International Journal of Electronics Communication and Computer Engineering
Volume 6, Issue 4, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Adaptive Method of Particle Swarm Optimization for Multimodal Function

Jiantao GUO, Xinwu CHEN, Youchao TU
College of Physics and Electronic Engineering, XinYang Normal University, Henan, 464000, China

***Abstract* – Multimodal optimization or finding more than one optimum is needed in many scientific and engineering application fields. Niching methods of particle swarm optimization has been successfully applied in many areas since it is a powerful, yet simple population based optimization strategy. A comprehensive analysis of neighborhood selection and information sharing mechanism was given from two aspects including topology technique and measure criterion. Moreover, most of existing niching particle swarm optimization algorithms which include SPSO, NichePSO, ARPSO, r2PSO, FER-PSO, ANPSO, FIPS-PSO, nbestPSO etc were described and compared. Furthermore, advantages and disadvantages existing in these algorithms were pointed out. Therefore, some ideas to improve the performance of niching particle swarm optimization algorithms were proposed.**

***Keywords* – Evolutionary Computation, Multimodal Function Optimization, Particle Swarms, Niching Technique, Adaptive Method.**

I. Introduction

Particle swarm optimization (PSO) is one of the swarm intelligence (SI) algorithms that was first introduced by Kennedy and Eberhart in 1995[1], inspired by swarm behaviors such as birds flocking and fishes schooling. PSO has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control, and other areas where GA can be applied since it is a powerful, yet simple population based optimization strategy.

In a canonical PSO, Individuals adjust their trajectory through the search space toward the position $\vec{p}_n$ of best particle from any member of the neighborhood (social influence) and its personal best position $\vec{p}_i$ it has found so far (learning from experience). Let us use $\vec{v}_i$ to denote the velocity of the i[th] particle in the swarm, $\vec{x}_i$ its position. In the "Type 1" constricted PSO suggested by Clerc and Kennedy [2], $\vec{v}_i$ and $\vec{x}_i$ of the i[th] particle in the swarm are updated according to the following two equations:

$$\vec{v}_i \leftarrow \chi(\vec{v}_i + U(0,\varphi_1)(\vec{p}_i - \vec{x}_i) + U(0,\varphi_2)(\vec{p}_n - \vec{x}_i)) \quad (1)$$

$$\vec{x}_i \leftarrow \vec{x}_i + \vec{v}_i \quad (2)$$

where $U(0,\varphi_1)$ and $U(0,\varphi_2)$ are two separate functions each returning a vector comprising random values uniformly generated in the ranges $[0,\varphi_1]$ and $[0,\varphi_2]$ respectively. φ_1 and φ_2 are commonly set to $\frac{\varphi}{2}$ (where φ is a positive constant 4.1). A constriction coefficient χ is used to prevent each particle from exploring too far away in the search space, since χ applies a dampening effect to the oscillation size of a particle over time.

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, so the common approaches of choosing $\vec{p}_n$ in (1) is gbest model which is influenced by the best-fit particle from the entire population. But In recent years, PSO has been used increasingly as an effective technique for solving complex and difficult optimization problems, especially for multimodal function optimization where exist a large number of such problems in the real world, this become too simple.

Multimodal optimization amounts to finding multiple global and local optima (as opposed to a single solution) of a function, so that the user can have a better knowledge about different optimal solutions in the search space and when needed, the current solution may be switched to a more suitable one while still maintaining the optimal system performance. Basically, however, in this case the entire swarm population can potentially be misled to local optima or result in redundant or missing of what have found solutions because it is not well preserved. If suitable particles can be determined as the appropriate neighborhood best particles to guide different portions of the swarm population moving toward 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. The question is how to determine which particles would be suitable as neighborhood bests, and how to assign them to the suitable particles in the population so that they will move toward different optima accordingly.

The paper is organized as follows. Section 2 provides a review of the classic niching methods, which require some parameters to determine the size of niche. Section 3 discusses the issue of improved search capacity methods based on parameter setting and population diversity giving some idea for following niches formation. Section 4 gives a comprehensive analysis of neighborhood selection and information sharing mechanism from two aspects including topology technique and measure criterion. Further, most of existing adaptive niching methods were described and discussed. Finally, Section 5 draws some conclusions and gives directions for future research regarding the adaptive multimodal function PSO without parameters.

II. Related Work

In Evolutionary Algorithms (EAs), optimization techniques that locate multiple optima in multi-modal function optimization problems are known as niching techniques [3]-[7]. Horn defines niching as a "form of cooperation around finite, limited resources, resulting in the lack of competition between such areas, and causing the formation of species for each niche" [8]. In EAs, individuals or particles compete for the use of available resources, niches emerge where individuals are organized into subgroups based on their resource requirements and species are partitions of a population competing within the environment.

Niching techniques have been carried out with the conflicting goals of maintaining a diversity that allows different niches finding with a view of preventing convergence to a single solution and decreasing local searching capability which need run a long time or even cannot find desirable solutions. A niching method generally modifies the behavior of a classical EA in order to maintain diversity or decrease convergence speed to locate different optima. Many niching methods have been developed over the years, including crowding [9], fitness sharing [10], restricted tournament selection (RTS) [11], and speciation [12]. Some of the classic niching techniques are presented in the following.

Li created SPSO[13] which use the notion of species to divide the swarm population into subpopulations based on their similarity measured by Euclidean distance. Typically, a fixed distance threshold called the niche radius r_s is specified. Two individuals are determined to be of the same species if their distance is below this threshold. Only individuals within the same niche may interact or breed while particles in different niches do not communicate, exploring different areas of the problem space in isolation. But the authors also need to investigate how to best choose the species radius defining the species because the niche radius setting has a strong impact on performance, and is difficult to tune, which is typically not known a priori in practice. Furthermore, by using a fixed niche radius setting, one implicitly assumes equally sized and spherically shaped niches.

Brits et al developed NichePSO [14] which differs from a normal PSO in that the unniched particles do not interact with each other, performing individual searches instead. If the fitness variance of a particle over the last 3 steps was less than a predefined amount δ, a subswarm is created consisting of that particle and its nearest topological neighbor. Subswarms are also merged when they intersect and are likely to converge on the same solutions. But a number of issues still need to be resolved, including the relationship between the number of niches and solutions, in addition to sensitivity of the algorithms to changes in parameter δ. Dependence on the parameters is weakening, but still associated with algorithm performance.

Table 1: Some thresholds about niche which affect the performance of algorithm

Algorithms	Cluster analysis	nbestPSO	SPSO[13]	NichePSO[14]
Parameters	C	k	r_s	δ
characters	Optima number	Particle neighborhood	species radius	Fitness variance

While the motivation and usefulness of niches is beyond doubt, most existing niching methods require specification of certain niching parameters as shown in Table 1. These niching parameters, often used to inform a niching algorithm how far apart between two closest optima or the number of optima in the search space, are typically difficult to set as they are problem dependent so as to choosing the right parameter values is thus a hard task. Especially, it is now understood that there are no "optimal values" for these parameters during evolution or in dynamic environment. Due to the algorithms are lack of self-adaptive adjustment function and some parameters need be set mutually according to problems, which limit the algorithms' flexibility and scope and make the algorithm difficult to practical applications widely [15].

III. Search Improvement

A. Parameter Setting

Attempts have been made to improve the PSO performance in recent years. Most of improved PSOs manipulate the control parameters with considering the varying states of evolution. Another approach to improve performance adopts a systematic treatment of evolutionary state to guide the search behavior of particles. These operations are not directly related to niches, but algorithms could enhance search capability that allows rapid and still avoid premature convergence.

To efficiently control the local search and convergence to the global optimum solution, time-varying acceleration coefficients are introduced in addition to the time-varying inertia weight factor in particle swarm optimization [16]. However, the inertia weight is only set to change randomly and the acceleration coefficients are changed with time, it has been observed that the performance of this modified method is similar or poor for multimodal functions.

Xueming Yang et al [17] proposes a modified particle swarm optimization algorithm with dynamic adaptation where the inertia weight is affected by the evolving state of algorithm and determined by the evolution speed factor of each particles and the aggregation degree factor of the swarm using the following two formulas:

$$h_i^t = \left| \frac{\min(F(pbest_i^{t-1}), F(pbest_i^t))}{\max(F(pbest_i^{t-1}), F(pbest_i^t))} \right| \tag{3}$$

$$s = \left| \frac{\min(F_{tbest}, \bar{F}_t)}{\max(F_{tbest}, \bar{F}_t)} \right| \tag{4}$$

where $F(pbest_i^{t-1})$ is the fitness value of best position of particle i in (t-1)th iteration $pbest_i^{t-1}$, $F_{tbest,}$ represents the optimal value in current iteration, while $\bar{F}_t$ is the mean

International Journal of Electronics Communication and Computer Engineering
Volume 6, Issue 4, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

fitness of all particle in the swarm at the t[th] iteration. So, the intertia weight can be written as the function of these two parameters, represented by

$$\omega_i^t = \omega_{ini} - \alpha(1 - h_i^t) + \beta s \quad (5)$$

where ω_{ini} is the initial value of ω, the constant α and β are chosen within the range [0,1]. The experiments have shown that the algorithm are not strongly dependent on parameters α and β, and remarkably improves the ability of PSO to jump out of the local optima and significantly enhance the convergence precision.

Similar method but distance-based is also developed. Zhan [18] adopts an evolutionary factor by using the population distribution information and relative particle fitness information in each generation. First，calculate the mean distance of particle i to all the other particles by

$$d_i = \frac{1}{N-1} \sum_{j=1, j \neq i}^{N} \sqrt{\sum_{k=1}^{D} (x_i^k - x_j^k)^2} \quad (6)$$

where N and D are the population size and problem dimension, respectively. Then, Compare all d_i's and determine the maximal distance d_{max} and the minimal distance d_{min}. Denote the global best particle of d_i by d_g. Define an *evolutionary factor F* as

$$F = \frac{d_g - d_{\min}}{d_{\max} - d_{\min}} \in [0,1] \quad (7)$$

which is set to 1 if d_{max} is equal to d_{min}, and is also initialized to 1 when the algorithm starts. Finally, the value of the ω is adaptively adjusted by the mapping $\omega(F)$

$$\omega(F) = \frac{1}{1 + 1.5e^{-2.6F}} \in [0.4, 0.9] \quad (8)$$

Note that, with the mapping function, ω now changes with F, with large value in exploration state and small value in exploitation state, but not purely with time or with the generation number. Hence, the adaptive inertia weight is expected to be changed efficiently according to the evolutionary states. The acceleration coefficients share the common attempts.

Most of these parameter setting methods focus on the overall performance of single peak function that is local search capability and convergence speed, but we can get some idea about multimodal function and niche formation from them.

B. Diversity Measure

It is known that the performance of PSO is highly related to diversity of particles, especially when attempts are made to avoid premature convergence and to escape from local optima. So, maintaining a higher diversity in PSO is a crucial task. Traditionally Diversity measures are used to analyze evolutionary algorithms. But some authors use it to guide the search behavior of particles which alternates between phase of exploiting and exploring.

Ursem [19] proposed a Diversity-Guided Evolutionary Algorithm (DGEA), which uses a diversity measure defined in (9) to alternate between exploring (mutation) and exploiting (recombination and selection) behaviors according to diversity threshold d_{low} and d_{high}.

$$diversity(P) = \frac{1}{|L| \cdot N} \sum_{i=1}^{N} \sqrt{\sum_{j=1}^{D} (x_{ij} - \bar{x}_j)^2} \quad (9)$$

where $|L|$ is the length of the diagonal in the search space, P is the swarm, N is the population size, D is the dimensionality of the problem, x_{ij} is the j[th] value of the i[th] individual, and $\bar{x}_j$ is the j[th] value of the average points. In a more general context this study shows the importance of both high and low diversity in optimization. High diversity allows the algorithm to escape local optima whereas low diversity ensures progress when fine-tuning the solutions.

Similar diversity-guided PSO, called ARPSO is also developed by Riget [20]. There defines a repulsion phase based on a modified velocity updating model which invert the velocity-update formula of the particles as in formula (10).

$$\vec{v}_i \leftarrow \chi(\vec{v}_i - U(0,\varphi_1)(\vec{p}_i - \vec{x}_i) - U(0,\varphi_2)(\vec{p}_n - \vec{x}_i)) \quad (10)$$

When the diversity of swarm drops below a predefined constant number d_{low}, the ARPSO switches to the repulsion phase, in which particles repel each other. When the diversity reaches to a high level (d_{high}), the ARPSO switches back to the attraction phase, in which particles are attracted by their pbests and the gbest. To avoid the diversity calculation and maintain the swarm diversity, Wang [21] proposes a novel diversity enhancing mechanism for PSO which is similar to the crossover operation to achieve a trade-off between exploration and exploitation abilities. When the trial particle is selected into the next generation with a predefined probability pr, the dissimilarities in the swarm will increase. A smaller pr makes larger dissimilarities, while a larger pr will decrease the diversity.

IV. Adaptive Niching

In PSO, particles move in a D-dimensional search space of possible problem solutions according to learning experience about itself and social companion. Then, the question is how to determine which particles would be suitable as neighborhood bests and how to assign them to the particles suitably so that they will move toward different optima accordingly. For the first question, majority of the existing works tried to tackle through two ways including topology technology and performance measures and discussed in section A and B. The second question, the so called Information sharing, is also a vital issue in niching since the members of the same niche need to communicate with each other and discussed in section C.

A. Topology Technique

It should be appreciated that the topological structure of the population controls its exploration versus exploitation tendencies at a low level. The arrangements of the neighborhoods can be thought of as social networks. A social network can be characterized by a series of statistics that convey some information about its structure and the speed of communication flow. The most descriptive statistics are the graph's average distance, its diameter and the distribution sequence and are given in Table 2 for some topologies [22].

Table 2 Topologies used in the study and the associated graph statistics [22]

Topology	Average distance	diameter	Distribution sequence
global	1	1	<19>
Ring	5.26	10	<2,2,2,2,2,2,2,2,2,1>
Four clusters	2.26	3	<4.6,4.8,9.6>
Pyramid	2.04	4	<5.4,7.8,5.4,0.4>
Square	2.32	4	<4,7,6,2>

Obviously, the global topology represents a fully connected graph that the information spreads quickly and the algorithm converges on single optima which the diameter is only 1. On the contrary, the ring topology represents a regular graph with a minimum number of edges between its nodes and the diameter is maximal in Table 2 which information of successful regions takes a long time to travel to the other side of the graph. This allows for different regions of the search space to be explored at the same time so as to form niches in the search space.

Xiaodong Li [23] describes a simple yet effective niching algorithm, called r2PSO, using a ring neighborhood topology, which individual particles' local memories form a stable network retaining the best positions found so far, while these particles explore the search space more broadly. Given a reasonably large population uniformly distributed in the search space, the particles search thoroughly in its local neighborhood due to the reduced convergence speed by the ring topology before propagating the information throughout the population which would result in the population converging onto a single optimum. Finally, PSO algorithms using the ring topology are able to form stable niches across different local neighborhoods, eventually locating multiple global/local optima.

Most importantly, one major advantage is that this algorithm does not require any niching parameters. This should pave the way for more widespread use of this kind of niching algorithms in real-world applications. However, the number of niches relates to the population size and this algorithm tends to generate multiple small niches. One interesting future research topic will be studying how to increase the search capability of small niches so that the performance of these niches will scale well with increasing dimensions.

B. *Measure Adoption*

The neighbors are likely to form from different areas or different niches which may oscillate between two peaks that waste the function evaluations in the topology-based neighborhood selection method, especially for multimodal function optimization. To combat this problem, T. Peram [24] proposes a measure-based neighborhood selection method, suggesting the name FDR-PSO for the algorithm, moving each particle towards nearby particles of higher fitness. The particle P_{nd} in (1) is selected by using the ratio of the relative fitness and the distance of other particles to determine the direction in which each component of the particle position needs to be changed by maximizing a FDR (Fitness-Distance-Ratio) value for the d-th dimension (assuming maximization):

$$FDR(i,j,d)=\frac{f(p_j)-f(x_i)}{\left|p_{jd}-x_{id}\right|} \tag{11}$$

x_{id} is the d[th] dimension of the i[th] current particle's position, p_j is prior best position of particle j.

However, using a particle's current position often leads to unstable and undesirable convergence behaviors since it changes more frequently than its corresponding personal best. To rectify this problem and suited to multimodal optimization, the calculation of FER（Fitness-Euclidean distance Ratio) in equation (12) is suggested using the fitness difference and the Euclidean distance between a particle's personal best and other personal bests of the particles in the population instead of FDR [25].

$$FER(j,i)=\alpha\cdot\frac{f(p_j)-f(p_i)}{\left\|\vec{p}_j-\vec{p}_i\right\|} \tag{12}$$

where α is a scaling factor, to ensure that neither fitness nor Euclidean distance becomes too dominated over one another.

In the above two algorithms, each particle is attracted towards a fittest-and-closest neighborhood point which is identified via computing its FER or FDR. FER-PSO is able to reliably locate all existing global optima over iterations, given that the population size is sufficiently large.

Instead of choosing the best neighborhood particle or requiring the user to specify the niche radius, S. Bird [26] proposes a new Adaptive Niching PSO (ANPSO) where population statistics are used to adaptively determine niche parameter at each iteration. This method uses the intrinsic nature of the particles to converge on optima and creates niches when they do so. For each member pi of the population, we measure the distance d_i to the closest other particle in the decision space using equation (13), then set r to the average of these distances, as shown in equation (14).

$$d_i=\min\{\left\|p_i-p_j\right\|;\forall j,p_i,p_j\in P\wedge p_i\neq p_j\} \tag{13}$$

$$r=\frac{\sum_{i=1}^{n}d_i}{n} \tag{14}$$

The algorithm finds the pairs of particles that are within r of each other. A counter is maintained for each pair. If the pair has been close for two or more steps, a niche is formed, while Particles will leave a niche if they are further than r from the nearest particle within their niche for 2 consecutive steps. While this algorithm requires more evaluations to locate every optimum, it does not require any niching parameter, allowing it to be very suitable for real-world tasks.

C. Information Sharing

In the canonical particle swarm, each individual has some number of neighbors according topology or measure, with a mutual influence between them. On each iteration, the individual queries its neighbors to determine which one has had the best performance so far and uses only one neighborhood best information to bias its search direction and ignores the others, where only the best neighbor has any effect, this means that the influence on the particle tends to be of a higher quality, which is not always good, since important information contained in neighborhood particles may be neglected through overemphasis on the single best neighbor, which may lead to premature or slower convergence.

This represents an oversimplification of the social-psychological view that individuals are more affected by those who are more successful, persuasive, or otherwise prestigious. In the human society, it is more accurate to say that the social neighborhood provides a wealth of possible models whose behavior may be emulated, and individuals seem to be affected by some kind of statistical summary of the state of their immediate social network rather than the unique performance of one individual. Then the question is how to choose these particles from their neighbors and how to give a particular weight according to how important or significant it is. The first question is already discussed above. The latter or the information sharing type could be classified two ways including topology and measure method in accordance with neighborhood selection method above.

Generalized the canonical PSO, Kennedy proposed the fully informed particle swarm modal FIPS [27] using the information from all other particles around it instead of single best particle, which is conceptually more concise and promises to perform better than the traditional PSO. The FIPS distributes the weight of φ in equation (15) across the entire neighborhood,

$$v_i \leftarrow x(v_i + \sum_{n=1}^{N_i} \frac{U(0,\varphi)(p_{n(n)} - x_i)}{N_i}) \qquad (15)$$

U(min, max) is a uniform random number generator, where Ni is the number of neighbors particle i has and n is i's neighbor index. If n includes only i itself and its best neighbor, then this formula degenerates to the canonical version. In the FIPS, all neighbors are a source of influence in leading the particles to fly. Therefore, the behavior of each particle is affected by its neighborhood identified by the topology used, which is able to make the whole population converge fast and more accurately. However, this information sharing type suffers the same problem, the neighbors are likely to form from different areas or different niches, in addition to the search behavior and algorithm performance is influenced greatly by the size and structure of topology.

Another approach is based on particle measure, for example distance. Kennedy described a clustering technique [28] which assigns each particle to a cluster based on distance between the current position and the cluster centre, which is selected randomly and evenly distributed topologically in the population. Then calculate the center, e.g., the mean point in vector space, for each cluster, actually, the equal weight. In order to identify the cluster centers, the method requires three iterations over all individuals in the population until centers stabilize. However, it is not necessarily the best-fit particle in that cluster. Although Kennedy's technique does not use a radius parameter, it is necessary to set the number of clusters and iterations which can be difficult since they are problem dependent. Kennedy does not report how the number of clusters affects performance.

Brits et al. in [29] proposed an nbest PSO algorithm which defines the "neighborhood" of a particle as its closest particles of all particles in the population by calculating the Euclidean distance. The neighborhood best $\hat{y}_i$ for each particle x_i is defined as the average of the positions

$$\hat{y}_i = \tfrac{1}{k} \sum_{j=1}^{k} B_{ij} \qquad (16)$$

where B_{ij} is the current position of the j[th] particle in neighborhood consisting of the k closest particles to particle x_i. Obviously, nbest PSO specifies the number of its neighbor k, while Kennedy's clustering PSO specifies the number of centers. But, the best neighbor of them all from several particles using distance measure and all suffers from the same problem that the neighborhood best is not always the best-fit particle in that neighborhood and the parameter C or k must also be prespecified by a user.

Recently, B. Y. Qu [30] modified the topology-based neighbor selection method of Kennedy [27] and proposed a distance-based locally informed particle swarm (LIPS) optimizer, which eliminates the need to specify any niching parameter and enhance the fine search ability of PSO. The velocity update of LIPS uses the formula given below while the position update keeps unchanged.

$$v_i^d = \omega \times (v_i^d + \varphi(p_i^d - X_i^d)) \qquad (17)$$

$$p_i = \frac{\sum_{j=1}^{nsize} (\varphi_j \cdot nbest_j) \big/ nsize}{\varphi} \qquad (18)$$

φ_j is a uniformly distributed random number in the range of [0, 4.1/nsize] and φ is equal to the summation of φ_j. $nbest_j$ is the j[th] nearest neighborhood to j[th] particle's pbest. nsize is dynamically increased from 2 to 5 over the function evaluations. The benefit of FIPS ensure good usage of all neighborhood information and Euclidean distance-based neighborhood selection to ensure the neighbors are from the same niche and this increases the algorithm's ability for local search and fine-tuning than topology-based one.

V. Conclusion and Future Work

Since a niching PSO searches for multiple optima in parallel, the probability of getting trapped on a local optimum may be reduced. Niching methods are of great value even when the objective is to locate a single global optimum. However, the parameter setting has been a bottleneck which makes it difficult to apply for real-world tasks. Interestingly, most of the approaches attempted to rectify some obvious drawbacks, for example, some algorithms are insensitive to the value of the parameters, allowing it to be used on a variety of problems without tuning, and some are designed to form niches adaptively without any parameter. Though niching PSO algorithms for multimodal function optimization have been researched for decade years and varieties of techniques appeared, the design and theoretical research of algorithms are far from perfect and future research on niching PSO is likely to focus on:

(1) From discussion above, the measure-based algorithm performance of existing niching algorithms is better than topology. Then choosing suitable measures such as FER, mean distance etc, which reflect the population distribution and evolution status information, is more important. If we find it, we can choose the best-fit particle to guide each particle in the swarm population and eventually niches are formed in different areas, although there are some parameters needs to setting, for example, the size of population.

(2) The kind of social interaction modeled within a PSO is used to guide particles moving toward different niches, so the information sharing type is also important. There exist two sides, including choose how many particles as its neighbor and how to assign them.

(3) From basis equation of PSO, the position and fitness of particle must be considered together. This includes algorithm design and study method, which often used by fitness-position figure.

(4) Other mechanisms can be introduced into PSO from other EAS, like differential evolution and the bees foraging. Conversely, the adaptive niching scheme may also be helpful for other EAS. Finally, it may be worthy to extend the niching frameworks to more complicated multimodal optimization scenarios including non-linear constraints, higher dimensions, and uncertain dynamic environments.

Niching methods of PSO are firstly categorized and studied in this paper. Advantages and limitations of niching methods have been pointed out and promising research directions are projected.

Acknowledgements

This work was supported by the Foundation for University Key Teacher by the Education Department of Henan Province (2013GGJS-122); Foundation of He’nan Educational Committee (13B510226).

References

[1] J.Kennedy, R.C.Eberhart, “Particle Swarm Optimization,” in Proc. of the IEEE International Conference on Neural Networks, Perth, Australia, 1995, pp. 1942–1948 .

[2] M. Clerc and J. Kennedy, “The particle swarm—explosion, stability, and convergence in a multidimensional complex space,” IEEE Trans. Evol. Comput., 2002, vol. 6, no. 1, pp. 58–73, Feb.

[3] W. Mahfoud, “Niching methods for genetic algorithms,” Ph.D. dissertation, Dept. Comput. Sci., Univ. Illinois Urbana-Champaign, Urbana, 1995.

[4] S. W. Mahfoud, “A comparison of parallel and sequential niching methods,” in Proc. 6th Int. Conf. Genet. Algorithms, Jul. 1995, pp. 136– 143.

[5] J. E. Vitela and O. Castanos, “A real-coded niching memtic algorithm for continuous multimodal function optimization,” in Proc. IEEE Congr. Evol. Computat., Jun. 2008, pp. 2170–2177.

[6] O. Mengsheol and D. Goldberg, “Probabilistic crowding: Deterministic crowding with probabilistic replacement,” in Proc. GECCO, Jul. 1999,pp. 409–416.

[7] B. Sareni and L. Krahenbuhl, “Fitness sharing and niching methods revisited,” IEEE Trans. Evol. Computat., vol. 2, no. 3, pp. 97–106, Sep.1998.

[8] J. Horn, The nature of niching: Genetic algorithms and the evolution of optimal, cooperative populations. Ph.D. thesis, Urbana,University of Illinois, Illinois, Genetic Algorithm Lab, 1997.

[9] R. Thomsen, “Multimodal optimization using crowding-based differentialevolution,” in Proc. IEEE Congr. Evol. Computat., Jun. 2004, pp.1382–1389.

[10] D. E. Goldberg and J. Richardson, “Genetic algorithms with sharing for multimodal function optimization,” in Proc. 2nd Int. Conf. Genet.Algorithms, 1987, pp. 41–49.

[11] G. R. Harik, “Finding multimodal solutions using restricted tournament selection,” in Proc. 6th Int. Conf. Genet. Algorithms, 1995, pp. 24–31.

[12] A. Petrowski, “A clearing procedure as a niching method for genetic algorithms,” in Proc. 3rd IEEE Congr. Evol. Computat., May 1996, pp.798–803.

[13] X. Li, “Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization,” in Proc. of Genetic and Evolutionary Computation Conference 2004 (GECCO’04) (LNCS 3102), 2004, pp. 105–116.

[14] R. Brits, A. Engelbrecht, and F. van den Bergh, “A Niching Particle Swarm Optimizer,” in Proc. of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL’02), 2002, volume 2, pp. 692–696.

[15] Y.Liu, X.Ling, Z.W.Shi, M.W.Lv, J.Fang, and L.Zhang, “A Survey on Particle Swarm Optimization Algorithms for Multimodal Function Optimization,” Journal of software, vol. 6, 2011, pp. 2449-2455.

[16] A.Ratnaweera, S.Halgamuge, H.Watson, “Self-organizing Hierarchical Particle Swarm Optimizer with Time-varying Acceleration Coefficients,” IEEE Trans. Evol.Comput. vol.8, 2004, pp. 240–255 .

[17] X.M.Yang, J.S Yuan, J.Y Yuan, Mao, H.N, “A modified particle swarm optimizer with dynamic adaptation,” Applied Mathematics and Computation, 2007,pp.1205–1213.

[18] Z.H.Zhan, J.Zhang, H.S-H, "Adaptive Particle Swarm Optimization," IEEE Transactions on Systems, Man, and Cybernetics ,vol.392009,pp.1362–1381.

[19] R.K. Ursem, “Diversity-guided evolutionary algorithms,” In Proc. of the Parallel Problem from Nature, vol. VII, 2002, pp. 462–471.

[20] J. Riget, J.S. Vesterstom, “A diversity-Guided Particle Swarm Optimizer – the arPSO,” Technical report, EVAlife, Denmark, 2002.

[21] W. Hui, S.Hui, C. Li, Shahryar Rahnamayan, “Diversity enhanced particle swarm optimization with neighborhood search,” Inf. Sci., 2013,pp. 119-135 .

[22] R. Mendes, J. Kennedy, J. Neves, “The fully informed particle swarm: simpler, maybe better,” IEEE Transactions on Evolutionary Computation, vol.8, 2004,pp. 204–210.

[23] X.D. Li, “Niching without niching parameters: particle swarm optimization using a ring topology,” IEEE Transactions on Evolutionary Computation, v.14 n.1, 2010,pp.150-169.

[24] T. Peram, K. Veeramachaneni, C.K. Mchan, “Fitness-distance-ratio based particle swarm optimization,” in Proceedings of the IEEE Swarm Intelligence Symposium, 2003, pp. 174–181.

[25] X.D. Li, “A multimodal particle swarm optimizer based on fitness Euclidean-distance ratio,” in Proc. of the 9th annual conference on Genetic and evolutionary computation, July 07-11, 2007, London, England.

[26] S. Bird and X. Li, “Adaptively choosing niching parameters in a PSO,” in Proc. of the 8th annual conference on Genetic and evolutionary computation, Seattle, Washington, USA, 2006, pp. 3-10.

[27] J. Kennedy and R. Mendes, “Neighborhood topologies in fully informed and best-of-neighborhood particle swarms,” IEEE Transactions on Systems, Man, and Cybernetics, Part C: Application and Reviews, vol. 36, no. 4, 2006, pp. 515–519.

[28] J. Kennedy, “Stereotyping: improving particle swarm performance with cluster analysis,” in Congress on Evolutionary Computation (CEC2000), volume 2, 2000,pp. 1507–1512.

[29] R. Brits, A. P. Engelbrecht, and F. van den Bergh, “Solving systems of unconstrained equations using particle swarm optimization,” in Proc. IEEE Conf. Syst., Man, Cybern., Hammamet, Tunisa, 2002, pp. 102–107.

[30] B.Y. Qu, Ponnuthurai Nagaratnam Suganthan, “Swagatam Das: A Distance-Based Locally Informed Particle Swarm Model for Multimodal Optimization,” IEEE Trans. Evolutionary Computation, vol.17,2013,pp. 387-402.

AUTHOR’S PROFILE

Guo Jiantao
was born in Nanyang city, Henan Province. He received the M.S. and Ph.D. degrees in signal and information processing from Dalian University of Science and Technology in 2004 and information and communication engineering from Huazhong University of Science and Technology in 2009, respectively. From 1997, He started teaching at the Xinyang Normal University where he is currently an associate Professor. His research and teaching interests are in the areas of digital signal processing, communications, and time frequency analysis.