Adaptive Radius Species Based Particle Swarm Optimization for Multimodal Optimization Problems

Zhi-Hui Zhan (Corresponding Author), *Member, IEEE*, Zi-Jia Wang, *Student Member, IEEE*, Ying Lin, *Member, IEEE*, and Jun Zhang (Corresponding Author), *Senior Member, IEEE*

***Abstract*—Multimodal optimization problem always has several peaks that are all optima of the problem. A promising approach to deal with such kind of problem should locate the peaks as many as possible (e.g., all the peaks) and should obtain high accuracy in each peak. The species-based particle swarm optimization (SPSO) divides the population into several subpopulations. Each subpopulation is gathered around a neighborhood best called species seed within the radius *r*, trying to locate different peaks. It does well in some low-dimensional multimodal optimization problems. However, the parameter *r*, which is associated with the efficiency and the accuracy of the algorithm, must be specified by the users. This makes SPSO very difficult for users to determine how much the parameter *r* should be. In this paper, a method of adaptively choosing radius *r* in SPSO is proposed, termed as adaptive SPSO (ASPSO). The experimental results show that the performance of ASPSO is more effective and accurate than standard SPSO in dealing with low-dimensional multimodal optimization problems.**

Keywords—species-based particle swarm optimization; adaptive radius; multimodal; optimization

I. Introduction

Particle swarm optimization (PSO) [1][2] is a popular swarm intelligence paradigm [3], which was first introduced by Kennedy and Eberhart in 1995 as global search algorithm. PSO mimics the behavior of birds flocking and fish schooling to guide the particle to search the optimal solutions. The members of the swarm are called particles, which mean possible solutions in the search space. Each particle in a population adjusts its position based on its own best position has found so far (the so-called *pbest*) and the best-fit particle in the entire population (the so-called *gbest*). In this way, the best-fit particle can guide the entire population to the optimal solution. At last, they will all converge to the optimal solution in the search space. As PSO is easy to accomplish, it has many successful applications in solving real-world optimization problems [4]-[9].

However, when dealing with multimodal optimization problems, the *gbest* particle guidance search fashion will face difficulties. Multimodal optimization problem is a kind of optimization problem that always has several peaks. These peaks are all optima of the problem. A promising approach to deal with such kind of problem should locate the peaks as many as possible (e.g., all the peaks) and should obtain high accuracy in each peak. Therefore, traditional PSO may lose the diversity easily due to the attraction of the *gbest* particle, resulting in maybe only one peak can be find but all the others are missed.

Trelea did many researches and analysis of the stability and convergence in PSO [10]. He pointed out that although the PSO algorithm can converge to a point in the search space stably, it can easily get trapped in the local optima when dealing with multimodal optimization problems. The weakness has restricted widely used of PSO. Recent years, how to overcome this weakness has becoming an increasingly critical issue for many researchers. A number of various PSO algorithms have been proposed to overcome the problem [11]-[19]. However, they also bring their own problems. For instance, Mendes and Kennedy [11] introduced a fully informed particle swarm to solve single global optimization problems. It is true that it is effective in solving single-objective global optimizations, but it is not suitable for multimodal optimization due to topology-based neighborhood selection method. Liang et al. [12] introduced a comprehensive learning particle swarm optimizer to solve the multimodal functions. It can avoid the local optima, but has slow convergence speed in simple functions, especially during the late state of the searching process.

To achieve a better performance in dealing with the multimodal optimization problems, adaptively choosing neighborhood bests using species in PSO was first introduced by Li [19], termed as the species-based PSO (SPSO). The population is divided into several subpopulations based on their similarity in term of Euclidean distance. Each species are gathered around a neighborhood best called species seed within the radius *r*, trying to locate different peaks. Each particle in a population adjusts its velocity and position based on its own *pbest* and species seed as the neighborhood best (the so-called *lbest*) rather than the *gbest*. SPSO does well in some multimodal optimization problems. However, the parameter *r* must be specified by the users. It is very hard for users to know how much it should be set. Nevertheless, this radius parameter *r* is so important because it is associated with the efficiency and accuracy of the algorithm. How to determine the value for *r* is significant for the SPSO

Z.-H. Zhan and J. Zhang are with the School of Computer Science and Engineering, South China University of Technology, 510006 Guangzhou, P. R. China. Zhi-Hui Zhan and Jun Zhang are the corresponding authors: cszhanzhh@scut.edu.cn; junzhanghk@qq.com.

Z. J. Wang is with the School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, 510275, China.

Y. Lin is with the Department of Psychology, Sun Yat-Sen University, Guangzhou, 510275, China.

This work was partially supported by the National Natural Science Foundations of China (NSFC) with No. 61402545, the Natural Science Foundations of Guangdong Province for Distinguished Young Scholars with No. 2014A030306038, the Project for Pearl River New Star in Science and Technology with No. 201506010047, and the NSFC Key Program with No. 61332002, and the Fundamental Research Funds for the Central Universities (15lgzd08).

algorithm.

In this paper, a method of adaptively choosing radius r in SPSO (termed adaptive SPSO, ASPSO) is proposed. As the convergence of the particles during the evolutionary process, the distances between each two particles are changed at the end of each iteration step. Therefore, the radius r should be changed during each iteration step dynamic, so that the particles can be divided into different species. In the initialization when no subpopulation information exists, the radius r is set as the average distance of each two particles. During the evolutionary process, as the subpopulations are formed, if the number of subpopulations is not smaller than the number of the known optimal (local or global), r remains unchanged. Otherwise, the r should be reset by the mean of distances from each particle to its species seed. Experimental results show that the performance of ASPSO is more effective and more accurate than standard SPSO in dealing with multimodal optimization problems.

The rest of the paper is organized as follows. Section II introduces the standard PSO and some variant algorithms of improved PSO. Section III describes the ASPSO based on the SPSO, and introduces the details of algorithm process. Section IV presents the test functions, the results, and discussions. Conclusions are given in Section V.

II. Background

A. Traditional PSO Algorithm

PSO is a search technique that emulates the swarm behavior of birds flocking and fish schooling, whose individuals represent points in the D-dimensions search space. Each particle i is associated with two vectors. The vector $V_i=[v_{i1},v_{i2},\ldots,v_{iD}]$ represents the velocity of the particle i, while the vector $X_i=[x_{i1},x_{i2},\ldots,x_{iD}]$ means the position of the particle i. Both the two vectors are initialized randomly within the corresponding ranges. At the end of the each iteration, each particle in a swarm population adjusts its velocity and position based on its own *pbest* has found so far and the *gbest* in the entire population. The velocity V_i and position X_i of each particle are updated according to the following formulas:

$$\begin{aligned} v_{id} &= \omega \cdot v_{id} + c_1 \cdot r_{1d}(pbest_{id} - x_{id}) \\ &+ c_2 \cdot r_{2d} \cdot (gbest_d - x_{id}) \end{aligned} \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

where ω is the inertia weight [17], to balance the global and local search performance, often initialized as 0.9, and decrease to 0.4 linearly by the process of the evolution. c_1 and c_2 are the acceleration coefficients, often set as 2.0. Parameter c_1 pulls the particle to its own *pbest*, ensuring the diversity of the population; while c_2 pushes the swarm to converge to the current *gbest*, ensuring the speed of the convergence. r_{1d} and r_{2d} are two uniformly distributed random numbers within the range [0, 1]. A particle's velocity and position on each dimension are clamped to the maximum $V_{\max}$ and $X_{\max}$ respectively, where $X_{\max}$ is the search range determined by the problem itself while $V_{\max}$ is often set as 20% of the search range. The flowchart can be seen in Fig. 1.

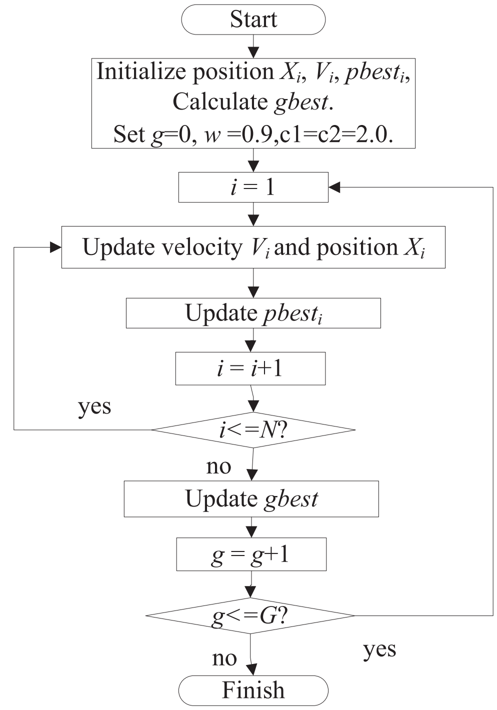


Fig. 1 Flowchart of the PSO.

B. Related Works of PSO

Because of its efficiency and easy to accomplish, PSO has been widely used in solving single modal problems. However, it can easily get trapped in the local optima when dealing with some sophisticated problems. Recent years, these disadvantages have drawn many scientists' attention to overcome. Many researches on performance improvement have been reported. Shi and Eberhart [17] proposed the large inertia weight ω is appropriate for global search, while the small inertia weight ω is suitable for local search. In that way, ω should decrease linearly with the evolutionary process. It can be described as:

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \cdot g / G \quad (3)$$

where g is the current evolutionary generation, and G is the maximum number of generations. Besides, the parameter $\omega_{\max}$ and $\omega_{\min}$ are often set as 0.9 and 0.4.

In addition, designing different types of topologies for improving PSO's performance was first described by Kennedy [11]. It seems simper but better using the particles' own local neighborhood, which can be regarded as a single region in the population topology. In that way, all neighbors are a source of influence in leading the particles to update. There are five different PSO topologies often used called all, ring, four clusters, pyramid, and square.

In order to solve multimodal optimization problem, the Species-based PSO (SPSO) was first proposed by Li [19]. In this algorithm, all the particles are divided into different subpopulations based on their similarity measured by the Euclidean distance. The smaller the Euclidean distance between two particles, the more similar they are. The distance of two particles p and s can be described with following:

$$dis(p,s) = \sqrt{\sum_{d=1}^{D}(p_d - s_d)^2} \quad (4)$$

All the subpopulations are identified by the dominant particles known as the species seed. The species seed is the best-fit particle in the species, other particles within the radius r can gather around the species seed and to follow the species seed as the identified neighborhood best (*lbest*). This allows particles with the same species seed to converge to the positions where make them even fitter. Because different species are formed and updated in parallel, the species seeds or the new neighborhood bests guide the particles in different species to locate multiple optima effectively.

The velocity V and position X of each particle in SPSO are updated according to the following formulate (5) and (6):

$$v_{id} = \chi(v_{id} + \varphi_1(p_{id} - x_{id}) + \varphi_2(p_{ld} - x_{id})) \quad (5)$$

$$x_{id} = x_{id} + v_{id} \quad (6)$$

where $\chi = \dfrac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|}$ and $\varphi = \varphi_1 + \varphi_2$, $\varphi > 4.0$, φ_1 and φ_2 are often set as 2.05.

The algorithm of determining species seeds of population can be seen in **Algorithm 1** in the form of a pseudo code. (We define the fitness value the larger the better).

Algorithm 1 Determining species seeds from the population

```
Lsorted: Sort the particles in decreasing order based on fitness
Begin
  S=Ø;  //S: The collection of species seeds
  While not reaching the end of Lsorted Do
    found=False;
    Pick up s from Lsorted;
    For all p ∈ S Do
      If d(p, s)<r //s is within the specie of p
        found=True;
        break;
      End
    End
    If found==False //s does not belong to any specie
      S=S∪{s}; //then s become a new specie seed
    End
  End
End
```

III. ASPSO WITH ADAPTIVE RADIUS

Although SPSO is effective in solving some multimodal optimization problems, the parameter r must be specified by the user. However, pre-define a value for r is difficult, even though the parameter r can significantly affect the efficiency and accuracy of the SPSO algorithm. Moreover, the distances between each two particles are changed at the end of each iteration step during the evolutionary process. Therefore, the r should also be changed if the algorithm wants to use the r value to help determine different species. Motivated by this observation, our proposed ASPSO is presented in this paper by setting the r value according with the distance information among the particles, so as to catch up with the search characteristics during the evolutionary process.

Similar to SPSO, we also define the particles that are within the radius r together around the species seed. As the initial particles are well distributed over the whole search space, the general level of the distance can be observed from average. Therefore, we initialize the parameter r as the mean of the distances of each two particles, as:

$$r = \left(\sum_{\substack{1 \le i \le N \\ 1 \le j \le N, j \ne i}} dis(x_i, x_j) \right) \Bigg/ \left(\frac{N \times (N-1)}{2} \right) \quad (7)$$

where N is the population size and $dis(x_i, x_j)$ means the distance of every two different particles.

This would be reasonable in the initialization because we know nothing about the species. However, as different species (subpopulations) have formed after the first generation, it seems not a good way to calculate the r value by including the distance information of particles in different species. This is because that the distance between particles in different species is often too large that will make (7) result in large value that is not suitable for locating correct species. Therefore, during the evolutionary process, it would be better to set the r value by using the mean distance of the particles to their species seeds, as:

$$r = \frac{1}{N - |S|} \times \left(\sum_{1 \le i \le N}^{x_i \notin S} dis(x_i, specie - seed - of - x_i) \right) \quad (8)$$

where N is the population size and $|S|$ is the number of species seeds. Moreover, $dis(x_i, specie\text{-}seed\text{-}of\text{-}x_i)$ means the distance between the particle x_i and its corresponding specie seed, where x_i can not be a specie seed itself.

That is, a very natural way to configure the r value in ASPSO is that: the r value is set by (7) in the initialization and is set by (8) every generation during the evolutionary process. However, when the subpopulations have formed during the evolutionary process, the distance between the particles and their species seeds will become smaller and smaller generation by generation. In this case, each particle will become a subpopulation finally. This is not good for the search because what we want is to make particles concentrate on different peaks whether local or global.

Based on this reason, the r should not be set as (8) in every generation, but should be changed or unchanged adaptively. That is, at the end of every generation, ASPSO compares the number of subpopulations formed in the current generation and the number of the known optima of the problem. This is reasonable because we may have the information of peak number of the problem, while the algorithm is to find out where the peeks locate and their accuracy values. Therefore, in ASPSO, at the end of each generation, if the number of subpopulations is not smaller than the known optimal (local or global), which means r is sufficiently small in the current generation, then r remains unchanged for the next generation. Otherwise, the r should be reset by the mean of distances from each particle to its species seed, as (8), which can be used in the next generation. In this method, it can effectively avoid the influence from the distances between different subpopulations and avoid each particle becoming a subpopulation finally. Hence, using this method to assign r is obviously effective than using r as a random number by the users. As a result, ASPSO is highly effective for fine convergence and thereby successfully locates the desired peaks with high accuracy.

There are two main advantages of ASPSO to solve multimodal optimization problems as follows:

1. The benefit of SPSO velocity update equation which mentioned above to ensure making full use of a fitter particle chosen from its neighborhood information, especially during the later state of the search process, which can lead to fast convergence and a high accuracy;
2. Adaptive radius selection to ensure the Euclidean distance calculation is from same subpopulation and this increases the algorithm's ability for local search.

With these two advantages, ASPSO can easily find most of the peaks and maintain them until the predefined evolutions for multimodal optimization. The details of ASPSO algorithm are shown in **Algorithm 2**.

Algorithm 2 Steps of ASPSO
Step1 Randomly generate the initial solutions.
Step2 Evaluate the initial solutions and initialize the *pbest*.
Step3 Initialize the radius r using the mean of Euclidean distance from each two particles in the population, as (7).
Step4 Generate the species seeds using the **Algorithm 1** mentioned above.
Step5 Update the particle's velocity using (5).
Step6 Update the particle's position using (6).
Step7 Evaluate the newly generated particles.
Step8 Update the *pbest* for each particle.
Step9 If the number of subpopulations is not smaller than the known optimal, go to Step 11. Otherwise, go to Step 10.
Step10 Reset r with the mean distances from each particle to its species seed, as (8).
Step11 Stop if a termination is satisfied. Otherwise, go to Step4

IV. Benchmark Tests and Comparisons

A. Benchmark Functions

As we wish to test the ASPSO on diverse functions and our main object is to improve the SPSO's performance on multimodal functions. We choose five test functions [20]. The properties and the formulas of these functions are presented below:

$$\mathrm{F1}(x) = \sin^6(5\pi x)$$

$$\mathrm{F2}(x) = \exp\left(-2\log(2)\bullet\left(\frac{x-0.1}{0.8}\right)^2\right)\bullet\sin^6(5\pi x)$$

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

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

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

As shown in Fig. 2, F1-F4 have one variable x, where $0\le x\le 1$; F5 has two variables x and y, where $-6\le x$, $y\le +6$. Besides, F1 has 5 evenly distributed maxima with the value of 1.0. F2 has 5 peaks, but with only one peak as the global maximum. F3 and F4 are similar to F1 and F2 but the peaks are unevenly distributed. F5 has 4 global maxima at approximately (3.58,-1.86), (3.0, 2.0), (-2.815, 3.125), and (-3.78,-3.28).

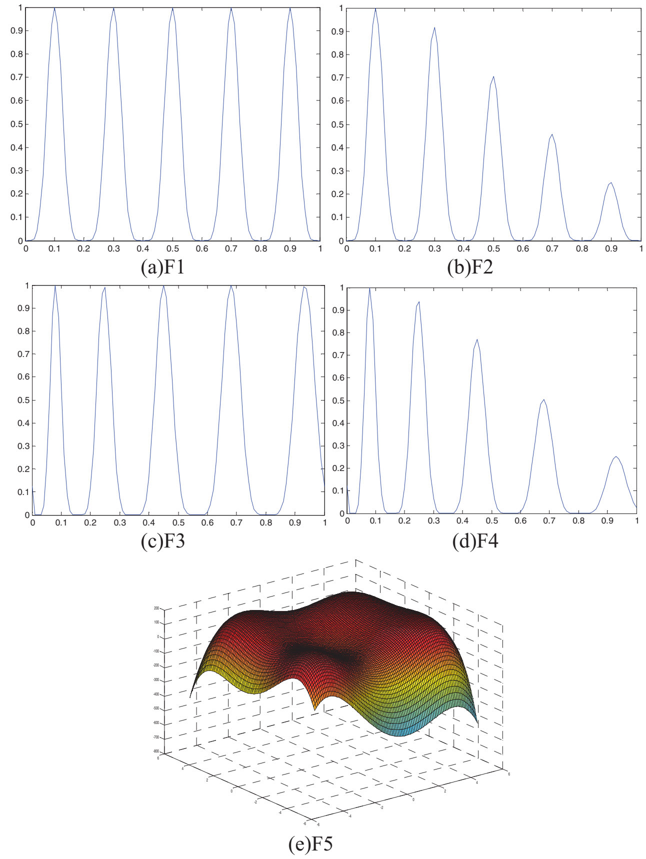


Fig 2. Test functions.

B. Parameter Setting and Performance Measure

In this experiment, a standard of accuracy $0<\varepsilon<1$, which means how close the computed solutions to the known global peaks, need to be same and satisfied in order to compare the two different algorithms. The accuracy is set on 0.0001.

A swarm population size of 50 was used for all the above test functions. In order to compare the performance of these two algorithms and avoid the influence of accidental factors, 30 independent runs for algorithms are taken on each problem. A run is terminated if either the required accuracy for all the global optima or the maximum of 1000 iteration steps is reached. The performance of the two multimodal algorithms is measured in terms of the following criteria:
(1) success rate (the percentage of runs in which all the global peaks are successfully located);
(2) number of evaluations required to reach the accuracy.

The success rates for the two algorithms on five test functions are recorded and presented in Table I. For clarity, the better results are marked in **boldface**. Moreover, the results are compared by Wilcoxon's rank sum tests and t-tests with significance level α =0.05 to see whether the difference is significant, where 'Y' means the difference is significant, 'N' means the difference is not significant.

As can be seen from the Table I, although both ASPSO and SPSO achieve the same success rate, the ASPSO achieves smaller evaluations on these five test functions. Also, from the t-tests, we can see the P value is smaller than 0.05 for the function F1, F3, and F5, which means on these test functions,

TABLE I. COMPARISION OF PERFORENCE RESULTS (AVERGED OVER 30 TIMES)

Function	Number of global optima	ε	Algorithms	r	Number of evaluations Required (mean $\pm$ std_dev)	Success rate	t-test (alpha=0.05)	Wilcoxon's rank sum tests (alpha=0.05)
F1	5	0.0001	ASPSO		**1216.67 $\pm$ 256.18**	100%	Y	Y
			SPSO	0.05	1765 $\pm$ 477.94	100%		
F2	1	0.0001	ASPSO		**308.33 $\pm$ 149.26**	100%	N	Y
			SPSO	0.05	396.67 $\pm$ 188.34	100%		
F3	5	0.0001	ASPSO		**1205 $\pm$ 325.77**	100%	Y	Y
			SPSO	0.05	1566.67 $\pm$ 346.25	100%		
F4	1	0.0001	ASPSO		**466.67 $\pm$ 201.70**	100%	N	Y
			SPSO	0.05	563.33 $\pm$ 288.37	100%		
F5	4	0.0001	ASPSO		**3071.67 $\pm$ 561.84**	100%	Y	Y
			SPSO	2.0	3726.67 $\pm$ 677.68	100%		

the two groups of data are obviously different, so the performance of ASPSO is better than SPSO. However, the *P* value is larger than 0.05 from the functions F2 and F4. The reason of this phenomenon is that when we do *t*-test, two samples are required for normal distribution. However, these statistical results are not necessarily normal distribution. Based on this problem, we use Wilcoxon's rank sum tests to analyze these data again.

Observed from the Table I of the Wilcoxon's rank sum tests, the *P* value is smaller than 0.05 for all the test functions. Therefore, the two groups of data are obviously different, indicating that the performance of ASPSO is better than SPSO.

The better performance is due to the better choosing parameter *r* rather than setting it randomly. As ASPSO selects *r* based on distances, the search can be easily located at different peaks with subpopulations during the later state of search, which is suitable for multimodal optimization. The Fig. 3 is the diagram of the radius and the number of iterations in F5, showing that the radius adaptively changed during the evolutionary process.

Besides, we can also see their results on the tested functions from Fig.4 to Fig.8. The left is SPSO, while the right is ASPSO.

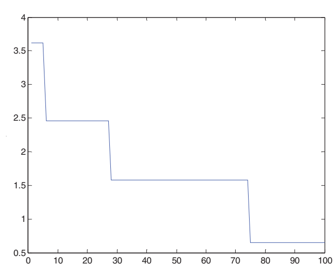
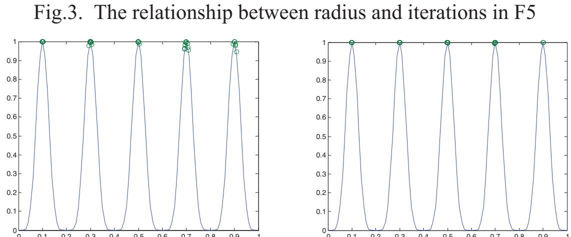
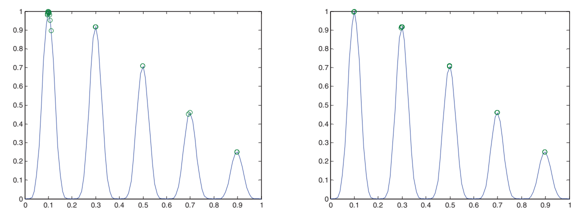


Fig.3. The relationship between radius and iterations in F5



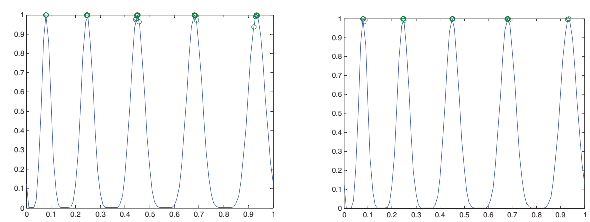
SPSO ASPSO

Fig.4. The results of two algorithms on F1, number of iterations is 50



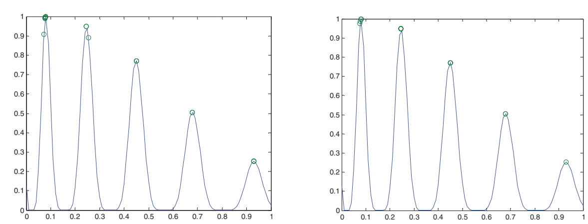
SPSO ASPSO

Fig.5. The results of two algorithms on F2, number of iterations is 50



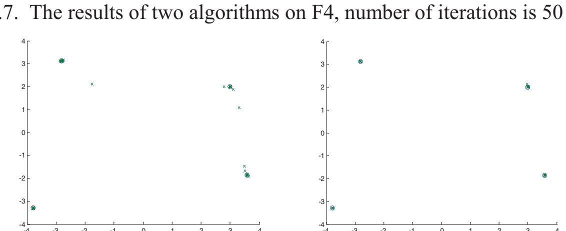
SPSO ASPSO

Fig.6. The results of two algorithms on F3, number of iterations is 50



SPSO ASPSO

Fig.7. The results of two algorithms on F4, number of iterations is 50



SPSO ASPSO

Fig.8. The results of two algorithms on F5, number of iterations is 80

C. *Discussions of the Results*

Fig. 4 shows that on F1, with the same number of iterations 50, both ASPSO and SPSO are able to locate all maxima. However, the ASPSO has faster speed of convergence and higher accuracy; Fig. 5 shows that on F2, with the same number of iterations 50, both SPSO and ASPSO can find the highest peak. Similarly, the ASPSO has faster speed of convergence and high accuracy. The results on F3 and F4 are similar to the results of F1 and F2. The only difference is in functions F3 and F4, their peaks are unevenly distributed. Also, the ASPSO can locate all maxima more quickly and accurately.

In Fig. 8, as we have known the 4 global maxima in F5, we can only use the known solutions in coordinate to see if the computation solutions from algorithms can get them during iterations. Of course, as we can see, with the same number of iterations 80, the ASPSO has faster speed of convergence and higher accuracy than SPSO.

V. Conclusion

In this paper, an ASPSO adaptively choosing radius algorithm was proposed to solve multimodal problems. ASPSO made use of the benefits both from the species seed (SPSO) and adaptive radius selection. With the neighborhood best information, the algorithm was enhanced with local search ability. Experimental results also showed that the proposed ASPSO algorithm can perform better than SPSO in this paper on a set of widely used multimodal test functions in a statistically meaningful way.

In the future, we wish we can apply ASPSO to more complex multimodal optimization problems with increasing the probability of more global or local peaks, even the problems we have little or no prior knowledge such as in a dynamic environment, and without the information of peak number.

References

[1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Network*, 1995, vol. 4, pp. 1942-1948.

[2] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micromachine Human Sci.*, Nagoya, Japan, 1995, vol. 1, pp. 39–43.

[3] J. Kennedy, R. C. Eberhart, and Y. H. Shi, *Swarm Intelligence*. San Mateo, CA: Morgan Kaufmann, 2001.

[4] L. Mezai and F. Hachouf, "Score-level fusion of face and voice using particle swarm optimization and belief functions," *IEEE Trans. Human-Machine Systems*, vol. 45, no. 6, pp. 761-772, 2015.

[5] J. J. Kim, and J. J. Lee, "Trajectory Optimization With Particle Swarm Optimization for Manipulator Motion Planning," *IEEE Transactions on Industrial Informatics,* vol. 1, no. 3, pp. 620-631, 2015.

[6] P. Hou, W. Hu, M. Soltani, and Z Chen, "Optimized Placement of Wind Turbines in Large-Scale Offshore Wind Farm Using Particle Swarm Optimization Algorithm," *IEEE Transactions on Sustainable Energy,* vol. 6, no. 4, pp. 1272-1282, 2015.

[7] R. D. Villarroel, and E. F. Caicedo, "Particle swarm optimization vs genetic algorithm, application and comparison to determine the moisture diffusion coefficients of pressboard transformer insulation," *IEEE Transactions on Dielectrics and Electrical Insulation,* vol. 22, no. 6, pp. 3574-3581, 2015.

[8] H. Wang and G. Yen, "Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system," *IEEE Trans. Evol. Computat.*, vol. 19, no. 1, pp. 1-18, 2015.

[9] J. Kim and J. Lee "Trajectory optimization with particle swarm optimization for manipulator motion planning," *IEEE Trans. Ind. Informact.*, vol. 11, no. 3, pp. 620-631, 2016.

[10] I. C. Trelea, "The particle swarm optimization algorithm: Convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, no. 6, pp. 317-325, 2003.

[11] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: Simpler, maybe better," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 204–210, Jun. 2004.

[12] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, Jun. 2006.

[13] Z. J. Wang, Z. H. Zhan and J. Zhang, "An Improved Method for Comprehensive Learning Particle Swarm Optimization," in *Proc. IEEE Symposium Series on Computational Intelligence*, Dec. 2015, pp.218-225.

[14] W. N. Chen, J. Zhang, H. S. Chung, W. L. Zhong, W.-G. Wu, and Y.-H. Shi, "A novel set-based particle swarm optimization method for discrete optimization problems," *IEEE Trans. Evol. Comput.*, vol. 14, no. 2, pp. 278-300, 2010.

[15] W. N. Chen, J. Zhang, Y. Lin, N. Chen, Z. H. Zhan, H. S. H. Chung, Y. Li, and Y. H. Shi, "Particle Swarm Optimization With an Aging Leader and Challengers," *IEEE Trans. Evol. Comput.*, vol. 17, no. 2, pp. 241-258, 2013.

[16] Y. J. Gong, J. J. Li, Y. Zhou, Y. Li, H. S. H. Chung, Y. H. Shi, and J. Zhang, "Genetic Learning Particle Swarm Optimization," *IEEE Trans. Cybern.*, in press, 2015.

[17] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE World Congr. Comput. Intell.*, 1998, pp. 69–73.

[18] B. Y. Qu, P. N. Suganthan, and S. Das, "A distance-based locally informed particle swarm model for multimodal optimization," *IEEE Trans. Evol. Comput.*, 2013, vol. 17, no. 3, pp. 387-402.

[19] X. Li, "Adaptively choosing neighborhood bests using species in a particle swarm optimizer for multimodal function optimization," in *Proc. Genetic and Evolutionary Computation Conference*, Jun. 2004, pp. 105-116.

[20] D. Bealey, D. R. Bull, and R. R. Martin, "A sequential niche technique for multimodal function optimization," *Evolutionary Computation*, 1993, vol. 1, no. 2, pp. 101-125.