Adaptively Choosing Neighbourhood Bests Using Species in a Particle Swarm Optimizer for Multimodal Function Optimization

Xiaodong Li

School of Computer Science and Information Technology
RMIT University, VIC 3001, Melbourne, Australia
xiaodong@cs.rmit.edu.au
http://www.cs.rmit.edu.au/~xiaodong

Abstract. This paper proposes an improved particle swarm optimizer using the notion of species to determine its neighbourhood best values, for solving multimodal optimization problems. In the proposed species-based PSO (SPSO), the swarm population is divided into species subpopulations based on their similarity. Each species is grouped around a dominating particle called the species seed. At each iteration step, species seeds are identified from the entire population and then adopted as neighbourhood bests for these individual species groups separately. Species are formed adaptively at each step based on the feedback obtained from the multimodal fitness landscape. Over successive iterations, species are able to simultaneously optimize towards multiple optima, regardless of if they are global or local optima. Our experiments demonstrated that SPSO is very effective in dealing with multimodal optimization functions with lower dimensions.

1 Introduction

In recent years, Particle Swarm Optimization has been used increasingly as an effective technique for solving complex and difficult optimization problems [3, 6,7]. However, most of these problems handled by PSOs are often treated as a task of finding a single global optimum. In the initial PSO proposed by Eberhart and Kennedy [7], each particle in a swarm population adjusts its position in the search space based on the best position it has found so far, and the position of the known best-fit particle in the entire population (or neighbourhood). The principle behind PSO is to use these particles with best known positions to guide the swarm population to converge to a single optimum in the search space.

How to choose the best-fit particle to guide each particle in the swarm population is a critical issue. This becomes even more acute when the problem being dealt with has multiple optima, as the entire swarm population can be potentially misled to local optima. One approach to combat this problem is to allow the population to search for multiple optima (either global or local) simultaneously. Striving to locate multiple optima has two advantages. Firstly, by locating multiple optima, the likelihood of finding the global optimum is increased; secondly,

K. Deb et al. (Eds.): GECCO 2004, LNCS 3102, pp. 105–116, 2004.

when dealing with real-world problems, for some practical reasons, it is often desirable for the designer to choose from a diverse set of good solutions, which may be equally good global optima or even second best optima.

The uniqueness of PSO's ability in adaptively adjusting particles' positions based on the dynamic interactions with other particles in the population makes it well suited for handling multimodal optimization problems. If suitable particles can be determined as the appropriate neighbourhood best particles to guide different portions of the swarm population moving towards different optima, then essentially we will be able to use a PSO to optimize over a multimodal fitness landscape. Ideally multiple optima will be found. Now the question is how to determine which particles would be suitable as neighbourhood bests; and how to assign them to the suitable particles in the population so that they will move towards different optima accordingly.

The paper is organized as follows: section 2 describes related work on multimodal optimization, and their relevance to the proposed species-based PSO (SPSO). Section 3 presents the classic PSO. Section 4 introduces the notion of species and its relation to multimodal optimization. Section 5 describes the proposed SPSO. Section 6 and 7 cover the performance measures and test functions respectively, followed by section 8 on experimental setup and then section 9 on results and discussion. Finally section 10 draws some conclusions and gives directions for future research.

2 Related Work

Although multimodal function optimization has been studied extensively by EA researchers, only few works have been done using particle swarm models. In [5], Kennedy proposed a PSO using a k-means clustering algorithm to identify the centers of different clusters of particles in the population, and then these cluster centers are used to substitute the personal bests or neighbourhood bests. However some serious limitations of this method can be identified:

1. In order to calculate the cluster centers, the method requires three iterations over all individuals in the population, which is very computationally expensive.
2. A cluster center identified is not necessarily the best-fit particle in that cluster. Consequently using these cluster centers as *lbest* is likely to lead to poor performance (see Fig. 1 of [5]).
3. The number of clusters must be pre-specified.

In [10] Parsopoulos and Vrahitis observed that when they applied the *gbest* method (i.e., the swarm population only uses a single global best) to a multimodal function, the swarm moved back and forth, failing to decide where to land. This behavior is largely caused by particles getting equally good information from those equally good global optima. To overcome this problem, they introduced a method in which a potentially good solution is isolated once it is found (if its fitness is below a threshold value ϵ), then the fitness landscape is "stretched" to keep other particles away from this area of the search space. The isolated particle is checked to see if it is a global optimum, and if it is below the desired accuracy, a small population is generated around this particle to allow a finer search in this area. The main swarm continues its search for the rest of the search space for other potential global optima. With this modification, their PSO was able to locate all the global optima of the test functions successfully.

Brits, et al. proposed a NichePSO [2], which has a number of improvements to Parsopoulos and Vrahitis's model. In NichePSO, multiple subswarms are produced from a main swarm population to locate multiple optimal solutions in the search space. Subswarms can merge together, or absorb particles from the main swarm. Instead of using the threshold ϵ as in Parsopoulos and Vrahitis's model, NichePSO monitors the fitness of a particle by tracking its variance over a number of iterations. If there is little change in a particle's fitness over a number of iterations, a subswarm is created with the particle's closest neighbour. The authors used a swarm of population size of 20-30, and NichePSO found all global optima of the test functions used within 2000 iterations.

Li, et al. introduced a species conserving genetic algorithm (SCGA) for multimodal optimization [9]. SCGA adopted a new technique for dividing the population based on the notion of species, which is added to the evolution process of a conventional genetic algorithm. Their results on multimodal optimization have shown to be substantially better than those found in literature.

The notion of species is very appealing. To some extent, it provides a way of addressing the three limitations we identified with the clustering approach used in the PSO proposed by Kennedy [5]. This paper proposes a species-based PSO (SPSO) incorporating the idea of species into PSO for solving the multimodal optimization problems. At each iteration step, SPSO aims to identify multiple species (each for a potential optimum) within a population and then determine a neighbourhood best for each species. These multiple adaptively formed species are then used to optimize towards multiple optima in parallel, without interference across different species.

3 Particle Swarm

The particle swarm algorithm is an optimization technique inspired by the metaphor of social interaction observed among insects or animals. The kind of social interaction modeled within a PSO is used to guide a population of individuals (so called particles) moving towards the most promising area of the search space. In a PSO algorithm, each particle is a candidate solution equivalent to a point in a d-dimensional space, so the i-th particle can be represented as $\mathbf{x_i} = (x_{i1}, x_{i2}, \ldots, x_{id})$. Each particle "flies" through the search space, depending on two important factors, $\mathbf{p_i} = (p_{i1}, p_{i2}, \ldots, p_{id})$, the best position the current particle has found so far; and $\mathbf{p_g} = (p_{g1}, p_{g2}, \ldots, p_{gd})$, the global best position identified from the entire population (or within a neighbourhood). The rate of position change of the i-th particle is given by its velocity $\mathbf{v_i} = (v_{i1}, v_{i2}, \ldots, v_{id})$. Equation (1) updates the velocity for each particle in the next iteration step, whereas equation (2) updates each particle's position in the search space [6]:

$$\boldsymbol{v}_{id}(t) = \chi(\boldsymbol{v}_{id}(t-1) + \varphi_1(\boldsymbol{p}_{id} - \boldsymbol{x}_{id}(t-1)) + \varphi_2(\boldsymbol{p}_{gd} - \boldsymbol{x}_{id}(t-1))) \quad (1)$$

$$\boldsymbol{x}_{id}(t) = \boldsymbol{x}_{id}(t-1) + \boldsymbol{v}_{id}(t) \, , \quad (2)$$

where

$$\chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad \text{and} \quad \varphi = \varphi_1 + \varphi_2, \ \varphi > 4.0. \quad (3)$$

Two common approaches of choosing $\boldsymbol{p_g}$ are known as *gbest* and *lbest* methods. In the *gbest* approach, the position of each particle in the search space is influenced by the best-fit particle in the entire population; whereas the *lbest* approach only allows each particle to be influenced by a fitter particle chosen from its neighbourhood. Kennedy and Mendes studied PSOs with various population topologies [8], and have shown that certain population structures could give superior performance over certain optimization functions.

4 Identifying Species

Central to the proposed SPSO in this paper is the notion of species. Goldberg and Richardson proposed a niching method based on speciation by fitness sharing [4], where a GA population is classified into groups according to their similarity measured by Euclidean distance. The smaller the Euclidean distance between two individuals, the more similar they are:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{n}(x_{ik} - x_{jk})^2}, \quad (4)$$

where $\mathbf{x_i} = (x_{i1}, x_{i2}, \dots, x_{in})$ and $\mathbf{x_j} = (x_{j1}, x_{j2}, \dots, x_{jn})$ are vectors of real numbers representing two individuals i and j from the GA population.

The definition of a species also depends on another parameter r_s, which denotes the radius measured in Euclidean distance from the center of a species to its boundary. The center of a species, so called species seed, is always the best-fit individual in the species. All particles that fall within the r_s distance from the species seed are classified as the same species.

4.1 Determining Species Seeds from the Population

The algorithm for determining species seeds introduced by Li et al. is adopted here [9]. By applying this algorithm at each iteration step, different species seeds can be identified for multiple species and then used as the *lbest* for different species accordingly. Fig.1 summarizes the steps for determining the species seeds.

The algorithm (as given in Fig. 1) for determining the species seeds is performed at each iteration step. The algorithm takes as an input, L_{sorted}, a list

```
input  : L_sorted - containing all particles sorted in decreasing order fitness
output : S - containing dominating particles identified as species seeds
begin
    S = Φ;
    while not reaching the end of L_sorted do
        found ← FALSE;
        for all p ∈ S do
            if d(s, p) ≤ r_s then
                found ← TRUE;
                break;
            end
        end
        if (not found) then
            let S ← S ∪ {S}
        end
    end
end
```

Fig. 1. The algorithm for determining the species seeds.

containing all particles sorted in decreasing order of fitness. The species seed set S is initially set to Φ . All particles are checked in turn (from best to the least-fit) against the species seeds found so far. If a particle does not fall within the radius r_s of all the seeds of S, then this particle will become a new seed and be added to S. Fig. 2 provides an example to illustrate the working of this algorithm. In this case, applying the algorithm will identify s_1, s_2 and s_3 as the species seeds. Note that since a species seed is the best-fit particle in a species, other particles within the same species can be made to follow the species seed as the newly identified neighbourhood best (*lbest*). This allows particles within the same species to be attracted to positions that make them even fitter. Because species are formed around different optima in parallel, making species seeds the new neighbourhood bests will provide the right guidance for particles in different species to locate multiple optima.

The complexity of the above procedure can be estimated based on the number of evaluations of Euclidean distances between two particles that are required. Assuming there are N individuals sorted and stored on L_{sorted}, the **while** loop steps through L_{sorted} to see if each individual is within the radius r_s of the seeds on S. If S currently contains i number of seeds, then at best the **for** loop is executed only once when the particle considered is within r_s of the first seed compared; and at worst the **for** loop is executed i times when the particle falls outside of r_s of all the seeds on S. Therefore the number of Euclidean distance calculations required for the above procedure $T(N)$ can be obtained by the following [9]:

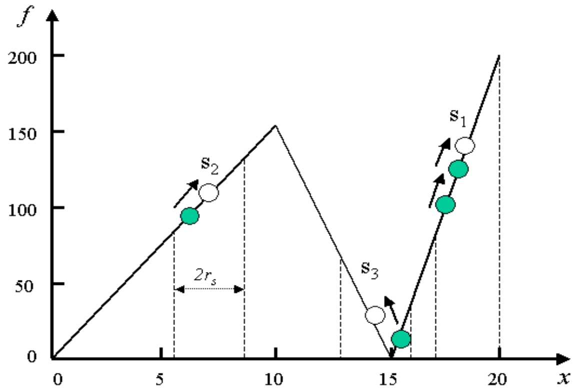


Fig. 2. An example of how to determine the species seeds from the population at each iteration step. s_1, s_2 and s_3 are chosen as the species seeds.

$$N \leq T(N) \leq \sum_{i=1}^{N}(i-1) = \frac{N(N-1)}{2}, \tag{5}$$

which gives the complexity of the procedure: $O(N^2)$.

5 The Species-Based PSO (SPSO)

Once the species seeds have been identified from the population, we can then allocate each seed to be the *lbest* to all the particles in the same species at each iteration step. The species-based PSO (SPSO) accommodating the above described algorithm for determining species seeds can be summarized in the following steps:

1. Generate an initial population with randomly generated particles;
2. Evaluate all particle individuals in the population;
3. Sort all particles in descending order of their fitness values (i.e., from the best-fit to least-fit ones);
4. Determine the species seeds for the current population (see Fig. 1);
5. Assign each species seed identified as the *lbest* to all individuals identified in the same species;
6. Adjusting particle positions according to equation (1) and (2);
7. Go back to step 2), unless the termination condition is met.

Considering the limitations of Kennedy's clustering-based PSO [5] (also discussed in section 2), SPSO improves in the following aspects:

1. SPSO only requires one iteration over all particles in the population in order to determine the species seeds, which are used as substitutes for neighbourhood bests (similar to the cluster centers in Kennedy's PSO).

2. In SPSO, an identified species seed is always the best-fit individual in that species.
3. There is no need to pre-specify the number of species seeds. They are automatically generated during a run.

6 Performance Measurements

The performance of SPSO in handling multimodal functions can be measured according to three criteria, *number of evaluations* required to locate the optima; *accuracy*, measuring the closeness to the optima, and *success rate*, i.e., the percentage of runs in which all global optima are successfully located.

To measure accuracy, we only need to check set S, which contains the species seeds identified so far. These species seeds are dominating individuals sufficiently different from each other, however they could be individuals with high as well as low fitness values (see Fig. 2). We can decide if a global optimum is found by checking each species seed in S to see if it is close enough to the known global optima (for all the test functions used in this study). A solution acceptance threshold ($0 < \epsilon \leq 1$) is defined to detect if the solution is close enough to a global optimum:

$$|f_{max} - f(x)| \leq \epsilon \tag{6}$$

where f_{max} is the known maximal (highest) fitness value for a test function (assuming maximization problems). If the number of global optima is greater than one, then all global optima will be checked for the required accuracy using equation (6) before a run is terminated.

7 Test Functions

The five test functions suggested by Beasley et al. [1] and the Rastrigin function (with different dimensions) were used to test SPSO's ability to locate a single or multiple maxima:

$$F1(x) = sin^6(5\pi x). \tag{7}$$

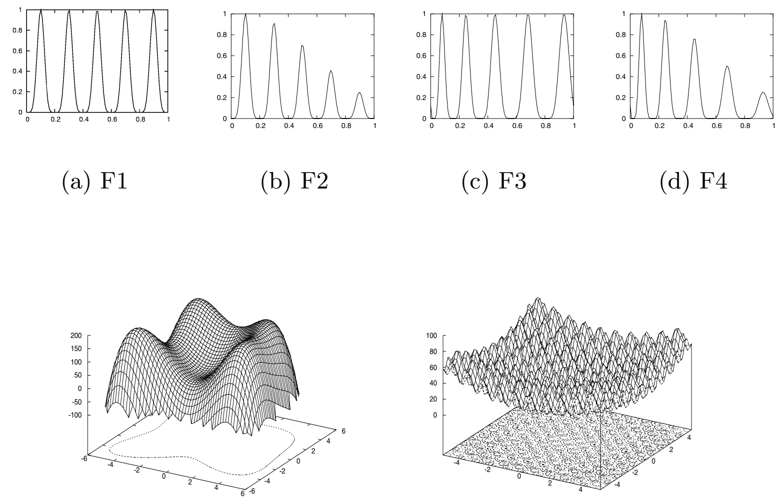
$$F2(x) = exp\left(-2log(2) \cdot \left(\frac{x-0.1}{0.8}\right)^2\right) \cdot sin^6(5\pi x). \tag{8}$$

$$F3(x) = sin^6(5\pi(x^{3/4} - 0.05)). \tag{9}$$

$$F4(x) = exp\left(-2log(2) \cdot \left(\frac{x-0.08}{0.854}\right)^2\right) \cdot sin^6(5\pi(x^{3/4} - 0.05)). \tag{10}$$

$$F5(x, y) = 200 - (x^2 + y - 11)^2 - (x + y^2 - 7)^2. \tag{11}$$

$$F6(\boldsymbol{x}) = \sum_{i=1}^{n} (x_i^2 - 10cos(2\pi x_i) + 10). \tag{12}$$



(a) F1 (b) F2 (c) F3 (d) F4

(e) F5:Himmelblau's function (f) F6:Rastrigin function

Fig. 3. Test functions.

As shown in Fig. 3, F1 has 5 evenly spaced maxima with a function value of 1.0. F2 has 5 peaks decreasing exponentially in height, with only one peak as the global maximum. F3 and F4 are similar to F1 and F2 but the peaks are unevenly spaced. F5 Himmelblau's function has two variables x and y, where $-6 \leq x, y \leq +6$. This function has 4 global maxima at approximately (3.58,-1.86), (3.0,2.0), (-2.815,3.125), and (-3.78,-3.28). F6 Rastrigin function, where $-5.12 \leq x_i \leq 5.12, i = 1, \ldots, 30$, has one global minimum (which is (0,0) for dimension=2), and many local minima.[1] F6 with a dimension of 2, 3, 4, 5 and 6 variables were used to test SPSO's ability in dealing with functions with numerous local minima and of higher dimensions.

[1] Rastrigin function can be easily converted to a maximization function.

Table 1. Summary of performance results (averaged over 30 runs).

Function	Num. of global optima	ϵ	r_s	Num. of evals. (mean and std dev)	Success rate
F1	5	0.0001	0.05	1383.33 ± 242.95	100%
F2	1	0.0001	0.05	351.67 ± 202.35	100%
F3	5	0.0001	0.05	1248.33 ± 318.80	100%
F4	1	0.0001	0.05	503.33 ± 280.07	100%
F5	4	0.0001	2.0	3155 ± 402.22	100%

Table 2. Comparison of results on F1 and F5.

Function	Num. of global optima	Algorithm	Num. of evals. required	Success rate
F1	5	Sequential Niched GA (SNGA)	1900	99%
		Species Conservation GA (SCGA)	3310	100%
		SPSO	**1383.33**	100%
F5	4	Sequential Niched GA (SNGA)	5500	76%
		SPSO	**3155**	100%

8 Experimental Setups

A swarm population size of 50 was used for all the above test functions. SPSO was run 30 times, each run with a maximum of 1000 iteration steps. The accuracy threshold ϵ was set to 0.0001. A run is terminated if either the required accuracy for all the global optima or the maximum of 1000 iteration steps is reached. r_s was set normally to a value between 1/20 to 1/10 of the allowed variable range. Success rate is measured by the percentage of runs (out of 30) locating all the global optima within the 1000 iteration steps. The number of function evaluations required for finding all the global optima are averaged over 30 runs. Table 1 provides a summary of the results.

For PSO parameters in equation (1) and (2), φ_1 and φ_2 were both set to 2.05. The constriction factor χ was set to 0.729844 [8]. Using this χ value produces a damping effect on the amplitude of an individual particle's oscillations, and as a result, the particle will converge over time. V_{max} was set to be the lower and upper bounds of the allowed variable ranges.

9 Discussion of the Results

As shown in Table 1, for F1 - F5, SPSO has converged to the required accuracy of 0.0001 with 100% success rate. SPSO found all the global optima in all runs with less than 1000 iteration steps. In comparison, NichePSO [2] only obtained similar accuracy values on F1, F3 and F5 after 2000 iterations. Furthermore, on F2 and F4, SPSO got better accuracy values than the NichePSO.

Table 2 shows that SPSO has the best results comparing with the results of SNGA proposed by Beasley, et al. [1] and SCGA proposed by Li, et al. [9] on F1 (equal maxima function) and F5 (Himmelblau's function).

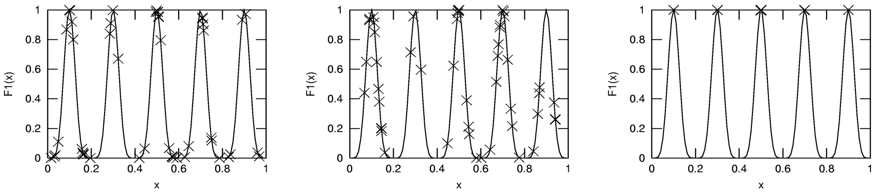


Fig. 4. A simulation run on F1(equal maxima), step 1, 4 and 74 from left to right.

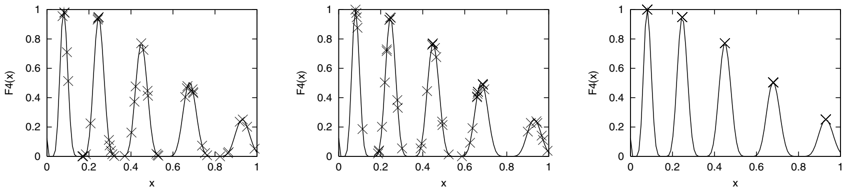


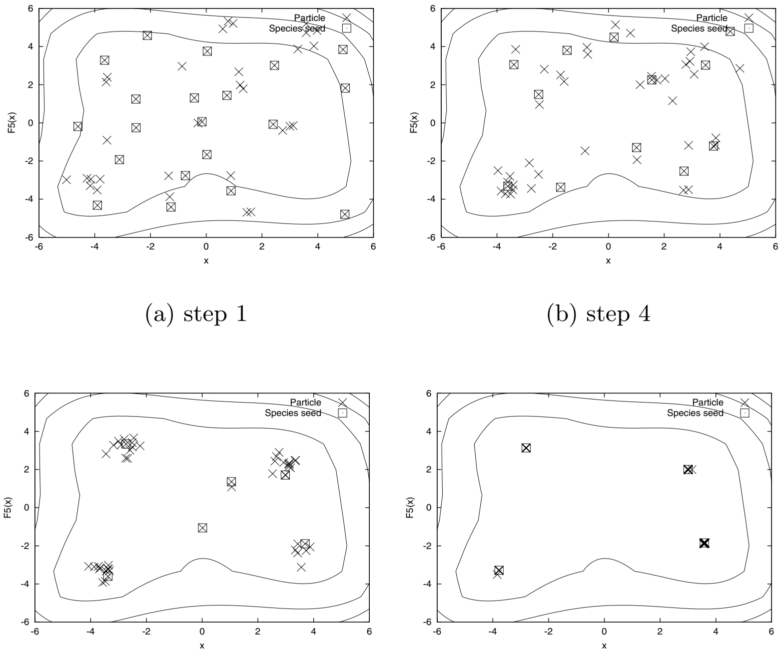
Fig. 5. A simulation run of SPSO on F4(uneven decreasing maxima) - step 1, 6 and 116 from left to right.

Fig. 4 shows that on F1, SPSO was able to locate all maxima, that is, at iteration step 74, all particles were able to converge to all 5 maxima. Fig. 5 shows that on F4, SPSO always found the highest peak first, then was also able to locate all other lower peaks in later iteration steps successfully, regardless of if they are global maxima or local optima. The results on F2 and F3 are similar to those of F1 and F4. Fig. 6 shows that on F5, many species seeds (based on the r_s value) were identified by SPSO initially as expected. Over the following iteration steps, these species were merged to form 4 groups around the 4 maxima. Eventually almost all particles converged to these 4 maxima at step 66.

Table 3 shows the results of SPSO on the Rastrigin function with dimension varying from 2 to 6. In this experiment the same parameter settings were used as the previous ones. It is interesting to note that on the Rastrigin function SPSO has increasing difficulty to converge to the required accuracy as the dimension is increased from 3 to 6. This is expected, as SPSO is designed to encourage forming species depending on the local feedback on the fitness landscape. The higher dimension and the presence of a large number of local minima of the Rastrigin function would demand SPSO to have a larger initial population in order to locate the global minimum. Further investigation on this will be carried out in future.

10 Conclusion

By using the concept of species, we have developed a PSO which allows the swarm population to be divided into different species adaptively, depending on the feedback obtained from the fitness landscape discovered at each iteration



(a) step 1 (b) step 4

(c) step 10 (d) step 66

Fig. 6. A simulation run of SPSO on F5 - step 1, 4, 10 and 66.

step during a run. Particles from each identified species follow a chosen neighbourhood best to move towards a promising region of the search space. Multiple species are able to converge towards different optima in parallel, without interference across different species. In a classic GA algorithm, crossover carried out over two randomly chosen fit individuals often produces a very poor offspring (imagining the offspring are somewhere between two fitter individuals from two distant peaks). In contrast, SPSO seems to be able to alleviate this problem effectively.

Tests on a suite of widely used multimodal test functions have shown that SPSO can find all the global optima for the all test functions with one or two dimensions reliably (with 100% success rate), and with good accuracy (< 0.0001), although SPSO seemed to show increasing difficulty to converge as the dimension of the Rastrigin function was increased to more than three. Comparison of SPSO's results with other published works has demonstrated that SPSO is comparable or better than the existing evolutionary algorithms as well as another niche-based PSO for handling multimodal function optimization, not only with regard to success rate and accuracy, but also on computational cost. In future, we will apply SPSO to large and more complex real-world multimodal optimization

Table 3. Results on **F6 Rastrigin** function (averaged over 30 runs).

Dimension	ϵ	r_s	Num. of evals. (mean and std dev)	Success rate
2	0.0001	2.0	3711.67 ± 911.87	100%
3	0.0001	2.0	9766.67 ± 4434.86	100%
4	0.0001	2.0	36606.67 ± 14662.38	33.3%
5	0.0001	2.0	44001.67 ± 10859.84	26.7%
6	0.0001	2.0	50000 ± 0.00	0%

problems, especially problems that we have only little (or no) prior knowledge about the search space. We also need to investigate how to best choose the species radius, for example perhaps looking at how to adaptively choose the species radius based on the feedback obtained during the search.

References

1. Beasley, D., Bull, D.R., and Martin, R.R.: A Sequential Niche Technique for Multimodal Function Optimization. Evolutionary Computation, 1(2): 101–125 (1993)
2. Brits, R., Engelbrecht, A. P., and van den Bergh, F.: A Niching Particle Swarm Optimizer. In: Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning 2002 (SEAL 2002) 692–696
3. Clerc, M. and Kennedy, J.: The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space. In: IEEE Transactions on Evolutionary Computation, vol. 6, no. 1, (2002) 58–73
4. Goldberg, D.E. and Richardson, J.: Genetic algorithms with sharing for multimodal function optimzation. In: Grefenstette, J.J. (eds.): Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms. Lawrence Earlbaum, Hillsidale, New Jersey (1987) 41–49
5. Kennedy, J.: Stereotyping: Improving Particle Swarm Performance with Cluster Analysis. In: Proceedings of the 2000 Congress on Evolutionary Computation. Piscataway, NJ: IEEE Service Center (2000) 1507–1512
6. Kennedy, J. and Eberhart, R.: Swarm Intelligence. Morgan Kaufmann Academic Press (2001)
7. Kennedy, J. and Eberhart, R.C.: Particle Swarm Optimization. In: Proceedings of the IEEE International Conference on Neural Networks, IV. Piscataway, NJ: IEEE Service Center (1995)1942–9148
8. Kennedy, J. and Mendes, R.: Population Structure and Particle Swarm Performance. In: Proceedings of the 2002 Congress on Evolutionary Computation. Piscatawat, NJ: IEEE service Center (2002) 1671–1675
9. Li, J.P., Balazs, M.E., Parks, G. and Clarkson, P.J.: A Species Conserving Genetic Algorithm for Multimodal Function Optimization. Evolutionary Computation 10(3): 207–234 (2002)
10. Parsopoulos, K. E. and Vrahatis, M. N.: Modification of the particle swarm optimizer for locating all the global minima. In: Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA 2001), Prague, Czech Republic. (2001) 324–327