

A Simple Strategy to Maintain Diversity and Reduce Crowding in Particle Swarm Optimization

Stephen Chen¹ and James Montgomery²

¹ School of Information Technology, York University
4700 Keele Street, Toronto, Ontario M3J 1P3
syichen@yorku.ca

² College of Engineering and Computer Science, Australian National University
Canberra, ACT 0200, Australia
james.montgomery@anu.edu.au

Abstract. Each particle in a swarm maintains its current position and its personal best position. It is useful to think of these personal best positions as a population of attractors – updates to current positions are based on attractions to these personal best positions. If the population of attractors has high diversity, it will encourage a broad exploration of the search space with particles being drawn in many different directions. However, the population of attractors can converge quickly – attractors can draw other particles towards them, and these particles can update their own personal bests to be near the first attractor. This convergence of attractors can be reduced by having a particle update the attractor it has approached rather than its own attractor/personal best. This simple change to the update procedure in particle swarm optimization incurs minimal computational cost, and it can lead to large performance improvements in multi-modal search spaces.

Keywords: Particle swarm optimization, crowding, niching, population diversity, multi-modal search spaces

1 Introduction

The development of particle swarm optimization (PSO) includes inspirations from “bird flocking, fish schooling, and swarming theory in particular” [11]. Each particle (e.g. a simulated bird) is attracted to its personal best position and the best position of a neighbouring member of the swarm. In original PSO [11], the neighbourhood for all particles is the entire swarm (i.e. a star topology) – the global best position attracts all of the other particles towards it. This concentration of search around a single attractor can work well in unimodal search spaces, but this level of convergence can also lead to poor performance in multi-modal search spaces.

To improve the balance between exploration and exploitation, standard PSO [1] recommends a ring topology – each particle communicates with only two neighbours. With this reduced communication, a single good position will not immediately attract all of the other particles in the swarm. Specifically, several different positions can each act as the attractor for a small subset of particles, and the overall swarm can

subsequently explore many regions of the search space. This increased exploration generally improves PSO performance in multi-modal search spaces [1].

The use of a ring topology can lead to local behaviours that are similar to sub-swarms. For example, if particle 2 is the attractor for particles 1 and 3, and particle 5 is the attractor for particles 4 and 6, then this six particle swarm could temporarily behave like two independent swarms of three particles each. Many multi-swarm techniques exist which use sub-swarms (in sequence or in parallel) to explore multiple local optima (e.g. [3][10][12]). Compared to standard PSO, these multi-swarm techniques tend to have their most consistent performance improvements in multi-modal search spaces (e.g. as shown for locust swarms in [4]).

In population search techniques, another way to explore multiple local optima is niching (e.g. [2]). The effect of niching is to cause the overall population to divide into several sub-populations that each explores the area around a distinct local optimum. The intention of niching can be to simultaneously explore multiple local optima with the goal of finding many or all of the local optima in a search space.

A related idea that ultimately allows the population to converge is to reduce crowding (e.g. [6][13]). Crowding occurs when two or more population members are too close to each other. As crowds gather, population diversity is reduced and the explorative capacity of the search technique is similarly reduced. To prevent crowds, a new candidate solution should replace a similar solution in the population. This replacement strategy ensures that these two solutions will not be able to form a crowd.

If the personal best positions are viewed as a population of attractors, it can be seen that the basic operation of PSO promotes crowding. An attractor draws another particle towards it with the explicit purpose of having that other particle search in the nearby area. If the attracted particle subsequently finds a new personal best position near this local best attractor, it will update its own personal best attractor to be near the first attractor – these two attractors have now formed a crowd.

To reduce crowding, the standard procedure is to compare the new candidate solution with several existing members of the population. Among these solutions, the minimal loss of diversity occurs if the most similar solution is replaced [6]. Transferring this idea to PSO, a new personal best position should be compared with the nearest/most similar member in the population of personal best attractors. Subsequently, a new update strategy is proposed which allows particles to update the personal bests of other particles.

A modified PSO that implements the above strategy to reduce crowding has been developed. Starting with standard PSO [1] (and its ring topology), the procedure to update personal best (*pbest*) positions is changed to first check if the new position is close to its local best (*lbest*) attractor. If the new position is within a threshold distance to its *lbest* attractor, it is compared with and potentially updates this attractor. Outside of the threshold distance, normal PSO comparisons and updates occur. The effectiveness of this strategy is tested across a broad range of benchmark functions.

The proposed new strategy to maintain diversity in particle swarm optimization draws inspiration from crowding techniques which are reviewed in Section 2. A brief introduction to PSO is given in Section 3 before the details of the new update strategy are presented in Section 4. Experiments on a broad range of standard benchmark problems are performed in Sections 5 and 6. The results of these experiments are discussed in Section 7 before the paper is summarized in Section 8.

2 Background

The balance between exploration and exploitation is a recurring theme in many heuristic search techniques. For example, selection pressure in genetic algorithms will increase the proportion of “fit” schemata in a population [8]. However, the unselected schemata can be eliminated from the population which will lead to decreased diversity. In general, maintaining diversity will reduce the rate of convergence which will reduce the likelihood of stagnation in a poor local optimum, and this ability to continue progress can ultimately lead to the discovery of a better final solution. However, slower convergence also tends to increase the time required by the search process to produce a final result, so the balance between exploration and exploitation is important for both the efficient and effective performance of many heuristic search techniques.

One method to maintain diversity in a population is to reduce crowding. The basic technique is to compare each new candidate solution with its most similar individual in a subset of the overall population. The fitter of these two solutions survives as a member of the population. The size of the subset to find a neighbour for comparison can be small [6], which can cause “replacement errors”, or it can be large, which can cause significant increases to the required computational effort [13].

The basic crowding technique always replaces the most similar individual in the examined subset, but there is no guarantee (especially at early stages of the search process) that it would not be beneficial to have both of these solutions survive. This effect is related to a replacement error – one effect of a replacement error is that an unexamined solution is more similar and that its survival allows a crowd to form, and another effect of a replacement error is that a relatively diverse and potentially useful solution is unnecessarily removed from the population. This second effect can occur even if the crowd size is the entire population.

These two effects highlight the key objectives of crowd control: maintain a diverse set of promising solutions and reduce (premature) convergence. Similar goals are useful for the population of personal best attractors in particle swarm optimization. Specifically, a particle with crowded personal best and local best attractors will be drawn/constrained to this small region of the search space. Since this particle is not immediately affected by the position of other attractors, they can be (temporarily) ignored. Thus, there are only two attractors of concern, and the new strategy becomes similar to crowding with a subset of size two: the personal best and the local best for each particle.

3 Particle Swarm Optimization

The benchmark and baseline PSO for the current experiments is a constricted LBest version (i.e. standard PSO [1]) developed from the published source code for the constricted GBest version by El-Abd and Kamel [7]. In a constricted PSO, each dimension d of a particle’s velocity v is updated for the next iteration $i+1$ by

$$v_{i+1,d} = \chi(v_{i,d} + c_1\varepsilon_1(pbest_{i,d} - x_{i,d}) + c_2\varepsilon_2(lbest_{i,d} - x_{i,d})) \quad (1)$$

where χ is the constriction factor, c_1 and c_2 are weights which vary the contributions of personal best and local best attractors, ε_1 and ε_2 are independent uniform random numbers in the range of $[0,1]$, x is the position of the particle, $pbest$ is the best position found by the current particle, and $lbest$ is the best position found by any particle communicating with the current particle (e.g. all particles in the GBest star topology and only two neighbours in an LBest ring topology). The key parameters used in [7] are $\chi = 0.792$, $\chi * c_1 = \chi * c_2 = 1.4944$, i.e. $c_1 = c_2 \approx 1.887$, and $p = 40$ particles.

The following experiments use a fixed number of function evaluations (FE) based on the number of dimensions D . The chosen limit of $FE = 5000 * D$ promotes consistency with previous results. In particular, results for the original GBest version of this benchmark PSO are reported in [5][7], and results for the constricted LBest version are available in [4].

4 A New Update Strategy for PSO

In PSO, the update of a particle's velocity shown in (1) is based on three distinct components: a momentum term (m), an attraction to $pbest$ (f_p), and an attraction to $lbest$ (f_l). An example of how these three component vectors might combine to create the new velocity v_i at iteration i is shown in Fig. 1. Applying this new velocity to the previous position leads to a new position x_i .

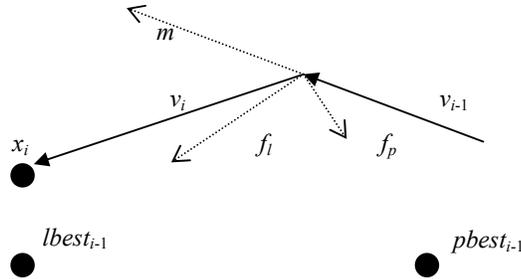


Fig. 1. A particle's path is influenced by attractions to $pbest$ and $lbest$ positions. In this example, the new particle position has been drawn close to its $lbest$ attractor.

After determining the new position, the fitness is calculated and the personal best position is updated if necessary.

Pseudo code for the standard update procedure used in PSO

```

if  $f(x_i) < f(pbest_{i-1})$  then
     $pbest_i = x_i$ 

```

Starting from the example in Fig. 1, assume that $f(x_i) < f(pbest_{i-1})$. The standard update procedure will then make x_i the new position for $pbest_i$, and this will cause the

two attractors to become very close (i.e. form a crowd). During the next iteration after the standard update shown in Fig. 2, the closeness of the attractors $pbest_i$ and $lbest_i$ will help to constrain the future search path of this particle to a small area of the search space around these two points. Low diversity in the population of attractors leads to reduced explorative behaviour in the flight paths of a swarm's particles.

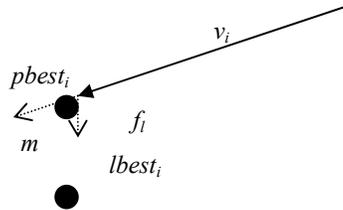


Fig. 2. If $f(x_i) < f(pbest_{i-1})$ in the example from Fig. 1, then the standard update procedure will update $pbest$ to be next to $lbest$.

A small number of converged particles might not be too damaging, but the convergence of attractors can have a cascading effect (even with a ring topology). For example, assume that the $pbest$ for particle 1 is the $lbest$ attractor for particle 2. After an update like the one shown in Fig. 2, it is possible that the new $pbest$ for particle 2 can become the $lbest$ attractor for particle 3. This third particle will now be drawn towards this area with a high concentration of $pbest$ attractors. If it also finds a new $pbest$ in this area, then this cascade of convergence in the population of $pbest$ attractors can continue until all particles have been drawn into this area.

Focusing on this population of $pbest$ attractors, the key concept from crowding is that a new solution should replace the most similar member in the existing population. Therefore, instead of replacing $pbest$ in Fig. 2, the new position x should replace $lbest$.

Pseudo code for the new update strategy

```

if  $||x_i - lbest_{i-1}|| < \text{threshold}$  then
  if  $f(x_i) < f(lbest_{i-1})$ 
     $lbest_i = x_i$ 
  else if  $f(x_i) < f(pbest_{i-1})$  then
     $pbest_i = x_i$ 

```

In crowding [6], a “crowding factor” specifies the size of the (randomly selected) subset from the overall population which can undergo replacement. The new solution is compared to the members of this subset, and it replaces the most similar solution (if the new solution is fitter). The new update strategy is similar to crowding with a crowding factor of two. However, these two solutions are not selected randomly – they are the $pbest$ and the $lbest$ for the current particle. Further, the closer of these two points is not automatically replaced. The new update strategy also uses a threshold function to control the minimum required diversity. As discussed in Section 2, the most similar solution in a population can still represent a useful area for further exploration.

If the new update strategy is applied to the example in Fig. 1, it will prevent the creation of a crowd between x_i (which becomes $pbest_i$ in Fig. 2) and $lbest_i$. Instead x_i will replace $lbest_{i-1}$ (see Fig. 3). With this update of $lbest$, the new strategy separates the two roles of $pbest$: store the best known position and act as an attractor in the search space. The swarm as a whole still remembers the best known position (which is stored in $lbest$), but greater diversity is maintained in the population of $pbest$ attractors. The effect of reduced crowding is to maintain diversity in the attractors and subsequently to encourage a greater exploration of the overall search space.¹

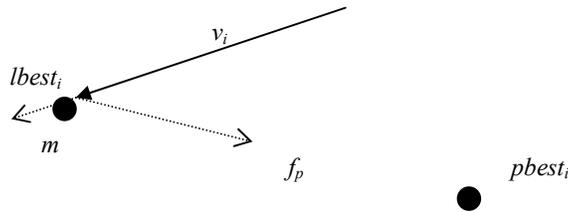


Fig. 3. Compared to the standard update procedure, the new update strategy will update $lbest$ instead. This will help maintain diversity in the $pbest$ and $lbest$ attractors.

5 Results on Multi-Modal Functions

The value of increased diversity is to lessen the risk of premature convergence to a poor local optimum. Since local optima do not exist on unimodal functions, the new strategy is not expected to provide benefits on these functions. The following experiments compare the performance of standard PSO [1] based on the benchmark implementation of [7] with a modified version which replaces the “standard update procedure” with the “new update strategy”.

The functions (with their ranges) for the following experiments are Fletcher-Powell $[-\pi, \pi]$, Langerman (with $m = 7$) $[0, 10]$, Rastrigin $[-5.12, 5.12]$, Schwefel $[-500, 500]$, and Shubert $[-10, 10]$, and all functions are in $D = 20$ dimensions. The details for the benchmark PSO are available in the published source of [7], and the key features and parameters are repeated in Section 3. Preliminary experiments with this modified PSO determined that a “threshold” parameter was required to properly calibrate the new balance between exploration and exploitation.

¹ If the distances between the new position x and the previous $pbest$ and $lbest$ positions are both less than the threshold, these two distances should both be measured to ensure that x replaces its nearest attractor. Without the extra distance calculation, approximately 1% of the updates under the new strategy can replace a more distant $lbest$ attractor. However, since $pbest$ and $lbest$ must already be quite close for this event to occur, it is not expected to have a large effect on the overall performance.

The parameter tuning experiments revealed that the *threshold* should decay over time (to allow the swarm to converge), and that the threshold should only be applied to *lbest*. If the new update strategy is applied to all attractors/*pbests* in the population, the computational effort is much larger and the performance is much worse – a result presumably caused by a complete lack of convergence. The *threshold* used in the following experiments starts with an initial value of 10% of the search space diagonal (i.e. $\alpha = 0.10$), and it decays with a cubic function (i.e. $\gamma = 3$) – in (2), n is the total number of iterations and i is the current iteration.

$$threshold = (\alpha * diagonal) * ((n - i) / n)^\gamma \quad (2)$$

The experiments involve 50 independent runs started with different random seeds. The final solution from each technique is collected after 100,000 function evaluations (i.e. $5,000 * D$). For these 50 runs, the minimum (min), mean, maximum (max), and standard deviation (std dev) are presented in Table 1. Except for the maximum and the standard deviation on Rastrigin, PSO with the modified update strategy (Mod) leads to better results (or same for minimum on Langerman) when compared to standard PSO (Std). The p-value for a one-tailed *t*-tests show that the differences in performance have some variability – since the p-values are not all much less than 5%, these results represent more of a promising trend than a strongly significant result.

Table 1. Results for the new update strategy on several benchmark multi-modal functions

Function	PSO	min	mean	max	std dev	t-test
Fletcher-Powell	Std	1,245	10,460	34,181	8,173	1.8%
	Mod	997	7,258	18,828	4,387	
Langerman	Std	-0.513	-0.399	-0.100	0.118	5.2%
m = 7	Mod	-0.513	-0.440	-0.272	0.086	
Rastrigin	Std	7.96	28.77	46.76	8.01	3.0%
	Mod	7.39	23.75	47.96	8.23	
Schwefel	Std	890	1,605	2,360	347	0.0%
	Mod	594	1,139	1,780	272	
Shubert	Std	-3.14e+22	-3.77e+21	-6.34e+19	5.16e+21	5.0%
	Mod	-5.75e+22	-7.34e+21	-2.50e+20	1.15e+21	

6 Results on Other Functions

The modified update strategy is designed explicitly for multi-modal search spaces, but it is still useful to observe its effects across a board range of search spaces. The following experiments use the Black-Box Optimization Benchmarking (BBOB) functions [9]. The BBOB problems are broken into five sets – (1) separable functions, (2) functions with low or moderate conditioning, (3) unimodal functions with high conditioning, (4) multi-modal functions with adequate global structure, and (5) multi-modal functions with weak global structure.

The results for standard PSO on the BBOB functions with dimension $D = 20$ are taken from previous work by the authors [4]. These results (means and standard deviations) are for 25 independent trials of 100,000 function evaluations each (i.e. 5

trials on each of the first 5 instances of each BBOB function). On these functions, standard PSO is able to get within $1e-8$ of the optimal solution on every trial of BBOB fn 1, 2, and 5. Errors of this size are considered negligible on the BBOB, so these functions are considered as fully solved. The following experiments only consider the remaining 21 BBOB functions which cannot be solved by standard PSO.

Several sets of parameters for the modified PSO were tried. Preliminary experiments determined that values of 1 and 4 for γ never led to the best-overall results. Thus, the results in Table 2 represent the best performance by the modified PSO across a total of eight parameter pairs – 0.01, 0.04, 0.10, or 0.33 for α and 2 or 3 for γ . For the best set of parameters as shown, the mean errors from optimum (mean), standard deviations (std dev), percent improvement in the mean for the results of modified PSO compared to the results of standard PSO (%-diff), and the p-value for a one-tailed *t*-test are reported.

Table 2. Results for the new update strategy on Black-Box Optimization Benchmarking functions

Set	fn	Standard PSO		Modified PSO		Parameters		% -diff	<i>t</i> -test
		mean	std dev	mean	std dev	α	γ		
1	1	0.00e+0	0.00e+0						
	2	0.00e+0	0.00e+0						
	3	2.56e+1	4.99e+0	2.18e+1	6.17e+0	0.10	3	14.9%	1.0%
	4	3.23e+1	8.55e+0	2.80e+1	6.492e+0	0.04	3	13.4%	2.5%
	5	0.00e+0	0.00e+0						
2	6	8.53e-1	8.89e-1	7.77e-1	5.20e-1	0.10	3	8.9%	35.8%
	7	7.04e+0	2.68e+0	5.40e+0	2.27e+0	0.04	2	23.4%	1.2%
	8	1.22e+1	3.67e+0	1.07e+1	5.00e+0	0.01	3	11.8%	12.6%
	9	1.55e+1	2.24e+0	1.51e+1	2.95e+0	0.01	3	3.0%	26.7%
3	10	6.85e+3	3.39e+3	8.54e+3	3.36e+3	0.01	3	-24.6%	4.2%
	11	6.54e+1	1.71e+1	5.72e+1	1.50e+1	0.01	3	12.6%	3.8%
	12	1.53e+0	4.23e+0	7.38e-1	9.04e-1	0.01	3	51.7%	18.5%
	13	1.50e+0	1.99e+0	1.09e+0	6.16e-1	0.04	3	27.4%	16.6%
	14	1.34e-3	2.66e-4	2.28e-3	4.84e-4	0.01	3	-70.7%	0.0%
	15	6.05e+1	1.46e+1	4.89e+1	1.37e+1	0.01	3	19.2%	0.3%
4	16	5.37e+0	1.53e+0	4.42e+0	1.21e+0	0.01	3	17.6%	1.0%
	17	6.61e-1	2.64e-1	4.30e-1	1.49e-1	0.04	2	34.9%	0.0%
	18	2.87e+0	1.28e+0	2.33e+0	8.00e-1	0.10	3	18.9%	4.0%
	19	3.61e+0	4.32e-1	3.50e+0	5.11e-1	0.01	2	3.1%	20.7%
5	20	1.14e+0	1.38e-1	9.07e-1	1.46e-1	0.01	2	20.1%	0.0%
	21	1.41e+0	1.21e+0	5.68e-1	7.70e-1	0.33	2	59.8%	0.3%
	22	1.69e+0	1.51e+0	1.05e+0	6.44e-1	0.33	3	38.1%	2.9%
	23	1.33e+0	2.49e-1	1.26e+0	3.02e-1	0.01	2	5.2%	19.3%
	24	1.13e+2	1.12e+1	1.10e+2	1.54e+1	0.01	2	2.6%	22.5%

Although only the best results are shown, it is worth mentioning that the unreported results for non-multi-modal functions are highly inconsistent. On many of the functions, the modified PSO was able to produce an improvement for only the reported parameter set, and none of the eight parameter sets led to an improvement on BBOB fn 10 and 14. Conversely, the modified PSO performed much more

consistently on the multi-modal functions for which it was designed. On these functions, the bold values represent statistically significant improvements of more than 10%. Further, there is some robustness to these results as each of these functions had at least one additional parameter set that also led to an improvement of more than 10%. From these observations, it is hypothesized that matching the α parameter to the spacing of the local optima in the search space will lead to the best performance for the proposed strategy, and that the best value for the γ parameter may depend on the contour of the fitness landscape around each local optimum.

7 Discussion

Standard particle swarm optimization shows broad improvements over original PSO across a diverse range of problems [1], but it is still only a starting point for the design of a practical application. In accordance with “no free lunch” [14], there is no single set of parameters that can be expected to lead to the best possible performance of a technique on multiple problems. Therefore, parameter tuning and other modifications are a necessary part of achieving the best possible results for any specific application of a heuristic search technique. Given the large performance improvements that can be achieved with the new strategy, the addition of a new threshold function should not be unduly cumbersome.

The proposed modification to the update strategy in PSO is generally ineffective outside of the targeted multi-modal functions. This is not a major concern since multi-modal functions are the primary application for heuristic search techniques like PSO – gradient descent methods tend to be much more effective than heuristic search techniques on unimodal functions (e.g. BBOB set 3). The underlying mechanisms of the new update strategy attempt to maintain diversity by reducing crowding, and the value of this increased diversity is primarily realized in multi-modal search spaces where it can help prevent premature convergence to a poor local optimum.

The new modification is also simple and computationally efficient. To change from the “standard update procedure” to the “new update strategy”, only a distance calculation between two specific points is required – the position of a particle and the position of its *lbest* attractor. In comparison, other diversification strategies based on niching and crowding are either computationally expensive (as distances between a new solution and all existing population members must be calculated) or prone to “replacement errors” (if only a subset of the population is compared against) [13]. In PSO, it is possible to identify the most likely population member that a new candidate solution might form a crowd with – its *lbest* attractor. This insight allows the proposed modification to achieve many of the benefits of niching and crowding at a fraction of the computational cost.

8 Summary

Particle swarm optimization must find the proper balance between exploration and exploitation to maximize its performance. The proposed modification to improve

exploration by maintaining diversity is simple and computationally efficient. The reduction in crowding achieved by the new update strategy leads to significant performance improvements in the targeted multi-modal search spaces. The key insight in the current research is the ability to identify with which existing population member a new solution might form a crowd. Future work will attempt to apply this insight to other population search techniques.

References

1. Bratton, D., Kennedy, J.: Defining a Standard for Particle Swarm Optimization. In: Proceedings of the 2007 IEEE Swarm Intelligence Symposium, pp. 120–127. IEEE Press, Los Alamitos (2007)
2. Brits, R., Engelbrecht, A.P., Van den Bergh, F.: A Niching Particle Swarm Optimizer. In: Proceedings of the 4th Asia-Pacific conference on Simulated Evolution and Learning, pp. 692–696. IEEE Press, Los Alamitos (2002)
3. Chen, S.: Locust Swarms – A New Multi-Optima Search Technique. In: Proceedings of the 2009 IEEE Congress on Evolutionary Computation, pp. 1745–1752. IEEE Press, Los Alamitos (2009)
4. Chen, S., Montgomery, J.: Selection Strategies for Initial Positions and Initial Velocities in Multi-optima Particle Swarms. In: Proceedings of the 2011 Genetic and Evolutionary Computation Conference, pp. 53–60. ACM, New York (2011)
5. Chen, S., Noa Vargas, Y.: Improving the Performance of Particle Swarms through Dimension Reductions – A Case Study with Locust Swarms. In: Proceedings of the 2010 IEEE Congress on Evolutionary Computation, pp. 2950–2957. IEEE Press, Los Alamitos (2010)
6. De Jong, K.A.: An Analysis of the Behavior of a Class of Genetic Adaptive Systems, PhD thesis. Dept. of Computer and Communication Sciences, University of Michigan. (1975)
7. El-Abd, M., Kamel, M.S.: Black-Box Optimization Benchmarking for Noiseless Function Testbed using Particle Swarm Optimization. In: Proceedings of the 2009 Genetic and Evolutionary Computation Conference, pp 2269–2273. ACM, New York (2009)
8. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, Reading (1989)
9. Hansen, N., Finck, S., Ros, R., Auger, A.: Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions. INRIA Technical Report RR-6829 (2009)
10. Hendtlass, T.: WoSP: A Multi-Optima Particle Swarm Algorithm. In: Proceedings of the 2005 IEEE Congress on Evolutionary Computation, pp. 727–734. IEEE Press, Los Alamitos (2005)
11. Kennedy J., Eberhart R.C.: Particle Swarm Optimization. In: Proceedings of the IEEE International Conference on Neural Networks, Vol. 4, pp. 1942–1948. IEEE Press, Los Alamitos (1995)
12. Liang, J.J., Suganthan, P.N.: Dynamic Multi-Swarm Particle Swarm Optimizer. In: Proceedings of the 2005 IEEE Swarm Intelligence Symposium, pp. 124–129. IEEE Press, Los Alamitos (2005)
13. Thomsen, R.: Multimodal Optimization using Crowding-based Differential Evolution. In: Proceedings of the 2004 IEEE Congress on Evolutionary Computation, pp. 1382–1389. IEEE Press, Los Alamitos (2004)
14. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, 1: 67–82 (1997)