	7.69e+1	1.16e+1	**25.2%**	0.00
18	1.70e+2	2.68e+1	1.57e+2	3.12e+1	**7.2%**	0.00
19	5.76e+0	1.30e+0	3.67e+0	7.82e−1	**39.0%**	0.00
20	1.17e+1	5.12e−1	1.17e+1	4.84e−1	0.0%	
6–20					5.27%	
21	2.20e+2	5.30e+1	2.30e+2	4.57e+1	-4.1%	0.31
22	2.96e+3	5.15e+2	1.97e+3	5.34e+2	**34.2%**	0.00
23	4.66e+3	7.06e+2	4.76e+3	7.97e+2	-2.2%	0.51
24	2.76e+2	5.64e+0	2.61e+2	7.88e+0	**5.3%**	0.00
25	2.92e+2	7.12e+0	2.77e+2	6.53e+0	**5.2%**	0.00
26	2.13e+2	4.61e+1	2.13e+2	4.30e+1	0.7%	0.92
27	1.03e+3	7.43e+1	8.99e+2	6.09e+1	**13.3%**	0.00
28	3.00e+2	0.00e+0	2.96e+2	2.80e+1	1.2%	0.32
21–28					6.7%	

Table 2: Results of multi-start PSO with thresheld convergence (PSO+TC) – including uniform resets – on the CEC'13 benchmark. The %-diff and t-test values are with respect to the PSO results in Table 1. Bold %-diff values indicate statistical significance.

func.	Multi-Start PSO+TC (uniform restarts)		%-diff	t-test
	mean	std dev		
1	0.00e+0	0.00e+0	0.0%	
2	6.81e+6	3.59e+6	**-75.5%**	0.00
3	1.51e+8	2.06e+8	**-73.0%**	0.01
4	2.40e+4	1.55e+4	**-39.4%**	0.01
5	1.52e−6	8.23e−6	-100.0%	0.19
1–5			-57.6%	
6	3.10e+1	1.50e+1	**-55.4%**	0.00
7	2.46e+1	7.21e+0	**64.2%**	0.00
8	2.09e+1	5.84e−2	0.0%	
9	2.04e+1	2.56e+0	**29.0%**	0.00
10	4.82e+0	3.69e+0	**-98.2%**	0.00
11	3.73e+1	1.19e+1	**44.0%**	0.00
12	5.05e+1	2.17e+1	**33.7%**	0.00
13	1.03e+2	2.79e+1	**28.6%**	0.00
14	1.56e+3	5.24e+2	**41.0%**	0.00
15	3.87e+3	8.57e+2	0.6%	0.57
16	1.34e+0	3.46e−1	**19.1%**	0.00
17	6.71e+1	9.62e+0	**34.9%**	0.00
18	1.22e+2	2.70e+1	**28.4%**	0.00
19	3.01e+0	7.13e−1	**49.7%**	0.00
20	1.12e+1	6.40e−1	**4.4%**	0.00
6–20			14.9%	
21	2.30e+2	5.71e+1	-4.8%	0.37
22	1.39e+3	4.37e+2	**54.4%**	0.00
23	3.92e+3	7.54e+2	**15.9%**	0.00
24	2.53e+2	7.93e+0	**8.3%**	0.00
25	2.68e+2	7.63e+0	**8.4%**	0.00
26	2.00e+2	2.12e−1	**10.2%**	0.04
27	7.85e+2	7.30e+1	**24.4%**	0.00
28	3.00e+2	1.23e−1	0.0%	
21–28			14.6%	

In order to be able to choose a small (yet effective) s value, the particles that are kept are then re-spaced equally among the S particles in the ring topology. This structure ensures that the information about the best solutions found so far is more accessible to all of the particles. Two reinitialization approaches for the $S - s$ particles are studied.

In the first approach, the particles are resampled uniformly within the bounds of the search space. The number of particles that are kept was set to 10, which showed the best performance among the tested values of $s \in \{2, 5, 10, 25\}$. Table 2 summarizes results of the proposed heuristic with uniform resets during the second phase, compared to the PSO baseline. The beneficial effects of the uniform resets on the overall behaviour of the algorithm are clear: the average performance of the heuristic on the three groups of problems was improved. In particular, the improvement on the targeted group of multi-modal functions is now close to 15%. The newly sampled solutions introduce diversity into the swarm, which translates into better exploration and less risk of stagnation.

The second reinitialization variant is motivated by the fact that the uniformly sampled solutions do not exploit information about promising search space regions already acquired by the algorithm. As it was illustrated in Section 3 using PSO with simple restarts, this can result in a considerable waste of FEs. To make use of this information implicitly encoded in the population of *pbests*, a sampling strategy inspired by the univariate marginal distribution algorithm (UMDA) for continuous domains [15] is used. Each component $j = 1, \ldots, n$ of a reinitialized solution is sampled from a univariate normal distribution $\mathcal{N}(\mu_j, c_\sigma \sigma_j^2)$, where μ_j and σ_j are respectively the mean and standard deviation of the sample of the variable j in the final population of *pbests* of the previous restart. The additional parameter c_σ is a factor that scales the variance of the sampling distribution, which was set empirically to 0.5 after experiments with the values of 0.5, 1.0, and 1.5.

Table 3 shows the results of the proposed technique including normal resets. It produces consistent improvements on the multi-modal problems reaching a mean performance improvement of around 21% in this group. Compared to the results using uniform resets, the effect on the other two groups is negligible. Therefore, the normal resets are selected over the uniform resets to be incorporated into the algorithm. The next section focuses on improving the local search step in the third phase of the heuristic.

5.3 Improving the Local Search Phase

A closer look at the results of the proposed strategy with normal resets presented in Table 3 shows that the two functions with negative results among the multi-modal functions share common characteristics in terms of the fitness landscape. Functions 6 (rotated Rosenbrock's) and 10 (rotated Griewank's) present a flat region that the algorithm must

Table 3: Results of multi-start PSO with thresheld convergence (PSO+TC) – including normal resets – on the CEC'13 benchmark. The %-diff and t-test values are with respect to the PSO results in Table 1. Bold %-diff values indicate statistical significance. Compare to the results in Table 2.

func.	Multi-Start PSO+TC (normal restarts) mean	std dev	%-diff	t-test
1	0.00e+0	0.00e+0	0.0%	
2	7.77e+6	3.89e+6	**-78.4%**	0.00
3	1.06e+8	2.28e+8	-70.9%	0.25
4	2.03e+4	1.49e+4	-30.0%	0.32
5	2.75e−7	9.96e−7	-100.0%	0.05
1–5			-55.9%	
6	2.90e+1	1.10e+1	**-47.9%**	0.00
7	2.19e+1	7.61e+0	**66.7%**	0.00
8	2.09e+1	6.24e−2	0.0%	
9	1.96e+1	3.07e+0	**31.9%**	0.00
10	3.35e+0	1.87e+0	**-96.9%**	0.00
11	2.63e+1	8.46e+0	**61.1%**	0.00
12	3.13e+1	2.42e+1	**58.2%**	0.00
13	6.87e+1	2.92e+1	**53.1%**	0.00
14	1.47e+3	3.66e+2	**45.0%**	0.00
15	3.55e+3	8.71e+2	**8.2%**	0.01
16	1.39e+0	4.64e−1	**14.6%**	0.00
17	6.02e+1	9.12e+0	**41.7%**	0.00
18	1.10e+2	3.24e+1	**35.7%**	0.00
19	2.85e+0	7.56e−1	**52.8%**	0.00
20	1.13e+1	6.75e−1	**3.7%**	0.00
6–20			21.8%	
21	2.30e+2	5.38e+1	-4.7%	0.27
22	1.61e+3	5.95e+2	**46.2%**	0.00
23	3.91e+3	9.38e+2	**15.4%**	0.00
24	2.52e+2	7.74e+0	**8.8%**	0.00
25	2.68e+2	6.62e+0	**8.3%**	0.00
26	2.06e+2	2.89e+1	5.6%	0.35
27	8.10e+2	6.79e+1	**22.0%**	0.00
28	2.96e+2	2.79e+1	1.2%	0.33
21–28			12.9%	

transverse during the optimization. Our hypothesis is that the algorithm becomes stalled on these regions because of insufficient local search to reach a local optimum.

In order to improve the local search capabilities of the third phase of the multi-start strategy, the swarm is reduced to the 10 best particles (i.e. the same number kept on restarts 2, 3, and 4 of PSO with thresheld convergence). This change both accelerates the communication among the particles and allows PSO to perform a greater number of generations within the same 10% of the allotted FEs. Additionally, the initial positions are set equal to the *pbests* and the initial velocities are set to the difference vectors between the corresponding *pbest* and the *gbest*. The goal is to ensure a quick convergence to the best local optimum in the attraction basins discovered during the previous phase.

Table 4 summarizes the results of the proposed multi-start PSO including the normal resets and the changes for better local search described above. The mean performance on the principal group of multi-modal functions has been raised to just over 24% and the results on both functions 6 and 10 are better. After having adjusted phases two and three of the proposed technique for better performance on the wide range of multi-modal problems included in the CEC'13 benchmark, 11 out of the 15 problems exhibit statistically significant performance improvements, and most of them constitute sizable differences of over 40%.

Table 4: Results of the final version of multi-start PSO with thresheld convergence (PSO+TC) – including normal resets and improved local search – on the CEC'13 benchmark. The %-diff and t-test values are with respect to the PSO results in Table 1. Bold %-diff values indicate statistical significance.

func.	Multi-Start PSO+TC (imp. local search) mean	std dev	%-diff	t-test
1	0.00e+0	0.00e+0	0.0%	
2	4.32e+6	3.53e+6	**-67.8%**	0.00
3	1.00e+8	2.66e+8	-81.2%	0.41
4	1.42e+4	1.04e+4	**6.2%**	0.02
5	0.00e+0	0.00e+0	0.0%	
1–5			-28.5%	
6	2.02e+1	6.03e+0	**-19.9%**	0.00
7	2.13e+1	6.27e+0	**68.3%**	0.00
8	2.09e+1	6.20e−2	0.0%	
9	1.95e+1	2.74e+0	**32.0%**	0.00
10	2.64e−1	2.17e−1	**-68.1%**	0.00
11	2.89e+1	1.39e+1	**56.7%**	0.00
12	3.80e+1	3.42e+1	**41.7%**	0.00
13	7.63e+1	3.90e+1	**46.8%**	0.00
14	1.54e+3	6.22e+2	**40.7%**	0.00
15	3.87e+3	1.23e+3	-1.3%	0.67
16	1.31e+0	4.50e−1	**18.5%**	0.00
17	6.14e+1	8.38e+0	**40.4%**	0.00
18	8.65e+1	4.64e+1	**47.3%**	0.00
19	2.82e+0	9.00e−1	**52.9%**	0.00
20	1.10e+1	6.78e−1	**5.6%**	0.00
6–20			24.1%	
21	2.25e+2	5.60e+1	-2.%	0.60
22	1.54e+3	7.18e+2	**48.7%**	0.00
23	4.29e+3	1.19e+3	5.6%	0.05
24	2.51e+2	9.18e+0	**9.1%**	0.00
25	2.69e+2	6.63e+0	**8.0%**	0.00
26	2.06e+2	2.79e+1	5.9%	0.32
27	8.00e+2	8.10e+1	**23.0%**	0.00
28	3.00e+2	0.00e+0	0.0%	
21–28			12.3%	

6. SUMMARY AND CONCLUSIONS

Thresheld convergence is a generally applicable mechanism for improving the performance of search heuristics on multi-modal problems. It is based on the hypothesis that simultaneous exploration and exploitation processes can interfere with a correct assessment of the potential fitness of a newly sampled region. Ideally, thresheld convergence will use a minimum distance threshold that is correlated to the size of the attraction basins in the search space. This paper examined the use of a clustering-based method in the context of a multi-start PSO to identify the scale of the search space which thresheld convergence can then exploit.

As a conclusive note on the effect of the proposed multi-start PSO algorithm with thresheld convergence, we show the performance of the final version of the algorithm (discussed in Section 5.3) on Rastrigin. Figure 5 clearly illustrates the effect of each component of the strategy and the important differences with respect to the behaviour of standard PSO shown in Figure 2. Note the effect of thresheld convergence in delaying convergence (i.e. the gap between the two lines) and to effectively increase global search in a controlled manner (i.e. the continuous decrease of the solid line and the dips that occur when the threshold values change, e.g. at 150,000 FEs). The effect of the local search step is also illustrated after 270,000 FEs when the two lines

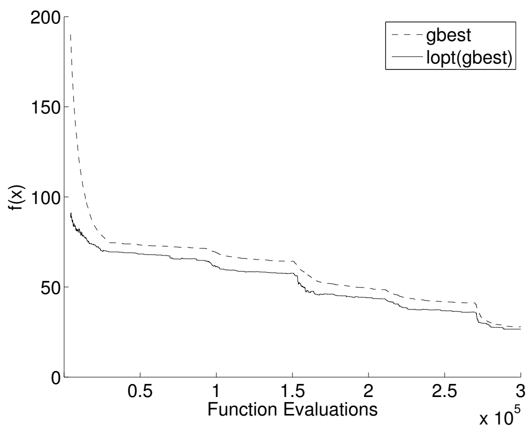

Figure 5: Results of the final version of multi-start PSO with thresheld convergence – including normal resets and improved local search – on Rastrigin for $n = 30$. The plot shows the average performance over 51 trials: mean $f(x)$ of *gbest* is 27.9.

converge quickly. Overall, the proposed strategy obtains a 56.2% performance improvement on this function.

The proposed technique not only outperformed standard PSO on Rastrigin, but it also led to large improvements across a broad range of multi-modal optimization problems. The reported experimental study validated the robustness of the proposed clustering approach to identify appropriate threshold values. Interestingly, the use of a similar approach on differential evolution (DE) with thresheld convergence led to divergent results [18]. The attraction vectors in PSO promote the formation of distinct clusters, whereas DE exhibits a more homogeneous convergence which does not produce the clusters which the proposed cluster-based method is designed to identify. These results constitute key insights for future research on thresheld convergence and its integration with different heuristics.

7. REFERENCES

[1] S. Bird and X. Li. Adaptively choosing niching parameters in a PSO. In *8th Annual Conference on Genetic and Evolutionary Computation*, 3–10, 2006.

[2] A. Bolufé-Röhler and S. Chen. Minimum population search – lessons from building a heuristic technique with two population members. In *IEEE Congress on Evolutionary Computation*, 2061–2068, 2013.

[3] A. Bolufé-Röhler, S. Estévez-Velarde, A. Piad-Morffis, S. Chen, and J. Montgomery. Differential evolution with thresheld convergence. In *IEEE Congress on Evolutionary Computation*, 40–47, 2013.

[4] D. Bratton and J. Kennedy. Defining a standard for particle swarm optimization. In *IEEE Swarm Intelligence Symposium*, 120–127, 2007.

[5] R. Brits, A. P. Engelbrecht, and F. van den Bergh. A niching particle swarm optimizer. In *4th Asia-Pacific Conference on Simulated Evolution and Learning*, 692–696, 2002.

[6] S. Chen and J. Montgomery. A simple strategy to maintain diversity and reduce crowding in particle swarm optimization. In *24th Australian Joint Conference on Artificial Intelligence*, 281–290, 2011.

[7] S. Chen and J. Montgomery. Particle swarm optimization with thresheld convergence. In *IEEE Congress on Evolutionary Computation*, 510–516, 2013.

[8] 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.

[9] A. Engelbrecht. Particle swarm optimization: Velocity initialization. In *IEEE Congress on Evolutionary Computation*, 70–77, 2012.

[10] S. Helwig, J. Branke, and S. M. Mostaghim. Experimental analysis of bound handling techniques in particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 17(2):259–271, 2013.

[11] T. Hendtlass. WoSP: A multi-optima particle swarm algorithm. In *IEEE Congress on Evolutionary Computation*, 727–734, 2005.

[12] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis.* Wiley, 1990.

[13] J. Kennedy. Stereotyping: Improving particle swarm performance with cluster analysis. In *IEEE Congress on Evolutionary Computation*, 1507–1512, 2000.

[14] J. Kennedy and R. Eberhart. Particle swarm optimization. In *IEEE International Conference on Neural Networks*, 1942–1948, 1995.

[15] P. Larrañaga, R. Etxeberria, J. A. Lozano, and J. M. Peña. Optimization by learning and simulation of Bayesian and Gaussian networks. Technical Report EHU-KZAA-IK-4/99, University of the Basque Country, Spain, 1999.

[16] J. J. Liang, B. Y. Qu, P. N. Suganthan, and A. G. Hernández-Díaz. Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. Technical Report 201212, Zhengzhou University, China and Nanyang Technological University, Singapore, 2013.

[17] J. Montgomery and S. Chen. A simple strategy for maintaining diversity and reducing crowding in differential evolution. In *IEEE Congress on Evolutionary Computation*, 2692–2699, 2012.

[18] J. Montgomery, S. Chen, and Y. Gonzalez-Fernandez. Identifying and exploiting the scale of a search space in differential evolution. In *IEEE Congress on Evolutionary Computation*, 2014. In press.

[19] H. Mühlenbein, M. Schomisch, and J. Born. The parallel genetic algorithm as function optimizer. *Parallel Computing*, 17(6–7):619–632, 1991.

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

[21] M. Črepinšek, S.-H. Liu, and M. Mernik. Exploration and exploitation in evolutionary algorithms: A survey. *ACM Computing Surveys*, 45(3):35:1–35:33, 2013.

[22] S. Z. Zhao, J. J. Liang, P. N. Suganthan, and M. F. Tasgetiren. Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization. In *IEEE Congress on Evolutionary Computation*, 3845–3852, 2008.