Particle Swarm Optimization with Minimum Spanning Tree Topology for Multimodal Optimzation

Yu-Hui Zhang[1,2,3], Ying Lin[2,3,4], Yue-Jiao Gong[1,2,3,*] and Jun Zhang[1,2,3]
[1] Department of Computer Science, Sun Yat-sen University
[2] Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education
[3] Engineering Research Center of Supercomputing Engineering Software, Ministry of Education
[4] Department of Psycology, Sun Yat-sen University, Guangzhou, China
*Corresponding Author: gongyuejiao@gmail.com

***Abstract*—Multimodal optimization amounts to finding multiple optima of a problem. In recent years, particle swarm optimization (PSO) algorithms have been widely used by the evolutionary computation community to tackle multimodal problems. However, the capability of using a suitable PSO communication topology to induce stable niching behavior has not been well explored. In this paper, we propose a minimum spanning tree (MST) topology for PSO to solve multimodal problems. In each iteration, a minimum spanning tree is built based on the configuration of particles. The neighbors of each particle are determined according to the MST. The MST topology is able to capture the distribution of particles using a small number of edges. Moreover, a number of max weighted edges in the MST are cut to avoid the genetic drift phenomenon and to enhance the niching performance. The proposed topology is integrated with a canonical PSO and a locally informed particle optimizer (LIPS) to tackle multimodal problems. Experiments have been conducted on the CEC2013 benchmark functions to test the performance the integrated algorithms. Experimental results show that PSOs with MST topology are very effective in solving multimodal problems.**

I. INTRODUCTION

Multimodal problems are commonly seen in many application domains. For example, in pattern recognition, multiple eclipse detection [1][2] that aims at extracting multiple eclipse from an image is a typical multimodal problem. Locating multiple optima of a problem has several benefits. First, it can provide users a diverse set of good solutions. If a solution cannot be implemented due to unexpected circumstances, remedial actions can be made quickly by adopting other equally good solutions. Second, by locating multiple optima, the possibility of getting stuck in one local optimum is decreased. Multimodal optimization has gained increasing attention in recent years because of its considerable significance.

Evolutionary computation (EC) [3][4] is one of the most promising candidates for multimodal optimization. It contains a family of nature-inspired optimization algorithms. Unlike traditional optimization techniques, the algorithms maintain a population of individuals, which are used to search for the global optimum cooperatively. It is a common phenomenon that individuals in the population eventually gather around one optima. However, this is not desirable when we are dealing with multimodal problems. To develop the algorithms' ability of finding and maintaining multiple niches, a number of techniques known as "niching" methods were proposed [5]-[8]. Some famous niching methods include fitness sharing [9], crowding [10], speciation [11], and restricted tournament selection [12].

Among the promising EC techniques is the particle swarm optimization (PSO) [13][14]. PSO is very popular owing to its simplicity and effectiveness. It has been used increasingly as an optimization technique for solving complex problems [15]. The fact that PSO is a population-based technique makes it a natural candidate for multimodal optimization. A number of PSO variants have been proposed to handle multimodal problems [16]-[23]. Most of the niching PSOs center on dividing the particles into sub-swarms. Each sub-swarm is assigned to search for one optimum. This sort of methods generally require a niching parameter (e.g., the niche radius required by SPSO [20]), which is very difficult to set without a priori knowledge of the problem being solved. To eliminate the need of specifying niching parameters, alternative methods that rely on the swarm communication topology are proposed [22], [23]. The swarm communication topology plays a very important role in the search behavior of PSO. However, the most commonly used global topology cannot meet the requirement of multimodal optimization. In [22], Li showed that PSO with a ring topology (rpso) is able to find and maintain multiple niches. This finding sheds light on the importance of swarm topology on finding multiple optima.

This paper tries to exploit the potential of using a well-designed swarm communication topology to induce stable niching behavior. To this end, we propose a minimum spanning tree (MST) topology for PSO to solve multimodal problems. Particles are viewed as vertices in the search space. In each iteration, a fully connected graph is constructed based on the configuration of particles. Each pair of particles are connected by a weighted edge. A minimum spanning tree is built on the weighted graph. Then, the neighborhood relation of two particles is determined by whether there is an edge in the MST connecting them. Further, to avoid particles oscillating between two niches that are far away from each other, a certain number of max weight edges are removed from the MST. Hence, the graph is divided into several connected components. Particles in each connected component are responsible for finding one optimum or several optima that are

This work was supported in part by the National High-Technology Research and Development Program (863 Program) of China under Grant 2013AA01A212, in part by the National Science Fund for Distinguished Young Scholars under Grant 61125205, and in part by the National Natural Science Foundation of China under Grant 6120002 and Grant 61502542.

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.43

close to one another. The MST topology has the following features: (1) It alleviates users' burden of setting any niching parameters. (2) It is able to capture the distribution of particles by using a small number of critical edges. (3) It is a distance-based topology. The neighbors of a particle are chosen in a way that they are close to the particle geometrically. Each particle is guided by its distance-based neighbors so that multiple niches can be detected simultaneously. (4) It is able to maintain found optima until the end of a run. The MST topology uses neighborhood information to perform local search. Once a niche is detected, particles around the niche will continue to search for better solutions within it.

The proposed MST topology is integrated with the canonical PSO and a state-of-the-art PSO variant called LIPS [23] to solve multimodal problems. The resulting algorithm is termed MST-PSO and MST-LIPS respectively. Experiments have been carried out to study the search behavior of the MST-based PSOs. In addition, MST-PSO and MST-LIPS are compared with several state-of-the-art multimodal algorithms. The experimental results demonstrate the effectiveness of the MST topology in solving multimodal problems.

The rest of this paper is organized as follows. Section II gives a brief review on PSO-based multimodal algorithms. Section III describes the proposed minimum spanning tree topology PSO. Experiments on the CEC2013 benchmark functions are conducted in Section IV. Discussion about the experimental results is also included in this section. Finally, concluding remarks are presented in Section V.

II. BACKGROUND

A. Particle Swarm Optimization

PSO is a simple and efficient population-based optimization technique proposed by Kennedy and Eberhart [13]. It mimics the group behavior of bird flocking and fish schooling. PSO maintains a swarm of particles that fly through the problem space with continuously updated velocities. Particles are initially distributed throughout the search space. Each particle memorizes the best position it has ever been (*pbest*) and the global best position (*gbest*). At each iteration, the position and velocity of each particle are updated according to the following formula:

$$V_i = \omega \times V_i + c_1 \times r_1 \times (pbest_i - X_i) + c_2 \times r_2 \times (gbest - X_i) \quad (1)$$

$$X_i = X_i + V_i \quad (2)$$

where X_i and V_i denote the position and velocity of the ith particle respectively. ω is the inertia weight that is used to balance the global and local search abilities. c_1 and c_2 are acceleration coefficients. r_1 and r_2 are random numbers within [0, 1]. The model that uses *gbest* to guide the search of each particle is commonly known as global topology. Besides the global topology, there are frequently used local topologies. Instead of using *gbest* to guide the particles, a local topology defines the neighbors of each particle. The velocity of a particle is updated using the information provided by its personal best position (*pbest*) and the neighborhood best position (*lbest*), as formulized in (3).

$$V_i = \omega \times V_i + c_1 \times r_1 \times (pbest_i - X_i) + c_2 \times r_2 \times (lbest_i - X_i) \quad (3)$$

B. PSO for Multimodal Optimization

PSO is one of the most promising candidates for solving multimodal problems. Brits *et al*. [17] proposed a *nbest* PSO algorithm that defines the neighbors of a particle as its *n* closest particles. The *lbest* of a particle is the average position of its *n* neighbors. Brits *et al*. [18] also proposed a PSO variant called NichePSO. Multiple sub-swarms are spawned from the main swarm to locate multiple optima. A sub-swarm can be merged with another sub-swarm. Each sub-swarm is able to absorb particles from the main swarm. Parsopoulos and Vrahitis [19] proposed an approach that isolates particles whose fitness values have reached a specific accuracy level. The isolated particles repel other particles by using a "stretching" technique. A small swarm of particles is generated around each isolated particle to facilitate fine search. Li [20] applied the concept of species to PSO and proposed a speciation-based PSO (SPSO) algorithm. Particles in the main swarm are divided into a number of species. Each species is dominated by a particle called species seed. The velocity update of a particle is performed using the information provided by its species seed. In [16], Barrera and Coello provided a comprehensive survey for PSO-based multimodal algorithms. In the following paragraphs, we briefly review some prominent PSO-based multimodal algorithms that used for comparison in this paper.

1) Fitness Euclidean-Distance Ratio PSO (FER-PSO): Noticing that setting the niching parameters is the most critical issue for existing multimodal algorithms, Li proposed [21] a Fitness Euclidean-distance Ratio based PSO (FER-PSO) algorithm that removes the need of any niching parameters. In FER-PSO, for each pair of particles (j, i), a quantity called Fitness-Euclidean distance Ratio (FER) is calculated as follows:

$$FER_{(j,i)} = \alpha \cdot \frac{fit(pbest_j) - fit(pbest_i)}{\left\| pbest_j - pbest_i \right\|} \quad (4)$$

where α is a scaling factor. $\|pbest_j\text{-}pbest_i\|$ denotes the Euclidean distance between $pbest_j$ and $pbest_i$. Instead of having a single *gbest* for the entire swarm, each particle is allocated a neighborhood best position. $pbest_j$ is chosen as the neighborhood best position of the ith particle if $FER_{(j,\,i)}$ gives the lowest FER value. By doing so, particles will fly towards their "fittest-and-closest" neighbors and multiple sub-swarms will form naturally around multiple optima.

2) Ring Topology PSO (rpso): The performance of many multimodal algorithms is very sensitive to the setting of niching parameters. To eliminate the need of specifying any niching parameters, Li [22] proposed a ring topology PSO (rpso) for solving multimodal problems. In rpso, particles are arranged in a circle. Each particle only interacts with immediate neighbors (i.e., particles on its left and right). Li showed that PSO with ring topology is able to detect and maintain multiple stable niches.

3) Distance-Based Locally Informed Particle Swarm (LIPS): To enhance the local search ability of PSO and solve multimodal functions, Qu *et al.* [23] proposed a locally informed particle swam (LIPS) optimizer. Instead of using *gbest* to guide the search of particles, LIPS uses several *local bests*. The velocity of a particle is updated by using the information provided by its Euclidean distance neighbors, which is formulized as:

$$V_i = \omega \cdot (V_i + \varphi(P_i - X_i)) \tag{5}$$

where

$$P_i = \frac{\sum_{j=1}^{nsize} (\varphi_j \cdot nbest_j)}{\varphi} \tag{6}$$

φ_j is a random number in the range [0, 4/*nsize*] and φ is the summation of φ_j. $nbest_j$ is the *j*th nearest neighbor to $pbest_i$. *nsize* is the neighborhood size. In [23], it is recommended that *nsize* dynamically increase from 2 to 5 with respect to the number of function evaluations. LIPS makes good use of the neighborhood information and has been shown to be very effective in solving multimodal problems.

III. The Proposed Minimum Spanning Tree Topology

A. Rationale

The characteristics of PSO topologies in locating a single global optimum have been extensively studied in the literature [24]. The global topology is distinguished by its fast convergence speed, while local topologies have strong ability of diversity maintenance. However, topologies used for tracing and locating multiple optima have not been well studied. This paper is devoted to the design of a communication topology that is suitable for multimodal problems.

When solving multimodal problems, the selection of examples used to guide the search of particles is very important. If only *gbest* is used, the entire swarm will probably converge to one optimum. Different from the global topology, local topologies may be competent to handle the multimodal optimization task. In [22], Li showed that PSO with a ring topology is able to locate multiple optima of a problem. In addition, it was observed that the ring topology with no overlapping neighborhoods is more effective than that with overlapping neighborhoods (illustrated in Fig. 1). Note that ring topology is a fixed index-based topology, particles are only allowed to interact with their neighbors determined by the particle indices. As the initial swarm is generated randomly in the search space, it is very likely that a particle and its neighbors belong to different niches, causing the oscillation of particles. In a later study, Qu *et al.* [23] found that it is possible to eliminate such oscillation by defining the neighborhood relation of two particles according to their Euclidean distance. They pointed out that PSO with Euclidean-based topology is able to perform exploration more freely than those with index-based topology. Moreover, the Euclidean distance-based neighborhood selection also increases PSO's ability for local search and fine-tuning.

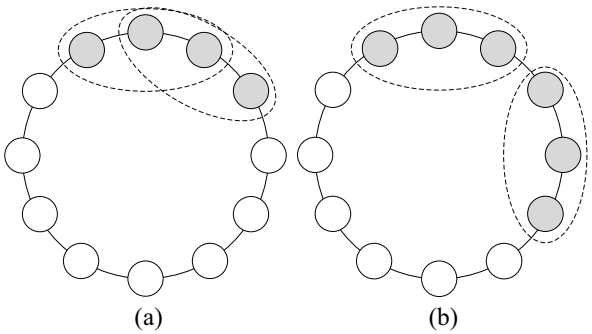

Fig. 1. Ring topology PSO. (a)r3pso with overlapping neighbors. (b) r3pso without overlapping neighbors.

Motivated by the findings that (1) topologies which divide the entire swarm into several disjoint sets of particles are able to exhibit more stable niching behavior; (2) distance-based topologies work better than index-based topologies, we propose a distance-based dynamic topology for PSO to tackle multimodal optimization problems. The detailed description of the topology is given in the next subsection.

B. Minimum Spanning Tree Topology

The proposed minimum spanning tree topology (MST) is a conceptually simple topology based on the graph theory. A graph is constructed according to the distribution of particles. Each particle is viewed as a vertex in the graph. For each pair of particles, there is a weighted edge connecting them. The weight of the edge connecting the *i*th and *j*th particles is given by the Euclidean distance:

$$w_{ij} = \left\| pbest_i - pbets_j \right\| \tag{7}$$

The distance between two particles is calculated with respect to their memory best positions, since the memory best position of a particle is more stable than its current position. We then build a minimum spanning tree on the complete graph. The max weighted edges in the spanning tree are probably connecting particles from different niches. Hence, a number of max weighted edges in the minimum spanning tree are cut to avoid possible genetic drift. In this way, the graph is divided into several connected components. Each connected component contains particles that are located in relatively close regions. The neighborhood relation of two particles is determined by whether there is an edge of the MST connecting them.

Fig. 2 gives an example of constructing the MST topology. There are seven vertices (represented by grey circles) in the graph. Every pair of distinct vertices is connected by a weighted edge (Fig. 2 (a)). The weight of an edge is determined by the Euclidean distance of its corresponding pair of particles in the search space. Then, a MST is built on the weighted complete graph, as shown in Fig. 2 (b). The longest edge connecting No. 2 and No. 4 particles is cut to facilitate the formation and maintenance of multiple niches. Each particle maintains a neighborhood list. We find the neighbors of each particle by looking through the remaining edges in the MST. The neighbors of all particles defined by the constructed MST are given in Fig. 2 (c). The velocity update of a particle is

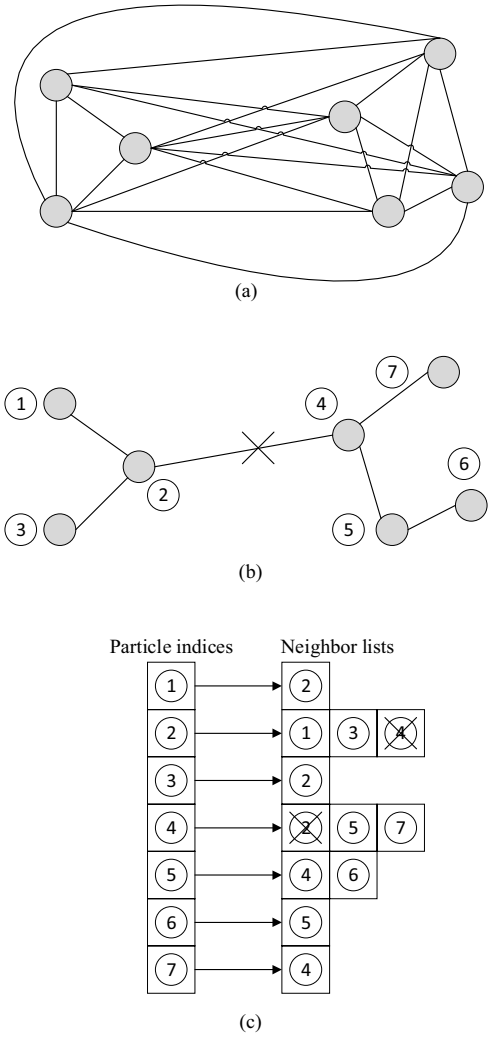

Fig. 2. Illustration of the minimum spanning tree topology.

performed by using the information provided by the neighbors in its neighborhood list.

Compared with existing PSO topologies, the MST topology has its distinct features. It is a distanced-based dynamic topology constructed from a global viewpoint. It captures the distribution of the particles with a relatively small number of critical edges. There are two main advantages of MST topology to solve multimodal problems. (1) The construction of the MST ensures the good usage of the local and global information provided by the particle distribution. Moreover, the topology is updated along with the evolution of particles, ensuring that there is no information delay. (2) The neighbors of a particle defined by the MST topology are close to each other geometrically. They are probably from the same niche and this increases the algorithm's ability of local search and fine-tuning. In addition, the removal of max weight edges increases the algorithm's ability of maintaining multiple stable niches.

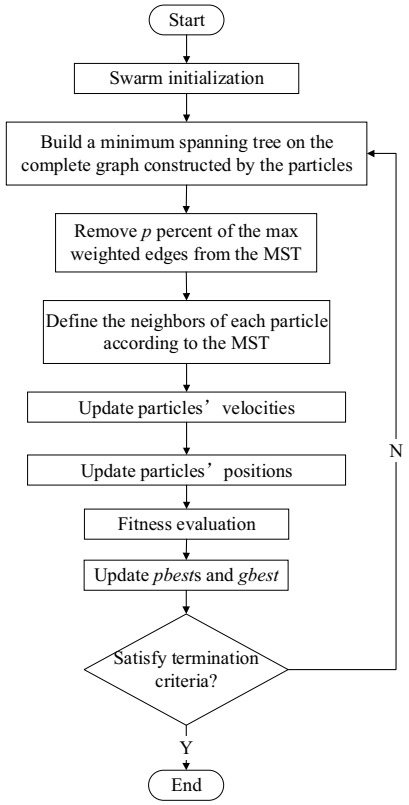

Fig. 3. Flowchart of MST- PSO.

C. Particle Swarm Optimization with MST Topology

The MST topology is integrated into the canonical PSO and the LIPS. The resulting algorithms are termed MST-PSO and MST-LIPS respectively. The flowchart of MST-based PSOs is plotted in Fig. 3. The difference between the MST-based variants and the original ones is the way in which the neighborhood of each particle is defined. Moreover, compared with LIPS, the advantages of MST-LIPS is that the parameter *nsize* is eliminated. The velocity update of MST-LIPS is given by (6), while the exemplar is changed to:

$$P_i = \frac{\sum_{j=1}^{nn_i} (\varphi_j \cdot nbest_j)}{\varphi} \tag{8}$$

where nn_i is the number of neighbors given by the MST topology. The subscript i implies that each particle may have different number of neighbors.

D. Complexity Analysis

The main computational cost of MST-based PSOs comes from the building of the minimum spanning tree. The simplest method to find the minimum spanning tree is to construct the complete graph on the particles, which has $Np(Np\text{-}1)/2$ edges, compute edge weights by finding the distance between each pair of particles, and then run Prim's algorithm on it. The total time complexity of the method is $O(Np^2)$, which is similar to

that of FER-PSO and LIPS. Therefore, the MST topology does not impose any serious burden on niching PSOs.

IV. Experiments

In this section, we carry out experiments to investigate the performance of MST-PSO and MST-LIPS.

A. Experimental Setup

1) Test Functions: The CEC2013 benchmark functions [25] are adopted to test the performance of the proposed algorithms. The function set contains 20 functions (some are identical functions with different dimensionalities). All of them are formulated as maximization problems. F_1-F_5 are simple, low dimensional functions. F_6-F_{12} are scable multimodal functions. The number of global optima for F_6 and F_7 increases rapidly as the dimensionality grows. For F_8-F_{12}, the number of global optima is independent from the dimension D. F_9-F_{12} are complex non-symmetric multimodal functions constructed by several basic functions. F_9-F_{10} and separable while F_{11}-F_{12} are non-separable. More detailed descriptions of the test functions can be found in [25].

2) Parameter Settings: An algorithm terminates when the given number of fitness evaluations (FEs) is exhausted. The maximum number of FEs for the test function set is listed in Table I. To determine whether a global optimum is found, we need to specify an accuracy level ε. In this paper, five levels of accuracy {1.0E-01, 1.0E-02, 1.0E-03, 1.0E-04, 1.0E-05} are used. For PSO algorithms, the accelarion coefficients c_1 and c_2 are fixed at 2.0. The ineria weight ω is set to 0.729 and the population size is fixed at 100. A test algorithm is run 50 times for each test function.

3) Performance Measure: Two popular performance measures, i.e., *peak ratio* (PR) and the *success rate* (SR) are adopted to evaluate the performance of MST based PSOs. PR is the percentage of the number of global optima found. SR is the percentage of runs in which all the global optima are found. They are calculated according to the following formula:

$$PR = \frac{\sum_{i=1}^{NR} NPF_i}{NKP \cdot NR}, \tag{9}$$

$$SR = \frac{NSR}{NR}, \tag{10}$$

where NPF_i is the number of found optima in the i-th run. NKP is the total number of global optima. NR is the number of runs. NSR is the number of runs in which all the global optima are successfully located.

B. Percentage of Edges to Be Cut

In MST-PSO, p percent of the longest edges are removed from the spanning tree to avoid possible genetic drift. We first investigate the effect of the percentage of removed edges. Specifically, the performance of MST-PSO is recorded when p is set to one of the following values: {0, 5%, 10%, 15%, 20%, 25%}. The experimental results are visualized in Fig. 4. It can be seen that the performance of MST-PSO is insensitive to p. Some settings of p perform slightly better than others in some functions, but the reverse is observed when dealing with other functions. Overall, the difference between the settings of p is moderate. However, if no edge is cut (i.e., p equals to 0), the possibility that all particles converge to a single optimum is increased, as revealed by the case of F_1. Hence, we recommend to remove 5%-20% of the longest edges. In the following experiments, p is fixed at 10%.

TABLE I
MAXFES USED FOR 3 RANGES OF TEST FUNCTIONS

Range of functions	MaxFEs
F_1 to F_5 (1D or 2D)	5.00E+04
F_6 to F_{11} (2D)	2.00E+05
F_6 to F_{12} (3D or higher)	4.00E+05

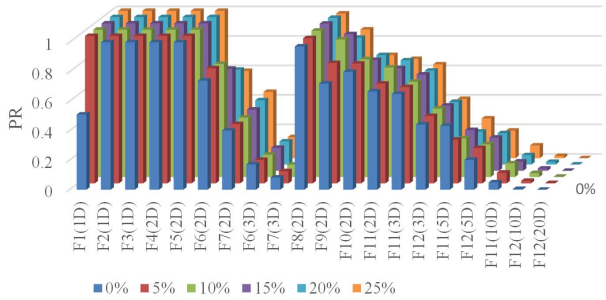

Fig. 4. Effect of the percentage of removed edges.

C. Comparison with FER-PSO and RPSO

In this part, we compare MST-PSO with FER-PSO [21] and r3pso [22]. Here, r3pso represent the version of ring topology PSO that has no overlapping neighbors (illustrated in Fig. 1(b)). The experimental results are reported in Tables II-III. The algorithms are compared pairwise and the better results are highlighted in boldface. From the tables, it can be seen that in most of the test functions, MST-PSO outperforms FER-PSO and r3pso regardless of the setting of the accuracy level. The better performance of MST-PSO owes to the way of choosing the neighbors. The distance-based neighbors defined by the MST topology ensures a high diversity. On the other hand, when particles are crowded with different regions of attractions, MST-PSO is able to perform search within each region independently. This improves MST-PSO's fine-search ability.

D. Comparison with LIPS

In this part, we compare MST-LIPS with LIPS [23] to show that the performance of LIPS can be improved by applying the MST topology. The experimental results are given in Table IV. Part of the results (ε is set to 1.0E-04) are visualized in Fig. 5. For basic multimodal functions (F_1-F_8), the performance of MST-LIPS is very similar to that of LIPS. However, for those complex functions (F_9-F_{12}), MST-LIPS outperforms LIPS. Compared with LIPS, the advantages of MST-LIPS is that the MST topology is built from a global point of view. The topology makes good use of the global and

TABLE II
EXPERIMENTAL RESULTS OF MST-PSO AND FER-PSO

Level of	1.00E-01				1.00E-02				1.00E-03				1.00E-04				1.00E-05			
accuracy	MST-PSO		FER-PSO		MST-PSO		FER-PSO		MST-PSO		FER-PSO		MST-PSO		FER-PSO		MST-PSO		FER-PSO	
Function	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_1(1D)	**1.000**	1.000	0.500	0.000	**1.000**	1.000	0.500	0.000	**1.000**	1.000	0.500	0.000	**1.000**	1.000	0.500	0.000	**1.000**	1.000	0.500	0.000
F_2(1D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	**1.000**	1.000	0.996	0.980	**1.000**	1.000	0.992	0.960
F_3(1D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_4(2D)	**1.000**	1.000	0.990	0.960	**1.000**	1.000	0.870	0.600	**1.000**	1.000	0.695	0.140	**1.000**	1.000	0.645	0.120	**1.000**	1.000	0.635	0.120
F_5(2D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	**1.000**	1.000	0.980	0.960
F_6(2D)	**0.803**	0.000	0.470	0.000	**0.791**	0.000	0.399	0.000	**0.781**	0.000	0.362	0.000	**0.769**	0.000	0.332	0.000	0.000	0.000	0.000	0.000
F_7(2D)	**0.404**	0.000	0.193	0.000	**0.404**	0.000	0.193	0.000	**0.402**	0.000	0.183	0.000	**0.399**	0.000	0.164	0.000	**0.395**	0.000	0.151	0.000
F_6(3D)	**0.170**	0.000	0.073	0.000	**0.163**	0.000	0.065	0.000	**0.160**	0.000	0.064	0.000	**0.157**	0.000	0.061	0.000	**0.155**	0.000	0.059	0.000
F_7(3D)	**0.086**	0.000	0.039	0.000	**0.085**	0.000	0.039	0.000	**0.083**	0.000	0.036	0.000	**0.081**	0.000	0.030	0.000	**0.078**	0.000	0.028	0.000
F_8(2D)	**0.985**	0.840	0.740	0.060	**0.985**	0.840	0.725	0.060	**0.985**	0.840	0.677	0.020	**0.985**	0.840	0.630	0.000	**0.983**	0.820	0.562	0.000
F_9(2D)	**0.817**	0.020	0.407	0.000	**0.817**	0.020	0.350	0.000	**0.817**	0.020	0.320	0.000	**0.817**	0.020	0.313	0.000	**0.817**	0.020	0.313	0.000
F_{10}(2D)	**0.823**	0.160	0.248	0.000	**0.820**	0.140	0.240	0.000	**0.813**	0.100	0.233	0.000	**0.810**	0.100	0.230	0.000	**0.808**	0.080	0.225	0.000
F_{11}(2D)	**0.680**	0.000	0.410	0.000	**0.677**	0.000	0.390	0.000	**0.677**	0.000	0.380	0.000	**0.673**	0.000	0.380	0.000	**0.673**	0.000	0.373	0.000
F_{11}(3D)	**0.660**	0.020	0.303	0.000	**0.653**	0.000	0.293	0.000	**0.653**	0.000	0.290	0.000	**0.653**	0.000	0.290	0.000	**0.653**	0.000	0.287	0.000
F_{12}(3D)	**0.460**	0.000	0.088	0.000	**0.458**	0.000	0.085	0.000	**0.458**	0.000	0.085	0.000	**0.458**	0.000	0.085	0.000	**0.458**	0.000	0.085	0.000
F_{11}(5D)	**0.297**	0.000	0.143	0.000	**0.297**	0.000	0.140	0.000	**0.297**	0.000	0.140	0.000	**0.297**	0.000	0.140	0.000	**0.297**	0.000	0.140	0.000
F_{12}(5D)	**0.240**	0.000	0.023	0.000	**0.240**	0.000	0.023	0.000	**0.240**	0.000	0.023	0.000	**0.238**	0.000	0.023	0.000	**0.238**	0.000	0.023	0.000
F_{11}(10D)	**0.073**	0.000	0.000	0.000	**0.073**	0.000	0.000	0.000	**0.073**	0.000	0.000	0.000	**0.073**	0.000	0.000	0.000	**0.073**	0.000	0.000	0.000
F_{12}(10D)	**0.018**	0.000	0.000	0.000	**0.018**	0.000	0.000	0.000	**0.018**	0.000	0.000	0.000	**0.018**	0.000	0.000	0.000	**0.018**	0.000	0.000	0.000
F_{12}(20D)	**0.003**	0.000	0.000	0.000	**0.003**	0.000	0.000	0.000	**0.003**	0.000	0.000	0.000	**0.003**	0.000	0.000	0.000	**0.003**	0.000	0.000	0.000

TABLE III
EXPERIMENTAL RESULTS OF MST-PSO AND R3PSO

Level of	1.00E-01				1.00E-02				1.00E-03				1.00E-04				1.00E-05			
accuracy	MST-PSO		r3pso		MST-PSO		r3pso		MST-PSO		r3pso		MST-PSO		r3pso		MST-PSO		r3pso	
Function	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_1(1D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_2(1D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_3(1D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_4(2D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_5(2D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_6(2D)	**0.803**	0.000	0.723	0.000	**0.791**	0.000	0.719	0.000	**0.781**	0.000	0.717	0.000	**0.769**	0.000	0.704	0.000	0.000	0.000	0.000	0.000
F_7(2D)	**0.404**	0.000	0.387	0.000	**0.404**	0.000	0.384	0.000	**0.402**	0.000	0.382	0.000	**0.399**	0.000	0.379	0.000	**0.395**	0.000	0.376	0.000
F_6(3D)	**0.170**	0.000	0.133	0.000	**0.163**	0.000	0.125	0.000	**0.160**	0.000	0.119	0.000	**0.157**	0.000	0.115	0.000	**0.155**	0.000	0.111	0.000
F_7(3D)	0.086	0.000	**0.103**	0.000	0.085	0.000	**0.100**	0.000	0.083	0.000	**0.095**	0.000	0.081	0.000	**0.091**	0.000	0.078	0.000	**0.087**	0.000
F_8(2D)	**0.985**	0.840	0.957	0.560	**0.985**	0.840	0.957	0.560	**0.985**	0.840	0.955	0.540	**0.985**	0.840	0.955	0.540	**0.983**	0.820	0.955	0.540
F_9(2D)	**0.817**	0.020	0.723	0.040	**0.817**	0.020	0.720	0.020	**0.817**	0.020	0.720	0.020	**0.817**	0.020	0.720	0.020	**0.817**	0.020	0.717	0.020
F_{10}(2D)	**0.823**	0.160	0.720	0.000	**0.820**	0.140	0.713	0.000	**0.813**	0.100	0.708	0.000	**0.810**	0.100	0.705	0.000	**0.808**	0.080	0.705	0.000
F_{11}(2D)	**0.680**	0.000	0.657	0.000	**0.677**	0.000	0.653	0.000	**0.677**	0.000	0.653	0.000	**0.673**	0.000	0.653	0.000	**0.673**	0.000	0.653	0.000
F_{11}(3D)	**0.660**	0.020	0.583	0.000	**0.653**	0.000	0.573	0.000	**0.653**	0.000	0.567	0.000	**0.653**	0.000	0.550	0.000	**0.653**	0.000	0.543	0.000
F_{12}(3D)	**0.460**	0.000	0.260	0.000	**0.458**	0.000	0.258	0.000	**0.458**	0.000	0.253	0.000	**0.458**	0.000	0.248	0.000	**0.458**	0.000	0.240	0.000
F_{11}(5D)	**0.297**	0.000	0.100	0.000	**0.297**	0.000	0.070	0.000	**0.297**	0.000	0.047	0.000	**0.297**	0.000	0.037	0.000	**0.297**	0.000	0.030	0.000
F_{12}(5D)	**0.240**	0.000	0.055	0.000	**0.240**	0.000	0.040	0.000	**0.240**	0.000	0.025	0.000	**0.238**	0.000	0.015	0.000	**0.238**	0.000	0.013	0.000
F_{11}(10D)	**0.073**	0.000	0.000	0.000	**0.073**	0.000	0.000	0.000	**0.073**	0.000	0.000	0.000	**0.073**	0.000	0.000	0.000	**0.073**	0.000	0.000	0.000
F_{12}(10D)	**0.018**	0.000	0.000	0.000	**0.018**	0.000	0.000	0.000	**0.018**	0.000	0.000	0.000	**0.018**	0.000	0.000	0.000	**0.018**	0.000	0.000	0.000
F_{12}(20D)	**0.003**	0.000	0.000	0.000	**0.003**	0.000	0.000	0.000	**0.003**	0.000	0.000	0.000	**0.003**	0.000	0.000	0.000	**0.003**	0.000	0.000	0.000

local information provided by the particle distribution. In LIPS, each particle only uses the information provided by its nearest neighbors without acknowledging the configuration of other particles. Therefore, MST-LIPS is more suitable for complex multimodal problems that have non-symmetric landscapes, as demonstrated in Fig. 5. Moreover, another advantage of MST-LIPS it that it eliminates the parameter *nsize* introduced by LIPS. Fig. 6 shows the particle distribution of MST-LIPS on 2-dimensional multimodal functions. The lines connecting two particles denote the edges in MST. A cross on the line indicates that the corresponding edge is removed. It can be seen that MST-LIPS is very effective in locating multiple optima.

E. Convergence Speed

We continue to study the convergence speeds of the MST-topology based PSOs. The convergence speed of a multimodal algorithm is defined by the number FEs required to locate all global optima. It is calculated according to the following formula:

$$avgFEs = \frac{\sum_{i=1}^{NR} FEs_i}{NR} \tag{11}$$

where FEs_i is the number of FEs consumed in the i-th run. If the algorithm cannot locate all global optima, FEs_i is given by the MaxFEs. In the experiment, the accuracy level is set to 1.0E-04 [25]. It is very difficult for the multimodal algorithms to generate a non-zero success rate on complex multimodal functions. Therefore, we concentrate on F_1-F_8. Table V presents the convergence speed results of the algorithms. It can be observed that in most of the considered functions, the MST topology based PSOs have faster convergence speeds,

TABLE IV
EXPERIMENTAL RESULTS OF MST-LIPS AND LIPS

Level of accuracy	1.00E-01				1.00E-02				1.00E-03				1.00E-04				1.00E-05			
	MST-LIPS		LIPS		MST-LIPS		LIPS		MST-LIPS		LIPS		MST-LIPS		LIPS		MST-LIPS		LIPS	
Function	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_1(1D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_2(1D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_3(1D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_4(2D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_5(2D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_6(2D)	**0.910**	0.140	0.772	0.000	**0.898**	**0.100**	0.766	0.000	**0.872**	0.060	0.754	0.000	**0.817**	0.040	0.733	0.000	0.000	0.000	0.000	0.000
F_7(2D)	**0.507**	0.000	0.498	0.000	**0.507**	0.000	0.498	0.000	**0.502**	0.000	0.497	0.000	0.488	0.000	**0.494**	0.000	0.454	0.000	**0.491**	0.000
F_8(3D)	**0.244**	0.000	0.224	0.000	0.205	0.000	**0.220**	0.000	0.175	0.000	**0.215**	0.000	0.146	0.000	**0.208**	0.000	0.124	0.000	**0.201**	0.000
F_9(3D)	**0.146**	0.000	0.129	0.000	**0.146**	0.000	0.129	0.000	**0.139**	0.000	0.128	0.000	0.126	0.000	**0.126**	0.000	0.110	0.000	**0.125**	0.000
F_8(2D)	0.983	0.820	**0.988**	0.860	0.983	0.820	**0.988**	0.860	0.983	0.820	**0.988**	0.860	0.982	0.800	**0.988**	0.860	0.982	0.800	**0.987**	0.840
F_9(2D)	0.827	0.060	**0.850**	0.180	0.813	0.000	**0.847**	0.160	0.807	0.000	**0.847**	0.160	0.803	0.000	**0.847**	0.160	0.800	0.000	**0.847**	0.160
F_{10}(2D)	0.823	0.140	**0.833**	0.140	0.805	0.100	**0.813**	0.120	0.788	0.060	**0.810**	0.120	0.775	0.040	**0.805**	0.100	0.760	0.020	**0.800**	0.080
F_{11}(2D)	0.670	0.000	**0.687**	0.000	0.667	0.000	**0.680**	0.000	0.667	0.000	**0.677**	0.000	0.667	0.000	**0.677**	0.000	0.667	0.000	**0.670**	0.000
F_{11}(3D)	**0.743**	0.000	0.640	0.000	**0.720**	0.000	0.640	0.000	**0.717**	0.000	0.633	0.000	**0.717**	0.000	0.633	0.000	**0.713**	0.000	0.633	0.000
F_{12}(3D)	**0.638**	0.000	0.383	0.000	**0.635**	0.000	0.383	0.000	**0.633**	0.000	0.380	0.000	**0.628**	0.000	0.380	0.000	**0.620**	0.000	0.378	0.000
F_{11}(5D)	**0.613**	0.060	0.203	0.000	**0.543**	0.000	0.200	0.000	**0.543**	0.000	0.200	0.000	**0.543**	0.000	0.200	0.000	**0.543**	0.000	0.200	0.000
F_{12}(5D)	**0.430**	0.000	0.198	0.000	**0.425**	0.000	0.188	0.000	**0.425**	0.000	0.188	0.000	**0.420**	0.000	0.188	0.000	**0.408**	0.000	0.188	0.000
F_{11}(10D)	**0.883**	**0.700**	0.047	0.000	**0.323**	0.000	0.047	0.000	**0.320**	0.000	0.047	0.000	**0.293**	0.000	0.043	0.000	**0.237**	0.000	0.040	0.000
F_{12}(10D)	**0.178**	0.000	0.005	0.000	**0.128**	0.000	0.005	0.000	**0.073**	0.000	0.003	0.000	**0.033**	0.000	0.003	0.000	**0.010**	0.000	0.003	0.000
F_{12}(20D)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

TABLE V
CONVERGENCE SPEED

Alg.		F_1(1D)	F_2(1D)	F_3(1D)	F_4(2D)	F_5(2D)	F_6(2D)	F_7(2D)	F_6(3D)	F_7(3D)	F_8(2D)
r2pso	Mean	**200.00**	1746.00	1080.00	6086.00	2776.00	200000.00	200000.00	400000.00	400000.00	81738.00
	St.D.	**0.00**	475.48	614.82	797.75	461.55	0.00	0.00	0.00	0.00	92608.27
r3pso	Mean	**200.00**	1998.00	1038.00	6070.00	2552.00	200000.00	200000.00	400000.00	400000.00	124388.00
	St.D.	**0.00**	658.63	547.32	1280.98	398.12	0.00	0.00	0.00	0.00	92661.95
LIPS	Mean	**200.00**	1672.00	1390.00	9236.00	3720.00	200000.00	200000.00	400000.00	400000.00	**31760.00**
	St.D.	**0.00**	637.51	872.53	1217.01	795.74	0.00	0.00	0.00	0.00	**49832.56**
FER-PSO	Mean	**200.00**	4992.00	956.00	42470.00	5610.00	200000.00	200000.00	400000.00	400000.00	200000.00
	St.D.	**0.00**	7844.59	516.20	14319.29	9131.53	0.00	0.00	0.00	0.00	0.00
MST-PSO	Mean	**200.00**	**1554.00**	980.00	**4770.00**	**2326.00**	200000.00	200000.00	400000.00	400000.00	42658.00
	St.D.	**0.00**	**404.58**	605.64	**395.60**	**309.07**	0.00	0.00	0.00	0.00	74059.33
MST-LIPS	Mean	**200.00**	1662.00	**804.00**	7904.00	3110.00	200000.00	200000.00	400000.00	400000.00	76472.00
	St.D.	**0.00**	378.36	**444.05**	1226.04	578.36	0.00	0.00	0.00	0.00	75423.54

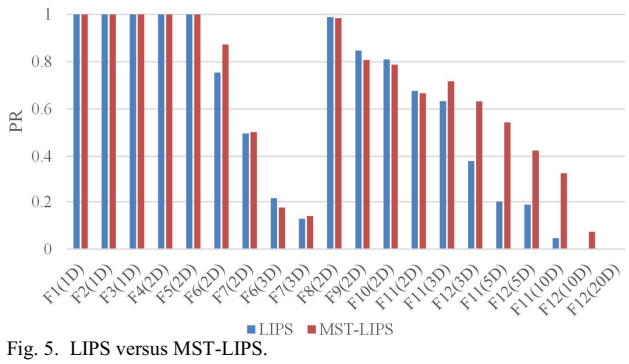

Fig. 5. LIPS versus MST-LIPS.

indicating that the MST topology can help accurately detect the basin of attractions using a relatively small number of fitness evaluations.

V. CONCLUSION

In this paper, we proposed a minimum spanning tree (MST) topology for PSO to solve multimodal problems. The proposed topology builds a minimum spanning tree on the complete graph constructed by the particles. Then, a number of max weighted edges in the minimum spanning tree are cut to enhance PSO's niching performance. The MST topology makes good use of the local and global information provided by the distribution of particles. Two MST-based PSO variants (MST-PSO and MST-LIPS) were presented. Experiments have been conducted on the CEC2013 benchmark set to study the effect of the MST topology. Experimental results show that the MST-based PSOs exhibit better performance than several state-of-the-art multimodal algorithms. The results also show that the MST-based PSOs are able to induce stable niching behavior and have faster convergence speeds. Future research may focus on developing an adaptive PSO topology that is capable of automatically enhancing the communication of particles in the same region of attraction.

REFERENCES

[1] J. Yao, N. Kharma, and P. Grogono, "Multipopulation genetic algorithm for robust and fast ellipse detection," *Pattern Anal. Applicat.*,vol.8, pp. 149–162, 2005.

[2] E. Cuevas, M. González, D. Zaldívar, and M. Pérez-Cisneros, "Multi-ellipses detection on images inspired by collective animal behavior," *Neural Computing and Applications*, vol. 24, no. 5, pp. 1019-1033, 2014.

[3] T. Back, D. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*. Oxford, U.K.: Oxford Univ. Press, 1997.

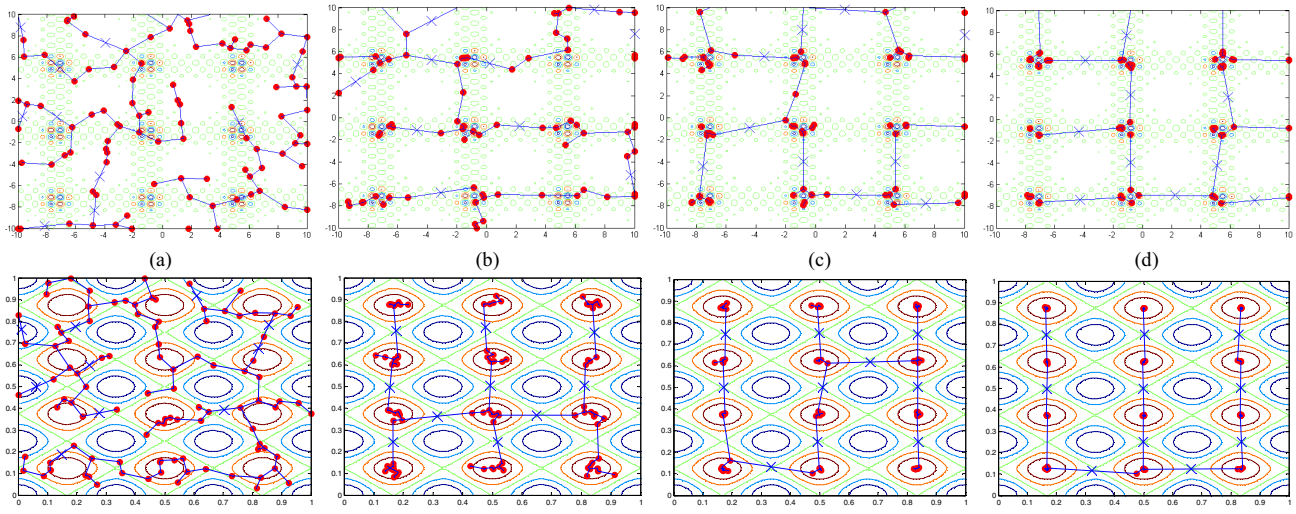

Fig. 6. Distributions of *pbests* of MST-LIPS on different stages. (a) Iteration 1 (F_6). (b) Iteration 20 (F_6). (c) Iteration 50 (F_6). (d) Iteration 100 (F_6). (e) Iteration 1 (F_8). (f) Iteration 20 (F_8). (g) Iteration 50 (F_8). (h) Iteration 100 (F_8).

[4] Y.-J. Gong, W.-N. Chen, Z.-H. Zhan, J. Zhang, Y. Li, Q. Zhang, and J.-J. Li, "Distributed Evolutionary Algorithms and Their Models: A Survey of the State-of-the-Art," *Applied Soft Computing*, vol. 34, pp. 286-300, 2015.

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

[6] O. Mengsheol and D. Goldberg, "Probabilistic crowding: Deterministic crowding with probabilistic replacement," in *Proc. Genet. Evol. Computat. Conf*, Jul. 1999, pp. 409–416

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

[8] G. Singh and K. Deb, "Comparison of multimodal optimization algorithms based on evolutionary algorithms," in *Proc. Genet. Evol. Comput. Conf.*, 2006, pp. 1305–1312

[9] 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

[10] R. Thomsen, "Multimodal optimization using crowding-based differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2004, pp. 1382–1389.

[11] A. Petrowski, "A clearing procedure as a niching method for genetic algorithms," in *Proc. 3rd IEEE Congr. Evol. Comput.*, May 1996, pp. 798–803.

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

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

[14] 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. on Cybern.*, In Press.

[15] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. G. Harley, "Particle swarm optimization: Basic concepts, variants and applications in power systems," *IEEE Trans. Evol. Comput.*, vol. 12, no. 2, pp. 171–195, Apr. 2008.

[16] J. Barrera and C. Coello, "A Review of Particle Swarm Optimization Methods Used for Multimodal Optimization," in *Innovations in Swarm Intelligence*, Springer Berlin Heidelberg, 2009, pp. 9-37.

[17] R. Brits, A. P. Engelbrecht, and F. van den Bergh, "Solving systems of unconstrained equations using particle swarm optimizers," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2002, pp. 102–107.

[18] R. Brits, A. Engelbrecht, and F. van den Bergh, "A niching particle swarm optimizer," in *Proc. 4th Asia-Pacific Conf. SEAL*, 2002, pp. 692–696.

[19] K. Parsopoulos and M. Vrahatis, "On the computation of all global minimizers through particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 211–224, Jun. 2004.

[20] X. Li, "Adaptively choosing neighborhood bests using species in a particle swarm optimizer for multimodal function optimization," in *Proc. Genet. Evol. Computat. Conf.*, vol. 3102. 2004, pp. 105–116.

[21] X. Li, "A multimodal particle swarm optimizer based on fitness Euclidean-distance ration," in *Proc. Genet. Evol. Computat. Conf.*, 2007, pp. 78–85.

[22] X. Li, "Niching without niching parameters: Particle swarm optimization using a ring topology," *IEEE Trans. Evol. Computat.*, vol. 14, no. 1, pp. 150–169, Feb. 2010.

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

[24] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," inProc. IEEE Congr. Evol. Comput., May 2002, vol. 2, pp. 1671–1676.

[25] X. Li, A. Engelbrecht, and M. G. Epitropakis, "Benchmark functions for cec'2013 special session and competition on niching methods for multimodal function optimization," Evolutionary Computation and Machine Learning Group, RMIT University, Melbourne, Australia, Tech. Rep., 2013