

# Adaptive Multimodal Continuous Ant Colony Optimization

Qiang Yang, *Student Member, IEEE*, Wei-Neng Chen, *Member, IEEE*, Zhengtao Yu, Tianlong Gu, Yun Li, *Member, IEEE*, Huaxiang Zhang, and Jun Zhang, *Senior Member, IEEE*

**Abstract**—Seeking multiple optima simultaneously, which multimodal optimization aims at, has attracted increasing attention but remains challenging. Taking advantage of ant colony optimization (ACO) algorithms in preserving high diversity, this paper intends to extend ACO algorithms to deal with multimodal optimization. First, combined with current niching methods, an adaptive multimodal continuous ACO algorithm is introduced. In this algorithm, an adaptive parameter adjustment is developed, which takes the difference among niches into consideration. Second, to accelerate convergence, a differential evolution mutation operator is alternatively utilized to build base vectors for ants to construct new solutions. Then, to enhance the exploitation, a local search scheme based on Gaussian distribution is self-adaptively performed around the seeds of niches. Together, the proposed algorithm affords a good balance between exploration and exploitation. Extensive experiments on 20 widely used benchmark multimodal functions are conducted to investigate the influence of each algorithmic component and results are compared with several state-of-the-art multimodal algorithms and winners of competitions on multimodal optimization. These comparisons demonstrate the competitive efficiency and effectiveness of the proposed algorithm, especially in dealing with complex problems with high numbers of local optima.

**Index Terms**—Ant colony optimization (ACO), multimodal optimization, multiple global optima, niching.

## I. INTRODUCTION

MULTIPLE optimal solutions, representing various designs with the same or very similar performance, are in demand in many practical applications, so that decision makers can have multiple choices [1]. To obtain multiple optima of a problem, practitioners turn their attention to population-based evolutionary algorithms (EAs), which possess potential to locate and preserve multiple optima simultaneously.

Even though different kinds of EAs [2]–[7], such as particle swarm optimization (PSO) [8]–[11], differential evolution (DE) [12]–[16], ant colony optimization (ACO) [17]–[22], and estimation of distribution algorithms (EDAs) [23]–[27], have been successfully applied to solve various problems [28]–[35], most of them focus on single optimization, rather than multimodal optimization. Owing to the global learning and updating schemes used, these EAs usually drive the whole population toward only one global optimum. Therefore, these EAs cannot be directly applied to deal with multimodal optimization. To solve multimodal problems efficiently, some special tactics should be designed to cooperate with classical EAs.

So far, the most adopted method to aid classical EAs deal with multimodal optimization is niching [36]–[43], which divides the whole population into smaller niches. Generally, each niche is responsible for seeking one or a small number of optima. Along this promising avenue, researchers have proposed various niching strategies [38]–[40], [42]–[49]. Then, through utilizing a certain niching method, a number of new updating schemes for classical EAs [47], [50]–[53] have emerged to deal with multimodal optimization. Recently, even some researchers have applied multiobjective techniques to tackle multimodal optimization [54]–[57]. The related work on these three aspects will be detailed in the following section.

In spite of the effectiveness of existing multimodal algorithms on tested problems, they are known to suffer from various drawbacks, such as inferior performance on irregular multimodal surfaces [41], the serious reliance on particular landscapes and the sensitive parameter settings [38], [39], etc. In particular, most existing multimodal algorithms would

Manuscript received November 11, 2015; revised May 8, 2016 and July 4, 2016; accepted July 6, 2016. Date of publication July 13, 2016; date of current version March 28, 2017. This work was supported in part by the National Natural Science Foundation of China under Project 61379061, Project 61332002, Project 61511130078, and Project 6141101191, in part by the Natural Science Foundation of Guangdong for Distinguished Young Scholars under Project 2015A030306024, in part by the Guangdong Special Support Program under Project 2014TQ01X550, and in part by the Guangzhou Pearl River New Star of Science and Technology under Project 201506010002 and Project 151700098. (Corresponding authors: Wei-Neng Chen and Jun Zhang.)

Q. Yang is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 51006, China, and also with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China.

W.-N. Chen and J. Zhang are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 51006, China (e-mail: cwnraul634@aliyun.com; junzhang@ieee.org).

Z. Yu is with the School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650504, China.

T. Gu is with the School of Computer Science and Engineering, Guilin University of Electronic Technology, Guilin 541004, China.

Y. Li is with the School of Engineering, University of Glasgow, Glasgow G12 8LT, U.K.

H. Zhang is with the School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China.

This paper has supplementary downloadable multimedia material available at <http://ieeexplore.ieee.org> provided by the authors.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TEVC.2016.2591064

lose efficiency when the dimensionality of multimodal problems increases [38], [39], [45]–[48], [51]–[57]. Such inferior performance may be attributed to the exponentially increasing number of optima resulted from the growth of dimensionality. Under this environment, high diversity preservation is especially important for EAs to deal with multimodal optimization.

In literature, GA [40], [42], [43], [48], [57], PSO [47], [53], [58], and DE [38], [39], [44], [46], [50]–[52], [55] are often employed to evolve the population. Although new learning or updating strategies [47], [50]–[53], have been especially developed to aid these optimizers, they still only locate a very small number of global optima, when solving complex problems with a large number of local optima. In contrast, in this paper, we take advantage of ACO algorithms in preserving high diversity to deal with multimodal optimization.

ACO [59]–[62], which is a novel nature-inspired method in evolutionary computation, is originally designed for optimizing discrete problems. Recently, Socha and Dorigo [63] have extended ACO to a continuous one named  $ACO_R$  to solve continuous problems through shifting a discrete probability distribution to a continuous one. In  $ACO_R$ , each ant constructs solutions using a Gaussian kernel function based on solutions selected probabilistically from an archive. This solution construction strategy arms  $ACO_R$  with high diversity [63], which is valuable for multimodal optimization. However,  $ACO_R$  cannot be directly utilized to locate multiple optima because the solution selection and construction strategies are based on global information, which is only fit for single optimization. As far as we know, there is no previous work on extending ACO to cope with multimodal optimization.

The above mentioned motivations stimulate the proposal of an adaptive multimodal continuous ACO (AM-ACO), for multimodal optimization in this paper. More specifically, the main characteristics of AM-ACO are as follows.

- 1) Instead of operating on the whole archive in traditional ACOs, AM-ACO operates on the niche level by incorporating niching methods, and an adaptive parameter adjusting strategy is introduced, which takes the differences among niches into consideration.
- 2) A DE mutation operator is absorbed in AM-ACO, so that the convergence speed (CS) can be accelerated.
- 3) A local search scheme based on Gaussian distribution is embedded to promote the exploitation, which is adaptively conducted around seeds of niches.

To verify the efficiency and effectiveness of the proposed AM-ACO, extensive experiments on 20 widely used benchmark multimodal functions are conducted to investigate the influence of each algorithmic component and make wide comparisons with state-of-the-art multimodal algorithms and the winners of the CEC'2013 and the CEC'2015 competitions on multimodal optimization.

Following a comprehensive review of the recent multimodal algorithms and a brief description of the related ACO algorithms in Section II, the proposed AM-ACO will be detailed in Section III. Then, a series of experiments are conducted in Section IV, to verify the efficiency and effectiveness of

the proposed algorithm. Finally, conclusions together with discussions are given in Section V.

## II. RELATED WORK

Without loss of generality, in this paper, maximization problems are taken into consideration as in [38]–[40] and [42]–[57]. In addition, this paper aims at seeking multiple global optima of a problem, which is the main focus of the current multimodal optimization researches [38]–[40], [42]–[57].

### A. Multimodal Optimization Methods

Various multimodal optimization algorithms have been put forward in recent years. To better review these respectable works, we attempt to briefly describe them in three aspects.

1) *New Niching Strategies*: Most of the current researches on multimodal optimization focus on proposing new niching strategies [38]–[40], [42]–[49]. At present, the two most fundamental and famous niching methods are crowding [39] and speciation [38]. However, these two niching strategies are sensitive to their parameters, such as the crowding size in crowding and the species radius in speciation. Therefore, to liberate the niching methods from the sensitivity to parameters, some researchers brought up parameter-free or parameter-insensitive niching strategies.

A hill-valley (HV) niching tactic [64], [65] was developed through sampling enough intermediate points within the line segment connected by two individuals to detect HVs. If there exists at least one point whose fitness is smaller than those of both individuals, then a valley is detected, indicating these two individuals belong to different niches. A drawback of this method is that enough points should be sampled so that the accurate detection can be achieved. To reduce the number of sampled points, recursive middling [57], [66] was put forward by borrowing ideas from binary search. It continuously samples the middle point of the line segment connected by two updated endpoints until the demanded point is found or the two endpoints converge to the same one. Further, a topological species conservation [48], [67] strategy was brought up by introducing a seed preservation method to avoid the extinction of some niches, which have very few individuals. Although these methods are promising in partitioning the population into niches, they usually cost a large number of fitness evaluations to perfectly detect all valleys. To circumvent this dilemma, a history-based topological speciation (HTS) method [45] was proposed through maintaining a large archive to store historical individuals, which are used to detect valleys. Though HTS can avoid costing fitness evaluations in detecting valleys, it can detect only few or even no valleys at early stages, because very few historical individuals exist in the archive.

Although the above niching strategies are promising, they encounter two limitations. First, they are either at the sacrifice of fitness evaluations [48], [57], [64]–[67], or at the sacrifice of memory space [45]. Second, such niching strategies may lead to imbalance in the number of individuals among niches. Consequently, to tackle this predicament, a clustering-based niching method [44], [46] arose. Algorithms 1 and 2 [44], [46]

**Algorithm 1** Clustering for Crowding [36]

---

**Input:** population  $P$ , cluster size  $M$   
 Step 1: Randomly generate a reference point  $R$  and compute its distance to all individuals;  
 Step 2: While  $P$  is not empty  
     Select the individual  $P_{\text{near}}$  nearest to  $R$  in  $P$  ;  
     Build a crowd by combining  $P_{\text{near}}$  and  $M-1$  individuals nearest to it;  
     Eliminate these  $M$  individuals from  $P$ ;  
 End While  
**Output:** a set of crowds

---

**Algorithm 2** Clustering for Speciation [36]

---

**Input:** population  $P$ , cluster size  $M$   
 Step 1: Sort  $P$  according to fitness;  
 Step 2: While  $P$  is not empty  
     Select the best individual  $P_{\text{best}}$  in  $P$  as a new seed;  
     Build a species containing  $P_{\text{best}}$  and  $M-1$  individuals nearest to it;  
     Eliminate these  $M$  individuals from  $P$ ;  
 End While  
**Output:** a set of species

---

display the clustering frameworks for crowding and speciation, respectively. Both methods transfer the sensitive parameter (the crowding size or the species radius) to a less sensitive parameter (the cluster size).

2) *Novel Updating Strategies for EAs:* The former researches put emphasis on the development of niching methods, with the optimizer for evolution set as the basic EAs, for instance, basic GA [40], [42], [43], [48], [57], PSO [47], [53], [58], and DE [38], [39], [44], [46], [50]–[52], [55]. However, these basic EAs may have limitations in exploring and exploiting the search space to locate all global optima [50], [53]. Therefore, taking advantage of the above mentioned niching strategies, some researchers direct their attention to proposing new update strategies for classical EAs to deal with multimodal optimization efficiently.

Li [47] proposed a ring topology-based PSO utilizing the ring topology to form stable niches across neighborhoods. Qu *et al.* [53] put forward a distance-based locally informed PSO (LIPS), which uses several local best positions to guide each particle. Then, a local informative niching DE was brought up by Biswas *et al.* [52], which introduces two different types of individual generation schemes based on selected individuals. Subsequently, they further developed an improved parent centric normalized neighborhood mutation operator for DE (PNPCDE) [51], which is then integrated with crowding [39]. In addition, utilizing speciation [38], Hui and Suganthan [50] enhanced the exploration ability of DE by applying an arithmetic recombination strategy, leading to ARSDE, which is further combined with an ensemble tactic, resulting in EARSDE. Recently, taking advantage of EDAs, Yang *et al.* [68] developed multimodal EDAs to deal with multimodal optimization.

3) *Multiobjective Techniques:* In contrast to the above mentioned researches on integrating a niching scheme with a single-objective EA to cope with multimodal optimization,

**Algorithm 3** ACO

---

While the termination criterion is not satisfied  
     *AntBasedSolutionConstruction*( );  
     *PheromoneUpdate*( );  
     *DaemonAction*( );  
 End While

---

a few approaches [54]–[57] recently have been proposed to recast multimodal optimization as a multiobjective optimization problem. This is feasible because both multimodal optimization and multiobjective optimization involve multiple optimal solutions.

Generally, the multiobjective techniques [54]–[57] transform a multimodal problem into a bi-objective problem, with the first objective to be the multimodal function itself and the second to be a self-designed function. Thus, the differences among these multiobjective methods mainly lie in the design of the second objective. In [57], the second objective is the absolute value of the gradient of the multimodal function, while in [56], it is constructed based on the norm of the gradient vector. These two algorithms require that multimodal functions are differentiable, which may not always be met in practice. Subsequently, Basak *et al.* [55] made use of the mean distance of one individual to all other individuals in the current population as the second objective, which should be maximized so that the diversity of the population can be improved. Different from the above three techniques, Wang *et al.* [54] designed a novel transformation, which not only redesigns the second objective, but also redesigns the first objective. This transformation makes the two transformed objectives conflict with each other, which matches the requirement of multiobjective optimization more.

Even though these techniques are promising for multimodal problems, especially for low dimensional ones, it becomes very difficult for them to locate global optima for problems with high dimensionality. With the dimensionality increasing, the number of local optima usually grows exponentially, which requires that optimizers should maintain considerably high diversity. This motivates us to seek for an optimizer which can preserve high diversity for multimodal optimization.

**B. Ant Colony Optimization**

ACO [59]–[61] is inspired from the foraging behavior of real ants. When ants find a food source, they will deposit pheromone trails on the ground. The amount of the pheromone deposited depends on the quantity and quality of the food, indicating the degree of attracting other ants to the food source. This indirect cooperation among ants enables them to find the shortest path between their nest and the food source [69].

Originally, ACO is designed for discrete optimization, and has been widely applied to solve real world problems [70]–[76]. The general framework of an ACO is displayed in Algorithm 3. Subsequently, Socha and Dorigo [63] extended  $ACO_R$ , through shifting a discrete probability distribution to a continuous one. The brief procedure of  $ACO_R$  is as follows.

1) *AntBasedSolutionConstruction()*: In  $ACO_R$ , the construction of new solutions by ants is accomplished in an incremental way, namely variable by variable. First, before generating a new value for a variable, each ant probabilistically selects one solution from the archive containing the already found solutions. The probability of the  $j$ th solution is calculated by

$$p_j = \frac{w_j}{\sum_{i=1}^{NP} w_i} \quad (1)$$

where NP is the archive size and  $w_j$  is the weight of the  $j$ th solution and is given by

$$w_j = \frac{1}{\sigma NP \sqrt{2\pi}} e^{-\frac{(\text{rank}(j)-1)^2}{2\sigma^2 NP^2}} \quad (2)$$

where  $\text{rank}(j)$  returns the rank of the  $j$ th solution sorted in descending order according to fitness values; and  $\sigma$  is a parameter, which has a significant effect on the weight. A small  $\sigma$  indicates that the top-ranked solutions are strongly preferred, while a large  $\sigma$  suggests a uniform probability distribution of solutions. The larger the value of  $\sigma$ , the more uniform the probability distribution [63].

Then, based on the selected solutions, an ant samples new values for variables using Gaussian distribution defined by

$$g(x^d, \mu^d, \delta^d) = \frac{1}{\delta^d \sqrt{2\pi}} e^{-\frac{(x^d - \mu^d)^2}{2(\delta^d)^2}} \quad (3)$$

where  $d$  is the dimension index and  $\delta$  is computed by

$$\delta^d = \xi \sum_{i=1}^{NP} \frac{|x_i^d - x_j^d|}{NP - 1} \quad (4)$$

where  $\xi$  is a parameter that has an effect similar to that of the pheromone persistence in the discrete ACO [59]–[61]. The higher the value of  $\xi$ , the lower the CS of the algorithm [63]. When sampling the  $d$ th dimension of a new solution,  $\mu^d$  is set as the  $d$ th dimension of the selected  $j$ th solution.

Through the above process, each ant constructs a new solution. Such random construction based on Gaussian distribution potentially equips the algorithm with high diversity, which is precious for multimodal optimization.

2) *PheromoneUpdate()*: In  $ACO_R$ , there are no apparent pheromone representation and updating strategies. Actually, these strategies are embedded into the calculation of the weight for each solution in the archive. In (2), the weight of a solution decreases exponentially with its rank [17] and in (1), this weight determines the probability of the solution chosen by ants. Thus, the weight operates as the pheromone.

Once NP new solutions have been obtained, they are added into the archive, obtaining 2NP solutions totally. Then, the NP best solutions remain as the new solutions in the archive. In this way, the search process is biased toward the best solutions found during evolution. Overall, the update of the archive plays the role of updating pheromone.

3) *DaemonAction()*: Daemon action is an optional action, which can be used to implement centralized actions [63]. Examples include the utilization of local search schemes to

refine the obtained solutions, or the collection of global information that can be used to decide whether it is useful or not to deposit additional pheromone to bias the search process. However, in the original  $ACO_R$ , no daemon action is adopted.

After  $ACO_R$ , researchers have developed other variants of continuous ACO [19], [21], [77] to deal with continuous domains and even mixed-variable problems [17], [20]. Even though a lot of attempts have been made [78]–[80], ACO is still restricted to single optimization. To the best of our knowledge, there is no work on applying ACO to deal with multimodal optimization. This observation and the considerable potential of  $ACO_R$  in preserving high diversity motivate the following work.

### III. PROPOSED ALGORITHM

In this section, taking advantage of  $ACO_R$  in preserving high diversity, we propose an AM-ACO, to deal with multimodal optimization. Furthermore, to accelerate the CS, a basic DE mutation operator is incorporated into AM-ACO. To enhance exploitation, an adaptive local search technique is further absorbed into the algorithm. At last, a random-based niche size setting strategy is developed for AM-ACO to deal with the dilemma that the niche size is problem-dependent. Particularly, each algorithmic component is detailed as follows.

#### A. Adaptive Multimodal ACO

To make ACO suitable for multimodal optimization, we first couple  $ACO_R$  with existing niching methods, resulting in multimodal ACO (M-ACO). Instead of operating on the whole solution archive in  $ACO_R$ , M-ACO operates on the niche level. Thus, before ants construct solutions, the already found solutions in the archive are partitioned into several niches according to the used niching strategies.

This paper mainly focuses on developing a new optimizer (the second aspect in Section II-A) for multimodal optimization. Thus, we directly incorporate the clustering-based niching methods [44], [46], presented in Algorithms 1 and 2, into M-ACO. Consequently, two variants of the proposed M-ACO are developed, namely crowding-based M-ACO (MC-ACO) and speciation-based M-ACO (MS-ACO).

Subsequently, we talk about one key parameter in M-ACO, namely  $\sigma$ , which makes significant difference on M-ACO and then develop an adaptive adjusting strategy for this parameter, leading to adaptive M-ACO (AM-ACO).

First, suppose the archive size is NP and the number of solutions in each niche, called niche size, is NS,<sup>1</sup> then the number of niches is  $T = NP/NS$ . Generally, NS is much smaller than NP. In this paper, for brevity, the ant colony size is set the same as the archive size and each niche is assigned to NS ants to construct NS new solutions based on AM-ACO.

<sup>1</sup>When  $NP\%NS \neq 0$ , the remaining  $NP\%NS$  solutions are set as a new niche. Thus, the number of niches is  $T = NP/NS + 1$ . However, for the convenience of description, we generally use NS to denote the number of individuals in each niche and  $T = NP/NS$  to denote the number of niches in this paper.

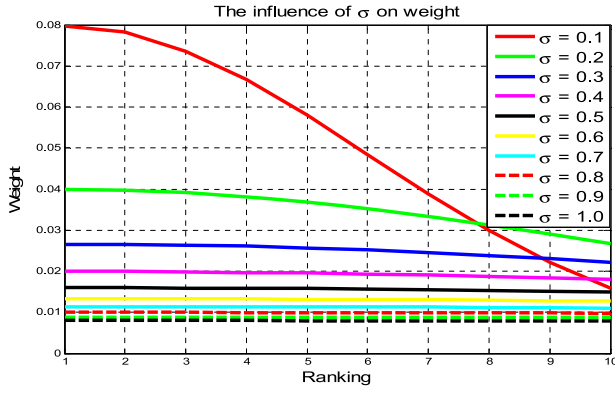


Fig. 1. Influence of  $\sigma$  on the weight of each solution.

Then, we talk about the influence of  $\sigma$  on M-ACO in detail. Through (1) and (2), we can see that  $\sigma$  plays a key role in determining the probability of each solution in the archive, and thus implicitly affects the selection of solutions for ants to construct new ones. To better understand the influence of  $\sigma$ , we plot the weight of each solution with  $\sigma$  varying from 0.1 to 1.0 and the results are presented in Fig. 1.

From this figure, we can see that the smaller the value of  $\sigma$ , the bigger the difference in the weight of each solution and the larger the value of  $\sigma$ , the more uniform the weight. In other words, a small  $\sigma$  leads to bias to the top-ranked solutions, while a large  $\sigma$  results in equivalence among solutions. In traditional  $ACO_R$  for single optimization, a small  $\sigma$ , such as  $10^{-4}$  in [63] and 0.05 in [17], is preferred. However, this is not suitable for multimodal optimization.

On one hand, it should be noticed that when locating multiple global optima simultaneously, it is highly possible that one niche may be responsible for locating a small number of global optima not just one, especially when the number of global optima is larger than that of niches. This indicates that solutions with the same or very similar fitness values in each niche should have nearly equal possibilities to be selected for ants. Therefore, in contrast to the original  $ACO_R$ , which biases to the top-ranked solutions, a large  $\sigma$  is preferred in M-ACO.

On the other hand, not all solutions in one niche are beneficial and usually the worst one should be less biased. This tells us that  $\sigma$  should not be too large, because the larger the value of  $\sigma$ , the more uniform the probability distribution.

In addition, the solution quality of different niches may be different, and the proportion of the best solutions within each niche may be different as well. This indicates that  $\sigma$  should be different for different niches.

Therefore, taking the above into consideration, we propose an adaptive adjusting strategy for  $\sigma$ , which is formulated as

$$\sigma_i = 0.1 + 0.3e^{-\frac{FS_{\max}^i - FS_{\min}^i}{FS_{\max} - FS_{\min} + \eta}} \quad (5)$$

where  $\sigma_i$  is the  $\sigma$  in (2) for the  $i$ th niche;  $FS_{\max}^i$  and  $FS_{\min}^i$  are the maximum and minimum fitness values of the  $i$ th niche, respectively;  $FS_{\max}$  and  $FS_{\min}$  are the maximum and minimum fitness values of the whole archive, respectively and  $\eta$  is a very small value used to avoid the denominator being zero.

Observing (5), we find that for each niche,  $\sigma_i$  is ranging within (0.1, 0.4]. Then, observing Fig. 1, we can conclude that, when a significant difference in solution quality exists in one niche, which is indicated by a large value of  $FS_{\max}^i - FS_{\min}^i$ ,  $\sigma_i$  tends to 0.1, leading to bias to the better solutions. This is beneficial for exploitation. On the contrary, when the fitness values of solutions in one niche are very close to each other, suggested by a small value of  $FS_{\max}^i - FS_{\min}^i$ ,  $\sigma_i$  has a tendency to 0.4, resulting in that each solution is nearly unbiased. This is profitable for exploration. Therefore, taking both the difference in solution quality of niches and that of solutions within each niche into consideration, this adaptive adjusting strategy for  $\sigma$  can potentially afford proper selections of solutions for ants to construct new ones. Through this, a good balance between exploration and exploitation can also be achieved.

After obtaining the proper  $\sigma$  for each niche, NS ants start to construct solutions using (3) and (4), where NP is replaced by NS. However, two changes should be noted in AM-ACO.

- 1) Instead of selecting one solution for each dimension in  $ACO_R$ , we use all dimensions of the selected solution as the base [namely  $\mu$  in (3)] to construct the corresponding new solution. Such operation can not only reduce the time complexity, but also potentially take the correlation among variables into consideration, which is beneficial for preserving useful information together.
- 2) As for  $\xi$  in (4), which has effects on both diversity and convergence through  $\delta$ , we set  $\xi$  as a uniformly random value generated within (0, 1] for each ant, instead of adopting a fixed value in  $ACO_R$ . The randomness of  $\xi$  is utilized because  $\sum_{i=1}^{NS} |x_i^d - x_j^d| / (NS-1)$  in AM-ACO is much smaller than  $\sum_{i=1}^{NP} |x_i^d - x_j^d| / (NP-1)$  in  $ACO_R$ . Thus,  $\xi$  may be different no matter for ants within one niche or for ants in different niches, which is potentially beneficial for obtaining a balance between exploration and exploitation.

Overall, compared with the original  $ACO_R$  [63], AM-ACO operating on the niche level is relieved from the sensitivity to parameters ( $\sigma$  and  $\xi$ ) by the adaptive adjusting strategy for  $\sigma$  and the random setting for  $\xi$ . The efficiency of AM-ACO in multimodal optimization is verified in Section IV-B.

### B. Enhancement Using DE Mutation

In AM-ACO, each ant in one niche constructs a new solution using (3) with  $\mu$  set as the selected solution, namely  $\mu = x_j$  (suppose the selected solution is  $x_j$  in the niche). Such sampling may cause slow convergence, especially when most solutions in one niche are of poor quality. In addition, when most solutions in one niche fall into local areas, it is hard for the ant colony in this niche to escape from local areas, leading to waste of fitness evaluations for useless exploration.

Therefore, to counteract such a predicament, we consider introducing a basic DE mutation operator to AM-ACO to shift the base vector [utilized in (3)] for an ant to construct solutions, which is defined as follows:

$$\mu^d = x_j^d + F(x_{\text{seed}}^d - x_j^d) \quad (6)$$

**Algorithm 4** Solution Construction for Ants

**Input:** niching size  $NS$ , niche set  $NI$ , maximum fitness  $FS_{\max}$  and minimum fitness  $FS_{\min}$  of the archive

Step 1: For the  $i$ th niche in  $NI$

- 1.1: Obtain  $FS_{\max}^i$  and  $FS_{\min}^i$  in this niche;
- 1.2: Calculate  $\sigma_i$  according to (5) for this niche;
- 1.3: Compute the probabilities of each solution in this niche according to (2) and (1);
- 1.4: For  $k=1 : NS$ 
  - 1.4.1: Randomly select a solution  $x_j$  in this niche using the roulette wheel selection method;
  - 1.4.2: If  $rand() \leq 0.5$ 
    - $\mu = x_j$
    - else
    - Compute  $\mu$  according to (6);
    - End If
  - 1.4.3: Compute  $\delta$  using (4) (note  $NP$  is replaced by  $NS$ );
  - 1.4.4: Construct a solution using (3) for an ant;

**Output:**  $NP$  new solutions and their fitness

where  $x_j = [x_j^1, \dots, x_j^d, \dots, x_j^D]$  is the selected solution for an ant with  $D$  denoting the dimension size and  $d$  representing the dimension index;  $x_{\text{seed}} = [x_{\text{seed}}^1, \dots, x_{\text{seed}}^d, \dots, x_{\text{seed}}^D]$  is the seed of the niche that  $x_j$  belongs to, and is defined as the best solution in that niche; and  $F$  is the scalar factor as in DE, but different from DE operators where a fixed  $F$  is used, the  $F$  in (6) is randomly generated within  $(0, 1]$  [81], which alleviates this operator from the sensitivity to  $F$ .

Such shifting of  $\mu$  from  $x_j$  to the updated  $\mu$  drives the ant to construct a new solution close to the promising area around the best solution in one niche. This provides ants with better chance to escape from local areas and find more promising solutions.

However, this shifting is a greedy strategy that drives ants to build solutions close to the best ones in the niches. This may result in loss of diversity. Therefore, to give a balance between diversity and convergence, we consider taking advantage of both kinds of  $\mu$  settings [ $\mu$  is set as the selected solution or set according to (6)]. Consequently, these two schemes are alternatively performed with equal probability. Additionally, the influence of this shifting strategy is observed in the experiments in Section IV-C and the experimental results demonstrate its usefulness in accelerating the convergence and the potential in locating multiple global optima.

The adaptive adjusting strategy for  $\sigma$  and the DE mutation operator are two main components in the proposed AM-ACO. Combining these two together, we deduce the framework of solution construction for ants, which is outlined in Algorithm 4.

### C. Local Search

As outlined in Algorithm 3, usually a local search method is incorporated in *DaemonAction()* to refine the obtained solutions. Likewise, in this paper, an adaptive local search is embedded into AM-ACO, leading to LAM-ACO.

Since promising solutions are generally found around the best ones and a local search scheme aims to promote the solution accuracy, we consider conducting local search around the

seed (the best solution) of each niche, so that the local search can refine solutions in different areas, avoiding convergence to one area.

In general, not all seeds fall into global areas and it would be useless to perform local search in local areas. Therefore, we further consider performing local search around seeds with probabilities, which are calculated as

$$P_i = \frac{FSE_i}{FSE_{\max}} \quad (7)$$

where  $P_i$  is the probability of the  $i$ th seed to do local search,  $FSE_i$  is the fitness of the  $i$ th seed and  $FSE_{\max}$  is the maximum fitness value among all seeds.

Further, to deal with multimodal problems with negative or zero fitness values, we extend (7) to

$$P_i = \frac{FSE_i + |FSE_{\min}| + \eta}{FSE_{\max} + |FSE_{\min}| + \eta} \quad (8)$$

where  $FSE_{\min}$  is the minimal fitness value among all seeds and  $\eta$  is a very small positive value used to avoid the denominator being zero.

Observing (7) and (8), we can obtain the following findings.

- 1) The better the fitness of one seed, the higher the probability of that seed to do local search. Thus, the local search is mainly conducted on better seeds.
- 2) For those seeds with the same fitness values, they have the same probabilities to do local search.
- 3) Local search is always performed at the best seed.

For the local search method, we propose to utilize a similar scheme used in the solution construction for ants in (3), because Gaussian distribution has a narrow sampling space, especially when the standard deviation  $\delta$  is small. In the proposed local search scheme, two modifications are made in (3). First, the mean value  $\mu$  of Gaussian distribution is set as the seed, at which the local search is performed. Second, the standard deviation  $\delta$  is set as a small value, so that better solutions can be found around the seed. In this paper, based on the preliminary experiments in the supplemental material,  $\delta = 1.0E-04$  is used.

In addition, to enhance the probability that the solution accuracy is promoted, enough points should be sampled when conducting the local search. However, the number of sampled points (termed as  $N$ ) should be neither too large nor too small. A too large number would waste fitness evaluations, especially when the local search is carried out around local areas. In contrast, a too small number may not afford the improvement of solutions. Thus, a proper number should be determined. In the preliminary experiments displayed in the supplemental material, we find  $N = 2$  is enough for AM-ACO.

Besides, since local search is only conducted at the seeds of niches, it is reasonable that we only compare the sampled points with the corresponding seeds. In this paper, we adopt a greedy strategy, namely once one sampled point is better than the seed, it replaces the seed and is used to sample the next point. Overall, the local search scheme is outlined in Algorithm 5, and the efficiency of this scheme is verified in Section IV-C.

---

**Algorithm 5** Adaptive Local Search

---

**Input:** seed set  $S$ , seed set size  $s$ , fitness values of these seeds  $FSE$ , local std value  $\delta$ , the number of sampled individuals  $N$

Step 1:  $FSE_{min} = \min(FSE)$ ,  $FSE_{max} = \max(FSE)$ , flag = false;

Step 2: If  $FSE_{min} \leq 0$   
 $FSE_{max} = FSE_{max} + |FSE_{min}| + \eta$ ;  
 flag = true;  
 End If

Step 3: For  $i = 1:s$   
 If flag  
 $Prob[i] = (FSE[i] + |FSE_{min}| + \eta) / (FSE_{max} + |FSE_{min}| + \eta)$ ;  
 else  
 $Prob[i] = FSE[i] / FSE_{max}$ ;  
 End If  
 End For

Step 4: For  $i = 1:s$   
 If  $rand() \leq Prob[i]$   
 For  $j = 1:N$   
 Generate new individual  $LS_j$  using  $Gaussian(S[i], \delta)$ ;  
 Replace  $S[i]$  with  $LS_j$ , if  $LS_j$  is better than  $S[i]$   
 End For  
 End If  
 End For

**Output:** Seeds  $S$  and their fitness  $FS$

---

*D. Random-Based Niche Size Setting for AM-ACO*

In this paper, the clustering-based crowding and speciation strategies presented in Algorithms 1 and 2 are used in AM-ACO, leading to AMC-ACO and AMS-ACO, respectively. However, Algorithms 1 and 2 have one limitation that a fixed niche size is utilized. As different problems have different features, the optimal niche size for different multimodal problems may be different. Besides, for a given problem, at different evolution stages, this niche size even may be different, since different subregions may have different fitness landscapes.

However, without any prior knowledge about the fitness landscape of a problem or a subregion of a given problem, it is difficult to determine the proper niche size for the niching strategies. To surmount this issue, a random-based niche size setting strategy is added to the niching methods, so that the sensitivity to the niche size for the used niching methods can be reduced. Specifically, during each generation, the niche size  $NS$  is randomly selected from a predefined niche size set  $G$ , which contains both small and large integers. In this way, AM-ACO can properly adapt to the fitness landscape of each problem and each subregion of a given problem.

Taking a closer observation at this niche size setting strategy, we can find that this method can potentially bring the following benefits to the proposed AM-ACO.

- 1) A potential balance between exploration and exploitation can be obtained from this strategy. During evolution, when ants in one niche fall into local areas, a larger niche size selected at the following generations would introduce more solutions for ants in niches to construct new solutions. This can potentially enhance the diversity of the niches and afford a chance for ants to escape from local areas. Thus, the exploration ability of the algorithm is enhanced. On the contrary, when the niches

---

**Algorithm 6** Local Search-Based AMC-ACO (LAMC-ACO)

---

**Input:** ant colony size  $NP$ , niching size set  $G$ , local search std  $\delta$ ,

Step 1: Randomly initialize  $NP$  solutions stored in the archive and evaluate their fitness;

Step 2: Obtain  $FS_{max}$  and  $FS_{min}$  in the archive;

Step 3: Randomly select a number from  $G$  as the niching size  $NS$ ;

Step 4: Using **Algorithm 1** to partition the archive into crowds;

Step 5: Using **Algorithm 4** to construct  $NP$  solutions;

Step 6: For each new solution  $c_k$   
 Compare  $c_k$  with the solution nearest to it in the archive and replace this solution if it is better;  
 End For

Step 7: Perform local search according to **Algorithm 5**;

Step 8: Stop if the termination criterion is met. Otherwise go to Step 2;

**Output:** the whole archive

---



---

**Algorithm 7** Local Search-Based AMS-ACO (LAMS-ACO)

---

**Input:** ant colony size  $NP$ , niching size set  $G$ , local search std  $\delta$ ,

Step 1: Randomly initialize  $NP$  solutions stored in the archive and evaluate their fitness;

Step 2: Obtain  $FS_{max}$  and  $FS_{min}$  in the archive;

Step 3: Randomly select a number from  $G$  as the niching size  $NS$ ;

Step 4: Using **Algorithm 2** to partition the archive into species;

Step 5: Using **Algorithm 4** to construct  $NP$  solutions;

Step 6: For each species  
 For each new solution  $c_k$  generated in this species  
 Compare  $c_k$  with the solution nearest to it in the species and replace this solution if it is better;  
 End For  
 End For

Step 7: Perform local search according to **Algorithm 5**;

Step 8: Stop if the termination criterion is met. Otherwise go to Step 2;

**Output:** the whole archive

---

contain too many solutions for ants to construct solutions caused by a large niche size, a smaller niche size selected at the following generations would reduce the number of solutions in the niches, potentially leading to narrow search space for ants to exploit. Thus, it may enhance the exploitation ability of AM-ACO.

- 2) This strategy can reduce the sensitivity to the niche size for the used niching methods and thus can liberate users from the tedious effort in fine tuning the optimal niche size for different problems.

In brief, this random-based niche sizing is promising for the proposed AM-ACO to deal with multimodal optimization efficiently. In the experiments in Section IV-C, the effect of this strategy on AM-ACO is observed.

*E. Complete Algorithm*

Overall, the complete frameworks of AM-ACO with the niching strategies presented in Algorithms 1 and 2 are outlined in Algorithms 6 and 7, respectively.

Generally speaking, the proposed algorithms differ from  $ACO_R$  in the following aspects.

- 1) LAM-ACOs utilize the niching methods with a random-based niche sizing scheme to seek multiple optima for multimodal optimization, while  $ACO_R$  is only to locate one global optima for single optimization.

- 2) The proposed adaptive parameter adjusting strategy takes the difference among niches and that among solutions within a niche into consideration and relieves AM-ACO from the sensitivity to parameters.
- 3) Instead of randomly selecting one solution for each dimension of a new constructed solution in  $ACO_R$ , the proposed AM-ACO only randomly selects one solution when constructing a new solution and a DE mutation operator is incorporated to improve the CS.
- 4) An adaptive local search is added to the proposed LAM-ACOs to refine the obtained solutions. Besides, the local search method is performed at the seed of each niche with self-determined probabilities.

To summarize, we can see that there are only two parameters needed to be set, namely the ant colony size NP and the less sensitive niche size set  $G$  in the proposed LAM-ACOs. This makes them relatively simple. Together, we can deduce that LAM-ACOs are promising for multimodal optimization, which will be verified in the following section.

#### IV. EXPERIMENTS AND ANALYSIS

In this section, the used multimodal benchmark functions and the evaluation criteria are first introduced in Section IV-A, following which is the investigation about the efficiency of AM-ACO in multimodal optimization in Section IV-B. Then, the influence of each algorithmic component embedded in the proposed approach is investigated in Section IV-C. In Section IV-D, wide comparisons between LAM-ACOs and several state-of-the-art multimodal algorithms are made to verify the efficiency and effectiveness of LAM-ACOs. At last, in Section IV-E, we further compare LAMS-ACO with the winners of the CEC'2013 and the CEC'2015 competitions on multimodal optimization, so that more convincing results can be obtained. It should be mentioned that due to the page limit, we attach all experimental results to the supplementary material.

##### A. Multimodal Benchmark Functions and Evaluation Criteria

In this paper, 20 benchmark multimodal functions [82] designed for the CEC'2013 special session on niching methods for multimodal optimization<sup>2</sup> are utilized to evaluate the performance of all algorithms. The characteristics of these functions are listed in Table SI in the supplemental material and readers can refer to [82] for more details.

In addition, the evaluation criteria used in both the special session and the state-of-the-art papers [44], [50]–[52], [54] are utilized to evaluate the performance of different algorithms. These criteria are peak ratio (PR), success rate (SR), and CS. Under a given accuracy level  $\varepsilon$  and a given maximum number of fitness evaluations (termed as Max\_Fes), these criteria are computed as follows [82]:

$$PR = \frac{\sum_{run=1}^{NR} NPF_i}{NKP \times NR}, SR = \frac{NSR}{NR}, CS = \frac{\sum_{run=1}^{NR} FE_i}{NR} \quad (9)$$

where  $NPF_i$  is the number of global optima found in the  $i$ th run, NKP is the number of all global optima, NR is the number

of runs, NSR is the number of successful runs and a successful run is defined as a run where all known global optima are found, and  $FE_i$  is the number of fitness evaluations used in the  $i$ th run to find all known global optima. If in a run, not all known global optima are found when the maximum number of fitness evaluations is exhausted,  $FE_i$  is set as Max\_Fes.

In this paper, five accuracy levels, namely  $\varepsilon = 1.0E-01$ ,  $\varepsilon = 1.0E-02$ ,  $\varepsilon = 1.0E-03$ ,  $\varepsilon = 1.0E-04$ , and  $\varepsilon = 1.0E-05$ , are adopted in the experiments. However, to save space, unless otherwise stated, we mainly report the results at  $\varepsilon = 1.0E-04$  as in [44], [50]–[52], and [54].

In the proposed LAM-ACOs, there are only two parameters needed to set, namely, the ant colony size (NP) and the niche size set  $G$ . To make fair comparisons, the maximum number of fitness evaluations (Max\_Fes) and the population size (NP) are set to the same for all algorithms as shown in Table I according to [46], which adopts the CEC'2013 test suite as well. As for  $G$ , it should be noticed that the niche size for the niching methods should be neither too large nor too small, because a too large niche size leads to a too wide area one niche covers, which results in that many global optima may be covered by one niche, while a too small niche size gives rise to too many niches, which may result in that many niches may fall into local areas and thus waste fitness evaluations. In this paper,  $G$  is set as a range of integers varying from 2 to 20, namely  $G = [2, 20]$ , which is enough for our algorithms according to the preliminary experiments shown in the supplementary material.

Additionally, it is worth mentioning that all results are averaged over 51 independent runs and all experiments are conducted on a PC with 4 Intel Core i5-3470 3.20 GHz CPUs, 4 Gb memory and Ubuntu 12.04 LTS 64-bit system.

##### B. Efficiency of AM-ACO in Multimodal Optimization

In this part, we mainly investigate the efficiency of AM-ACO in dealing with multimodal optimization through comparing with PSO and DE, which are often utilized in existing multimodal algorithms. For fairness, the niching strategies presented in Algorithms 1 and 2 are utilized for these three optimizers. For PSO, the local version [83], [84] is utilized instead of the global one [9], because this version can preserve higher diversity than the global one [83], [84]. When the clustering-based crowding (Algorithm 1) is combined with PSO, this algorithm is denoted by CC\_PSO. Similarly, the one with the clustering-based speciation (Algorithm 2) is denoted by CS\_PSO. For DE, Self\_CCDE, and Self\_CSDE in [44], which adopt the same niching methods are selected. While for the proposed AM-ACO, to make fair comparison, only the proposed adaptive parameter adjusting strategy is utilized.

Then, we conduct experiments on the 20 functions. Table SVI in the supplementary material shows the comparison results in PR among different algorithms at accuracy level  $\varepsilon = 1.0E-04$ . In this table, the left part of the bolded line displays the results of different algorithms with crowding, while the right part presents the results of these algorithms with speciation, and the best PRs are highlighted in bold.

From this table, obviously, we can see that AM-ACO is much better than DE and PSO in dealing with multimodal

<sup>2</sup><http://goanna.cs.rmit.edu.au/~xiaodong/cec13-niching/>



TABLE I  
SETTING OF MAXIMUM FES AND POPULATION SIZE

Function	Max Fes	Population Size
$F_1 - F_2$	5.0E+4	80
$F_4$	2.0E+5	100
$F_7$	2.0E+5	300
$F_8 - F_9$	4.0E+5	300
$F_{10}$	2.0E+5	100
$F_{11} - F_{13}$	2.0E+5	200
$F_{14} - F_{20}$	4.0E+5	200

optimization. More specifically, combined with the crowding strategy, AMC-ACO is much superior to CC\_PSO on almost all functions, except for  $F_3$  and  $F_8$  where these two algorithms achieve the same performance. Compared with Self\_CCDE, AMC-ACO shows its advantages especially on complex problems, such as  $F_{13}$ – $F_{19}$ , where many local optima exist. Combined with the speciation method, AMS-ACO demonstrates its great superiority to both CS\_PSO and Self\_CSDE, especially on complex problems as well, such as  $F_{10}$ – $F_{20}$ . Together, we can conclude that AM-ACO is much more efficient than PSO and DE in coping with multimodal optimization. The verified superiority of AM-ACO benefits from the high diversity it preserves, which results from the mechanism where the solutions are constructed in ACO.

In addition, from Table SVI in the supplementary material, we can see that both AMC-ACO and AMS-ACO achieve inferior performance on  $F_6$ – $F_9$ . This is because even though AM-ACO can preserve high diversity, it lacks of powerful exploitation ability to refine the obtained solutions. That is why a local search scheme is usually needed in ACO algorithms [59]–[62].

Overall, we can see that ACO can maintain high diversity in evolution and thus is very suitable for multimodal optimization.

### C. Observations of LAM-ACOs

1) *Influence of DE Mutation Operator and Local Search*: In this part, we investigate the influence of the DE mutation operator and the local search scheme on the proposed LAM-ACOs. First, the proposed methods without the local search scheme are denoted as AM-ACOs (AMC-ACO and AMS-ACO). Then, LAM-ACOs and AM-ACOs without (with) the DE mutation operator are denoted by attaching a tag “WDE” (“DE”). For example, LAMC-ACO-WDE means LAMC-ACO does not use the DE operator and AMC-ACO-DE indicates that AMC-ACO utilizes the DE operator.

Table SVII in the supplementary material reports the comparison results with respect to PR among these versions at accuracy level  $\varepsilon = 1.0E-04$ . Specifically, the left part of the bolded line shows the comparison results among different versions of LAMC-ACO and the right part presents the comparison results among versions of LAMS-ACO.

In terms of the DE mutation operator, we can see that from Table SVII in the supplementary material, even though on most functions the performance of AMC-ACO-WDE and LAMC-ACO-WDE is comparable to that of AMC-ACO-DE and LAMC-ACO-DE, respectively, the superiority of the DE

mutation operator is particularly evident on  $F_{20}$ . Likewise, the performance of AMS-ACO-WDE and LAMS-ACO-WDE is also very close to that of AMS-ACO-DE and LAMS-ACO-DE, respectively, and the advantage of the DE operator is especially obvious on  $F_{17}$  and  $F_{20}$ . In brief, we can see that the DE operator is helpful for the proposed LAM-ACOs, especially on complex and high dimensional problems, such as  $F_{20}$ . This is because the shifting provided by this operator in (6) can potentially afford more chances for LAM-ACOs to escape from local areas.

From the perspective of the local search scheme, observing Table SVII in the supplementary material, we can find that no matter for LAMC-ACO or LAMS-ACO the usefulness of the local search scheme is particularly evident on  $F_6$ ,  $F_8$ ,  $F_{12}$ ,  $F_{15}$ , and  $F_{17}$ , while on other functions, the local search scheme is helpful but not so obvious. The verified usefulness of the local search scheme can be ascribed to its refinement in solution accuracy.

Additionally, we further investigate the influence of the DE operator and the local search scheme on the CS of LAM-ACOs. According to the computation of CS in (9), it makes no sense to investigate CS on functions where PR is not equal to 1, namely not all global optima are found. This is because under this situation, the number of fitness evaluations used to find all global optima is set as the maximum number of fitness evaluations. Thus, we only report the comparison results with respect to CS among different versions of LAM-ACOs on  $F_1$ – $F_5$  at accuracy level  $\varepsilon = 1.0E-04$ . Table SVIII in the supplementary material presents the comparison results. In this table, the smallest CS is highlighted in bold in the left (comparisons among versions of LAMC-ACO) and right (comparisons among versions of LAMS-ACO) of the bolded line, respectively.

From this table, we can see that the DE operator is very helpful in accelerating the CS. In particular, AMC-ACO-DE and LAMC-ACO-DE use fewer fitness evaluations than AMC-ACO-WDE and LAMC-ACO-WDE on almost all the five functions, respectively. It is the same with the versions of LAMS-ACOs. In terms of the local search scheme, we find that LAM-ACOs usually cost more fitness evaluations than AM-ACOs, because the local search scheme needs extra fitness evaluations to refine the obtained solutions.

Overall, we can see that the DE operator is particularly helpful in accelerating the CS, while the local search scheme is especially useful in enhancing the exploitation ability. Both techniques are beneficial for the proposed approaches, especially on complex landscapes where massive local optima exist. In the following experiments, these two techniques are acquiescently included in LAM-ACOs.

2) *Influence of Random-Based Niche Sizing*: This part investigates the influence of the random-based niche sizing tactic added to the niching methods. First, we fix the niche size to be a member in  $\{2, 4, 8, 12, 16, 20\}$ , leading to the versions of LAM-ACOs with different fixed niche sizes. Then, we compare LAM-ACOs with the random-based niche size strategy to those with fixed niche sizes. To save space, we only report the comparison results at accuracy level  $\varepsilon = 1.0E-04$ , which are presented in Table SIX in

the supplementary material with the left part of the bolded line related to LAMC-ACO and the right part associated with LAMS-ACO.

First, observing this table, we can see that the optimal niche size for different multimodal problems is different for both LAMC-ACO and LAMS-ACO. Specifically, for functions  $F_1$ – $F_{10}$ , a small niche size (such as 2) is preferred for both algorithms. However, when it comes to  $F_{15}$ – $F_{20}$ , which have massive local optima, a large niche size (such as 12) is favourable for both algorithms. Thus, the proper niche size is hard to set without any prior knowledge about the landscapes of the functions to be optimized.

Second, from this table, we can also observe that no matter for LAMC-ACO or for LAMS-ACO, the performance of the version with random niche sizes is quite close to the one with the optimal niche size on almost all functions. In particular, for LAMS-ACO, the version with random niche sizes even performs better on some functions, such as  $F_6$ ,  $F_8$ ,  $F_{10}$ , and  $F_{20}$ .

In brief, we can observe that the random-based niche sizing strategy is helpful for LAM-ACOs, especially for LAMS-ACO. This is because this random-based niche size setting can potentially afford a balance between exploration and exploitation. In addition, this strategy also helps the proposed algorithms relieve from the sensitivity to the niche size and liberates users from tedious efforts in fine-tuning the parameter.

3) *Overall Performance of LAM-ACOs:* Following the observed influence of each algorithmic component, we investigate the overall performance of both LAMC-ACO and LAMS-ACO at all five accuracy levels. Table SX in the supplementary material presents the results of both algorithms with respect to PR and SR.

First, on functions  $F_1$ – $F_5$ , and  $F_{10}$ , both algorithms can locate all known global optima at all five accuracy levels in each run, which is indicated by that both algorithms achieve 1.0 in regard to both PR and SR on these six functions. In addition, on  $F_{14}$ ,  $F_{16}$ , and  $F_{18}$  both algorithms have the same performance with respect to PR and SR at all five accuracy levels. On these three functions, both algorithms can locate all global optima at accuracy level  $\varepsilon = 1.0E-01$ , while at the other four accuracy levels, even though both algorithms cannot locate all global optima, they locate most of the global optima (4 out of 6).

Second, at the last four accuracy levels, on  $F_6$ ,  $F_7$ , and  $F_9$ , LAMC-ACO is better than LAMS-ACO, while LAMS-ACO is significantly better than LAMC-ACO on  $F_8$ ,  $F_{11}$ ,  $F_{12}$ ,  $F_{15}$ ,  $F_{17}$ ,  $F_{19}$ , and  $F_{20}$ .

Third, on  $F_{13}$ , LAMC-ACO is a little better than LAMS-ACO at the first accuracy level, while at the following three levels, LAMS-ACO is slightly better. At the last level, they achieve the same performance.

To have a better view of comparisons between LAMC-ACO and LAMS-ACO, we plot the final landscape of the solutions stored in the archive for both algorithms. Fig. S1 in the supplementary material presents the comparison results between the two algorithms with respect to the final landscape on eight visual functions:  $F_1$ – $F_4$ ,  $F_6$ ,  $F_7$ , and  $F_{10}$ – $F_{13}$ .

From Fig. S1 in the supplementary material, two findings can be obtained.

- 1) When there is no local optima in the landscape, the solutions obtained by both algorithms are at or around the global optima [seen from Fig. S1(b), (d), (f), and (g) in the supplementary material]. Even when there are few local optima, both algorithms perform similarly, with the obtained solutions located at or around the global or local optima [seen from Fig. S1(a) and (c) in the supplementary material].
- 2) When massive local optima exist, these two algorithms perform very differently. For LAMC-ACO, the obtained solutions locate at or around the global optima and the local optima, while the solutions obtained by LAMS-ACO mainly locate at or around the global optima [seen from Fig. S1(i) and (j) in the supplementary material]. This difference may result from the difference in the niching methods and that in the archive updating strategies used in these two algorithms.

In summary, we can see that both algorithms are promising for multimodal optimization and in general, LAMS-ACO is slightly better than LAMC-ACO in locating the global optima.

#### D. Comparisons With State-of-the-Art Multimodal Algorithms

In this section, we make comparisons between LAM-ACOs (LAMC-ACO and LAMS-ACO) and several state-of-the-art multimodal algorithms. These algorithms include: 1) CDE [39], SDE [38], Self\_CCDE [44], Self\_CSDE [44], NCDE [46], and NSDE [46], which concentrate on the first aspect in dealing with multimodal optimization as stated in Section II-A; 2) LIPS [53], R2PSO [47], LoICDE [52], LoISDE [52], and PNPCCDE [51], which focus on the second aspect; and 3) MOMMOP [54], which is the latest multiobjective algorithm focusing on the third aspect in coping with multimodal optimization. The brief description of these algorithms can be found in Section II-A. To make fair comparisons, the population size and the maximum number of fitness evaluations are set as shown in Table I according to [46], which adopts the CEC'2013 test suite as well. Other parameters introduced in the corresponding algorithms are set as recommended in the related papers.

To save space, we leave all the comparison results at the five accuracy levels in the supplementary material. Tables SXI–SXXV in the supplementary material show the comparison results with respect to PR, SR, and CS of different multimodal algorithms at the five accuracy levels, and the best PRs are highlighted in bold. The row “bprs” counts the number of functions where one algorithm obtains the best PR results, namely the number of the bolded PRs. Table SXVI in the supplementary material presents the change of bprs of different algorithms with the accuracy level increasing. Additionally, Table SXVII in the supplementary material shows the comparison results with regard to CS between LAM-ACOs and 6 algorithms on  $F_1$ – $F_5$  at all accuracy levels. Tables SXVIII–SXXXII in the supplementary material present

nonparametric Wilcoxon rank-sum test<sup>3</sup> results with respect to PR between LAM-ACOs and the state-of-the-art methods at the five accuracy levels. In these tables, each compared algorithm is associated with two columns, of which the left one is the results compared with LAMC-ACO and the right one is the results compared with LAMS-ACO. In addition, the critical value of Wilcoxon rank-sum test with respect to the rank sum for 51 samples is 2873, which is computed according to [85] and displayed in the supplemental material. Therefore, the number larger than 2873 in the grayed unit in these five tables indicates that our algorithm is significantly better than the compared algorithm and the number smaller than 2380 highlighted in bold indicates our algorithm is significantly worse. The other cases mean our algorithm is equivalent to the compared algorithm. On the basis of these, the last row (*w/t/l*) of these tables counts the number of functions on which LAMC-ACO or LAMS-ACO significantly wins, ties, and significantly loses the competitions when compared with the corresponding counterparts. Further, Table SXXIII in the supplementary material presents the change of “*w/t/l*” of LAMC-ACO and LAMS-ACO in comparison to the compared algorithms with the accuracy level increasing.

From Tables SXI–SXXIII in the supplementary material, we can draw four conclusions.

- 1) From Tables SXI–SXV in the supplementary material, in terms of the number of the best PRs, namely *bprs*, we can see that with the accuracy level increasing, the superiority of the proposed methods becomes more and more evident as revealed by Table SXVI in the supplementary material. In detail, when the accuracy level increases from the first level to the last level, the *bprs* of CDE, SDE, LIPS, R2PSO, NCDE, NSDE, Self\_CCDE, Self\_CSDE, LoICDE, LoISDE, PNPCCDE, and MOMMOP gradually decreases from 12, 5, 6, 8, 14, 8, 12, 7, 15, 8, 13 and 17 to 5, 1, 4, 4, 7, 2, 5, 1, 6, 2, 5 and 10 and that of LAMC-ACO decreases from 15 to 9, while the *bprs* of LAMS-ACO first increases from 11 to 15 and then decreases to 14. In particular, at the last level, both LAMC-ACO and LAMS-ACO are significantly better than the compared algorithms except for MOMMOP. Additionally, both algorithms are particularly better than SDE, NSDE, Self\_CSDE, and LoISDE. Such observations with respect to *bprs* demonstrate the superiority of the proposed LAM-ACOs, especially LAMS-ACO.
- 2) Further, inspecting Tables SXI–SXV in the supplementary material, we see that with regard to PR, LAMS-ACO always achieves the best PRs on  $F_{15}$ – $F_{20}$  from the second level to the last level. In addition, LAMC-ACO also performs significantly better than the compared methods at the last four levels on these functions in terms of PR. Since these functions are more complex because of the existence of massive local optima, we can conclude that both LAMC-ACO and LAMS-ACO are better at handling complex problems than the compared multimodal algorithms, especially for LAMS-ACO.

- 3) With regard to the comparison in CS, in Table SXVII in the supplementary material, we only compare LAMC-ACO and LAMS-ACO with CDE, NCDE, Self\_CCDE, LoICDE, PNPCCDE, and MOMMOP on  $F_1$ – $F_5$ , because almost all these algorithms can achieve 1.0 for both PR and SR at all five accuracy levels on these functions. The comparison results are presented in Table SXVII in the supplementary material, with the first number in each unit indicating the number of the functions where LAMC-ACO achieves faster convergence, and the second number suggesting the number of the functions where LAMS-ACO obtains a smaller CS. From this table, we can see that as the accuracy level increases, the advantage of both LAMC-ACO and LAMS-ACO in achieving a faster CS becomes more and more evident. Particularly, at the last three accuracy levels, both LAMC-ACO and LAMS-ACO converge faster than Self\_CCDE on all the five functions, and faster than CDE, LoICDE, PNPCCDE, and MOMMOP on four functions. This observation verifies that the proposed LAM-ACOs have potential in maintaining a fast CS in locating all global optima.

- 4) Observing Tables SXVIII–SXXII in the supplementary material, from the perspective of the Wilcoxon rank sum test results, we can also see that with the accuracy level increasing, the superiority of both LAMC-ACO and LAMS-ACO becomes more and more obvious as clearly seen from Table SXXIII in the supplementary material. As the accuracy level increases, the number of the functions where LAMC-ACO or LAMS-ACO significantly wins the competitions gradually increases. In addition, we can see that both LAMC-ACO and LAMS-ACO are better than SDE, NSDE, and LoISDE on almost all functions, except for one or two functions where they tie the competitions.

Additionally, to observe the evolution behaviors of different algorithms, we plot the changes of the number of found global optima as the evolution process goes, which is shown in Fig. S2 in the supplementary material. In this experiment, only MOMMOP, PNPCCDE, LoICDE, Self\_CCDE, and NCDE are selected to compare with LAMC-ACO and LAMS-ACO, because they are comparable to the proposed algorithms, as indicated by the above mentioned experimental results.

From Fig. S2 in the supplementary material, on one hand, with respect to the number of found global optima, we can obviously see that the proposed LAMC-ACO or LAMS-ACO is much better than the compared five algorithms, especially on  $F_{11}$ ,  $F_{12}$ ,  $F_{15}$ , and  $F_{17}$ – $F_{20}$ . In particular, the superiority of LAMS-ACO is much more obvious on these functions. On the other hand, with regard to the CS, we can also notice that LAMC-ACO and LAMS-ACO can preserve a competitive or even faster CS. Specifically, on  $F_{11}$ – $F_{20}$ , LAMS-ACO converges much faster than all the compared algorithms.

Overall, the above comparison results indicate the consistent and statistically validated superiority of the proposed LAM-ACOs in handling multimodal optimization. This verified superiority is mainly attributed to the good balance between exploration and exploitation, which mainly benefits

<sup>3</sup>[http://en.wikipedia.org/wiki/Mann-Whitney\\_U\\_test](http://en.wikipedia.org/wiki/Mann-Whitney_U_test)

from three techniques: 1) the proposed adaptive parameter adjusting strategy; 2) the alternatively utilized base vector generation scheme; and 3) the adaptive local search scheme. The first technique takes the difference in solution quality among niches and within niches into consideration to determine a proper parameter for AM-ACO. The second technique incorporates a DE mutation operator and alternatively generates the base vectors for ants to construct new solutions, which is beneficial for ants to move fast to the promising areas and potentially helps ants escape from local areas. The third one is self-adaptively performed around seeds of niches to refine the obtained solutions, which is profitable for exploitation. In addition, the randomized number of niches may also offer a balance between exploration and exploitation. When ants fall into local areas, a large niche size may be selected to provide more choices for ants to construct new solutions, which may enhance the exploration. When ants are exploiting the promising areas, a small niche may be generated to narrow the search range, which may be beneficial for exploitation. Together, these techniques give rise to the competitive efficiency and effectiveness of the proposed LAM-ACOs.

#### E. Comparisons With Winners of CEC Competitions

In the above experiments, LAMS-ACO has presented its great dominance to LAMC-ACO and both are significantly superior to the state-of-the-art multimodal algorithms. To comprehensively demonstrate the effective performance of LAMS-ACO, we further compare it with the winners of the CEC'2013 and the CEC'2015 competitions on multimodal optimization, which are niching the CMA-ES via nearest-better clustering (NEA2) [86] and Niching migratory multi-swarm optimizer (NMMSO) [87], respectively. For simplicity, we directly cite the results of these two winners from the corresponding competitions (CEC'2013<sup>4</sup> and CEC'2015).<sup>5</sup>

Tables SXXIV–SXXVIII in the supplementary material present the comparison results with respect to PR and SR between LAMS-ACO and these two winners with each table associated with one accuracy level. The best PR results are highlighted in bold in these tables and the last row (*b/e/w*)<sup>6</sup> of these tables counts the number of functions on which LAMS-ACO is better than, equivalent to or worse than the compared winner, respectively. Please note that due to the absence of the detailed results in the associated competitions, whether LAMS-ACO is better than, equivalent to or worse than the compared winner is just determined by the values of PR without any statistical test validation, in these tables.

From Tables SXXIV–SXXVIII in the supplementary material, we can get the following findings.

<sup>4</sup><https://github.com/mikeagn/CEC2013/tree/master/NichingCompetition2013FinalData>

<sup>5</sup><https://github.com/mikeagn/CEC2013/tree/master/NichingCompetition2015FinalData>

<sup>6</sup>In this experiment, owing to the absence of the detailed results of these two winners in each run in the corresponding competitions, whether LAMS-ACO is better than, equivalent to or worse than the compared winners is judged just by the averaged PR results without any statistical test analysis. Thus, to tell apart from the results in the last section, the number of the functions on which LAMS-ACO is better than, equivalent to or worse than the compared winner, is respectively, denoted by “*b/e/w*” instead of “*w/t/l*”.

- 1) First, at the first accuracy level  $\varepsilon = 1.0E-01$ , LAMS-ACO is much better than both NEA2 and NMMSO. More specifically, at this level, LAMS-ACO dominates NEA2 and NMMSO on 13 and 9 functions, respectively, and is only dominated by these two winners on 1 and 4 functions, respectively. Particularly, LAMS-ACO is much better than NEA2 and NMMSO on  $F_{14}$ – $F_{20}$ , where massive local optima exist.
- 2) Second, at the last four accuracy levels, LAMS-ACO consistently remains its competitive performance compared with these two winners. More concretely, even though the number of the functions (6 or 7) where LAMS-ACO dominates NEA2 or NMMSO is one or two smaller than that of those functions (7 or 8) where LAMS-ACO is dominated by NEA2 or NMMSO, LAMS-ACO can achieve very similar performance to NEA2 and NMMSO on most of those functions where LAMS-ACO is dominated by NEA2 or NMMSO. For example, with respect to PR at the last accuracy level  $\varepsilon = 1.0E-05$ , on  $F_{16}$ ,  $F_{17}$ , and  $F_{20}$ , LAMS-ACO achieves 0.667, 0.625, and 0.333, respectively, which is very similar to NEA2 with 0.673, 0.695, and 0.350, respectively. On  $F_{11}$ , and  $F_{12}$ , LAMS-ACO obtains 0.944 and 0.980, respectively, which is very close to NMMSO with 0.990 and 0.990, respectively. However, at the last accuracy level (Table SXXVIII in the supplementary material), we find that LAMS-ACO performs much better than NEA2 on  $F_6$  and  $F_8$ , and much better than NMMSO on  $F_6$  and  $F_{15}$ – $F_{20}$ . Particularly, on  $F_6$ , LAMS-ACO can locate almost all the known global optima with 0.990 for PR and 0.824 for SR, while both NEA2 and NMMSO cannot locate any global optima with 0.000 for PR and 0.000 for SR at the last accuracy level.
- 3) Third, compared with NMMSO, both LAMS-ACO and NEA2 are better at handling complicated problems with massive local optima, such as  $F_{15}$ – $F_{20}$ . It can be obviously seen that LAMS-ACO and NEA2 achieve similar performance on  $F_{15}$ – $F_{20}$  and both are much better than NMMSO.

Overall, we can see that LAMS-ACO is competitive to the winners of the CEC'2013 and the CEC'2015 competitions.

## V. CONCLUSION

Taking advantage of  $ACO_R$  in preserving high diversity, this paper has proposed an AM-ACO algorithm to deal with multimodal optimization. First, combined with clustering-based crowding and speciation, AM-ACO incorporates an adaptive parameter adjusting strategy, which takes the difference between niches and that between solutions within a niche into consideration. Subsequently, to enhance the CS, a DE mutation operator is added to alternatively build base vectors for ants to construct new solutions. Besides, a local search scheme based on Gaussian distribution is added to refine solutions, leading to LAM-ACOs. At last, to relieve the algorithm from the sensitivity to the niche size, a random-based niche size setting strategy is further added to the used niching methods.

Together, these techniques give rise to a good balance between exploration and exploitation, leading to competitive efficiency and effectiveness of the proposed LAM-ACOs.

In the experiments, the efficiency of AM-ACO only with the first technique in dealing with multimodal optimization was first investigated and the usefulness of each algorithmic component was also observed. The comparison results between LAM-ACOs (LAMC-ACO and LAMS-ACO) and several state-of-the-art multimodal algorithms and the winners of two CEC competitions on multimodal optimization reveal the superiority of the proposed LAM-ACOs. In particular, the superiority of LAM-ACOs becomes more and more evident with the accuracy level increasing. In addition, the comparison results also demonstrate the better competence of LAM-ACOs (especially LAMS-ACO) in handling complex problems in comparison to the state-of-the-art algorithms.

Even though LAM-ACOs show their potential in dealing with multimodal optimization, especially on complex problems, they still cannot locate all the known global optima when a large number of local optima exist. Therefore, there is room to further improve the performance of the proposed algorithms on very complex problems, which forms a part of future work.

#### REFERENCES

- [1] K. Deb and A. Srinivasan, "Innovation: Innovating design principles through optimization," in *Proc. Conf. Genet. Evol. Comput.*, Seattle, WA, USA, 2006, pp. 1629–1636.
- [2] T. Liao, D. Aydin, and T. Stützle, "Artificial bee colonies for continuous optimization: Experimental analysis and improvements," *Swarm Intell.*, vol. 7, no. 4, pp. 327–356, 2013.
- [3] T. Liao, D. Molina, and T. Stützle, "Performance evaluation of automatically tuned continuous optimizers on different benchmark sets," *Appl. Soft Comput.*, vol. 27, pp. 490–503, Feb. 2015.
- [4] N. Chen *et al.*, "An evolutionary algorithm with double-level archives for multiobjective optimization," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1851–1863, Sep. 2015.
- [5] Y.-C. Lin, M. Clauß, and M. Middendorf, "Simple probabilistic population-based optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 245–262, Apr. 2016.
- [6] G. Karafotias, M. Hoogendoorn, and A. E. Eiben, "Parameter control in evolutionary algorithms: Trends and challenges," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 167–187, Apr. 2015.
- [7] M. A. Muñoz, M. Kirley, and S. K. Halgamuge, "Exploratory landscape analysis of continuous space optimization problems using information content," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 74–87, Feb. 2015.
- [8] M. Campos, R. A. Krohling, and I. Enriquez, "Bare bones particle swarm optimization with scale matrix adaptation," *IEEE Trans. Cybern.*, vol. 44, no. 9, pp. 1567–1578, Sep. 2014.
- [9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4, Perth, WA, Australia, 1995, pp. 1942–1948.
- [10] W.-N. Chen *et al.*, "A novel set-based particle swarm optimization method for discrete optimization problems," *IEEE Trans. Evol. Comput.*, vol. 14, no. 2, pp. 278–300, Apr. 2010.
- [11] W.-N. Chen *et al.*, "Particle swarm optimization with an aging leader and challengers," *IEEE Trans. Evol. Comput.*, vol. 17, no. 2, pp. 241–258, Apr. 2013.
- [12] R. A. Sarker, S. M. Elsayed, and T. Ray, "Differential evolution with dynamic parameters selection for optimization problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 5, pp. 689–707, Oct. 2014.
- [13] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.
- [14] S.-Y. Park and J.-J. Lee, "Stochastic opposition-based learning using a beta distribution in differential evolution," *IEEE Trans. Cybern.*, in press, 2016.
- [15] N. M. Hamza, D. L. Essam, and R. A. Sarker, "Constraint consensus mutation-based differential evolution for constrained optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 447–459, Jun. 2016.
- [16] S. Das, A. Mandal, and R. Mukherjee, "An adaptive differential evolution algorithm for global optimization in dynamic environments," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 966–978, Jun. 2014.
- [17] T. Liao, K. Socha, M. A. M. de Oca, T. Stützle, and M. Dorigo, "Ant colony optimization for mixed-variable optimization problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 503–518, Aug. 2014.
- [18] C.-F. Juang, C.-W. Hung, and C.-H. Hsu, "Rule-based cooperative continuous ant colony optimization to improve the accuracy of fuzzy system design," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 4, pp. 723–735, Aug. 2014.
- [19] X.-M. Hu, J. Zhang, H. S.-H. Chung, Y. Li, and O. Liu, "SamACO: Variable sampling ant colony optimization algorithm for continuous optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 6, pp. 1555–1566, Dec. 2010.
- [20] K. Socha, "ACO for continuous and mixed-variable optimization," in *Proc. ANTS 4th Int. Conf. Swarm Intell.*, Brussels, Belgium, 2004, pp. 25–36.
- [21] T. Liao, T. Stützle, M. A. M. de Oca, and M. Dorigo, "A unified ant colony optimization algorithm for continuous optimization," *Eur. J. Oper. Res.*, vol. 234, no. 3, pp. 597–609, 2014.
- [22] M. Mavrovouniotis, F. M. Muller, and S. Yang, "Ant colony optimization with local search for dynamic traveling salesman problems," *IEEE Trans. Cybern.*, in press, 2016.
- [23] V. A. Shim, K. C. Tan, and C. Y. Cheong, "A hybrid estimation of distribution algorithm with decomposition for solving the multiobjective multiple traveling salesman problem," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 5, pp. 682–691, Sep. 2012.
- [24] C. W. Ahn, J. An, and J.-C. Yoo, "Estimation of particle swarm distribution algorithms: Combining the benefits of PSO and EDAs," *Inf. Sci.*, vol. 192, pp. 109–119, Jun. 2012.
- [25] J. Ceberio, E. Irurozki, A. Mendiburu, and J. A. Lozano, "A distance-based ranking model estimation of distribution algorithm for the flow-shop scheduling problem," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 286–300, Apr. 2014.
- [26] H. Karshenas, R. Santana, C. Bielza, and P. Larrañaga, "Multiobjective estimation of distribution algorithm based on joint modeling of objectives and variables," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 519–542, Aug. 2014.
- [27] K. Kim and R. I. McKay, "Stochastic diversity loss and scalability in estimation of distribution genetic programming," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 301–320, Jun. 2013.
- [28] F. Rogai, C. Manfredi, and L. Bocchi, "Metaheuristics for specialization of a segmentation algorithm for ultrasound images," *IEEE Trans. Evol. Comput.*, in press, 2016.
- [29] H. Q. Dinh *et al.*, "An effective method for evolving reaction networks in synthetic biochemical systems," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 374–386, Jun. 2015.
- [30] C.-H. Yeh and C.-Y. Yang, "Social networks and asset price dynamics," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 387–399, Jun. 2015.
- [31] Z. Vasicek and L. Sekanina, "Evolutionary approach to approximate digital circuits design," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 432–444, Jun. 2015.
- [32] C. W. Fong, H. Asmuni, and B. McCollum, "A hybrid swarm-based approach to university timetabling," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 870–884, Dec. 2015.
- [33] D. Ashlock and A. McEachern, "Evolutionary nonlinear projection," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 857–869, Dec. 2015.
- [34] S. Cussat-Blanc, K. Harrington, and J. Pollack, "Gene regulatory network evolution through augmenting topologies," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 823–837, Dec. 2015.
- [35] M.-H. Tayarani-N, X. Yao, and H. Xu, "Meta-heuristic algorithms in car engine design: A literature survey," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 609–629, Oct. 2015.
- [36] W. Sheng, S. Swift, L. Zhang, and X. Liu, "A weighted sum validity function for clustering with a hybrid niching genetic algorithm," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 6, pp. 1156–1167, Dec. 2005.
- [37] L. Qing, W. Gang, and W. Qiuping, "Restricted evolution based multimodal function optimization in holographic grating design," in *Proc. IEEE Congr. Evol. Comput.*, Edinburgh, U.K., 2005, pp. 789–794.
- [38] X. Li, "Efficient differential evolution using speciation for multimodal function optimization," in *Proc. Genet. Evol. Comput. Conf.*, Washington, DC, USA, 2005, pp. 873–880.

- [39] R. Thomsen, "Multimodal optimization using crowding-based differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, vol. 2. Portland, OR, USA, 2004, pp. 1382–1389.
- [40] J.-P. Li, M. E. Balazs, G. T. Parks, and P. J. Clarkson, "A species conserving genetic algorithm for multimodal function optimization," *Evol. Comput.*, vol. 10, no. 3, pp. 207–234, 2002.
- [41] J. Gan and K. Warwick, "Dynamic niche clustering: A fuzzy variable radius niching technique for multimodal optimisation in GAs," in *Proc. IEEE Congr. Evol. Comput.*, Seoul, South Korea, 2001, pp. 215–222.
- [42] A. Pétrowski, "A clearing procedure as a niching method for genetic algorithms," in *Proc. IEEE Congr. Evol. Comput.*, Nagoya, Japan, 1996, pp. 798–803.
- [43] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proc. Int. Conf. Genet. Algorithms*, 1987, pp. 41–49.
- [44] W. Gao, G. G. Yen, and S. Liu, "A cluster-based differential evolution with self-adaptive strategy for multimodal optimization," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1314–1327, Aug. 2014.
- [45] L. Li and K. Tang, "History-based topological speciation for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 136–150, Feb. 2015.
- [46] B. Y. Qu, P. N. Suganthan, and J. J. Liang, "Differential evolution with neighborhood mutation for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 5, pp. 601–614, Oct. 2012.
- [47] X. Li, "Niching without niching parameters: Particle swarm optimization using a ring topology," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 150–169, Feb. 2010.
- [48] C. Stoean, M. Preuss, R. Stoean, and D. Dumitrescu, "Multimodal optimization by means of a topological species conservation algorithm," *IEEE Trans. Evol. Comput.*, vol. 14, no. 6, pp. 842–864, Dec. 2010.
- [49] G. R. Harik, "Finding multimodal solutions using restricted tournament selection," in *Proc. Int. Conf. Genet. Algorithms*, 1995, pp. 24–31.
- [50] S. Hui and P. N. Suganthan, "Ensemble and arithmetic recombination-based speciation differential evolution for multimodal optimization," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 64–74, Jan. 2016.
- [51] S. Biswas, S. Kundu, and S. Das, "An improved parent-centric mutation with normalized neighborhoods for inducing niching behavior in differential evolution," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1726–1737, Oct. 2014.
- [52] S. Biswas, S. Kundu, and S. Das, "Inducing niching behavior in differential evolution through local information sharing," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 246–263, Apr. 2015.
- [53] B. Y. Qu, P. N. Suganthan, and S. Das, "A distance-based locally informed particle swarm model for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 387–402, Jun. 2013.
- [54] Y. Wang, H.-X. Li, G. G. Yen, and W. Song, "MOMMOP: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 830–843, Apr. 2015.
- [55] A. Basak, S. Das, and K. C. Tan, "Multimodal optimization using a biobjective differential evolution algorithm enhanced with mean distance-based selection," *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 666–685, Oct. 2013.
- [56] K. Deb and A. Saha, "Multimodal optimization using a bi-objective evolutionary algorithm," *Evol. Comput.*, vol. 20, no. 1, pp. 27–62, 2012.
- [57] J. Yao, N. Kharma, and P. Grogono, "Bi-objective multipopulation genetic algorithm for multimodal function optimization," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 80–102, Feb. 2010.
- [58] X. Li, "Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization," in *Proc. Conf. Genet. Evol. Comput.*, Seattle, WA, USA, 2004, pp. 105–116.
- [59] M. Dorigo, "Optimization, learning and natural algorithms," Ph.D. dissertation, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy, 1992.
- [60] M. Dorigo, V. Maniezzo, and A. Colomi, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.
- [61] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.
- [62] M. Dorigo, M. Birattari, and T. Stützle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006.
- [63] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," *Eur. J. Oper. Res.*, vol. 185, no. 3, pp. 1155–1173, 2008.
- [64] R. K. Ursem, "Multinational evolutionary algorithms," in *Proc. IEEE Congr. Evol. Comput.*, Washington, DC, USA, 1999, pp. 1633–1640.
- [65] R. K. Ursem, "Multinational GAs: Multimodal optimization techniques in dynamic environments," in *Proc. Conf. Genet. Evol. Comput.*, 2000, pp. 19–26.
- [66] J. Yao, N. Kharma, and Y. Q. Zhu, "On clustering in evolutionary computation," in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, 2006, pp. 1752–1759.
- [67] C. L. Stoean, M. Preuss, R. Stoean, and D. Dumitrescu, "Disburdening the species conservation evolutionary algorithm of arguing with radii," in *Proc. Conf. Genet. Evol. Comput.*, London, U.K., 2007, pp. 1420–1427.
- [68] Q. Yang *et al.*, "Multimodal estimation of distribution algorithms," *IEEE Trans. Cybern.*, in press, 2016.
- [69] S. Goss, S. Aron, J.-L. Deneubourg, and J. M. Pasteels, "Self-organized shortcuts in the Argentine ant," *Naturwissenschaften*, vol. 76, no. 12, pp. 579–581, 1989.
- [70] Y. Lin *et al.*, "An ant colony optimization approach for maximizing the lifetime of heterogeneous wireless sensor networks," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 3, pp. 408–420, May 2012.
- [71] Y. Zhou, X. Lai, Y. Li, and W. Dong, "Ant colony optimization with combining Gaussian eliminations for matrix multiplication," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 347–357, Feb. 2013.
- [72] J.-W. Lee, B.-S. Choi, and J.-J. Lee, "Energy-efficient coverage of wireless sensor networks using ant colony optimization with three types of pheromones," *IEEE Trans. Ind. Informat.*, vol. 7, no. 3, pp. 419–427, Aug. 2011.
- [73] W.-N. Chen and J. Zhang, "Ant colony optimization for software project scheduling and staffing with an event-based scheduler," *IEEE Trans. Softw. Eng.*, vol. 39, no. 1, pp. 1–17, Jan. 2013.
- [74] W.-N. Chen and J. Zhang, "An ant colony optimization approach to a grid workflow scheduling problem with various QoS requirements," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 39, no. 1, pp. 29–43, Jan. 2009.
- [75] W.-N. Chen, J. Zhang, H. S.-H. Chung, R.-Z. Huang, and O. Liu, "Optimizing discounted cash flows in project scheduling—An ant colony optimization approach," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 40, no. 1, pp. 64–77, Jan. 2010.
- [76] M. Shen, W.-N. Chen, J. Zhang, H. S.-H. Chung, and O. Kaynak, "Optimal selection of parameters for nonuniform embedding of chaotic time series using ant colony optimization," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 790–802, Apr. 2013.
- [77] T. Liao, M. A. M. de Oca, D. Aydin, T. Stützle, and M. Dorigo, "An incremental ant colony algorithm with local search for continuous optimization," in *Proc. Genet. Evol. Comput. Conf.*, Dublin, Ireland, 2011, pp. 125–132.
- [78] D. Martens *et al.*, "Classification with ant colony optimization," *IEEE Trans. Evol. Comput.*, vol. 11, no. 5, pp. 651–665, Oct. 2007.
- [79] C. Blum and M. Dorigo, "Search bias in ant colony optimization: On the role of competition-balanced systems," *IEEE Trans. Evol. Comput.*, vol. 9, no. 2, pp. 159–174, Apr. 2005.
- [80] M. Birattari, P. Pellegrini, and M. Dorigo, "On the invariance of ant colony optimization," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 732–742, Dec. 2007.
- [81] S. Das, A. Konar, and U. K. Chakraborty, "Two improved differential evolution schemes for faster global search," in *Proc. Genet. Evol. Comput. Conf.*, Washington, DC, USA, 2005, pp. 991–998.
- [82] X. Li, A. Engelbrecht, and M. G. Epitropakis, "Benchmark functions for CEC'2013 special session and competition on niching methods for multimodal function optimization," *Evol. Comput. Mach. Learn. Group, RMIT Univ., Melbourne VIC, Australia, Tech. Rep.*, 2013.
- [83] J. Kennedy, "Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance," in *Proc. IEEE Congr. Evol. Comput.*, Washington, DC, USA, 1999, pp. 1931–1938.
- [84] P. N. Suganthan, "Particle swarm optimizer with neighbourhood operator," in *Proc. IEEE Congr. Evol. Comput.*, Washington, DC, USA, 1999, pp. 1958–1962.
- [85] C. J. Wild, *Chance Encounters: A First Course in Data Analysis and Inference*. New York, NY, USA: Wiley, 2000.
- [86] M. Preuss, "Niching the CMA-ES via nearest-better clustering," in *Proc. Genet. Evol. Comput. Conf.*, Portland, OR, USA, 2010, pp. 1711–1718.
- [87] J. E. Fieldsend, "Running up those hills: Multi-modal search with the niching migratory multi-swarm optimizer," in *Proc. IEEE Congr. Evol. Comput.*, Beijing, China, 2014, pp. 2593–2600.



**Qiang Yang** (S'14) received the M.S. degree from Sun Yat-sen University, Guangzhou, China, in 2014, where he is currently pursuing the Ph.D. degree.

He is also currently a Research Assistant with the School of Computer Science and Engineering, South China University of Technology, Guangzhou. His current research interests include evolutionary computation algorithms and their applications on real-world problems, large scale optimization algorithms, multimodal optimization algorithms, and distribute evolutionary algorithms and their applications on real-world problems, like intelligent transportation.



**Wei-Neng Chen** (S'07–M'12) received the bachelor's and Ph.D. degrees from Sun Yat-sen University, Guangzhou, China, in 2006 and 2012, respectively.

He is currently a Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou. He has published 50 papers in international journals and conferences. His current research interests include swarm intelligence algorithms and their applications on cloud computing, operations research, and software engineering.

Dr. Chen was a recipient of the IEEE Computational Intelligence Society Outstanding Dissertation Award in 2016, for the doctoral thesis, the Natural Science Foundation for Distinguished Young Scientists of Guangdong Province, China, in 2015, the Guangdong Special Support Program for Outstanding Young Scientists in 2015, and the Pearl River New Star in Science and Technology in 2014.



**Zhengtao Yu** received the Ph.D. degree in computer application technology from the Beijing Institute of Technology, Beijing, China, in 2005.

He is currently a Professor with the School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China. His current research interests include natural language processing, information retrieval, and machine learning.



**Tianlong Gu** received the M.Eng. degree from Xidian University, Xi'an, China, in 1987, and the Ph.D. degree from Zhejiang University, Hangzhou, China, in 1996.

From 1998 to 2002, he was a Research Fellow with the School of Electrical and Computer Engineering, Curtin University of Technology, Perth, WA, Australia, and a Post-Doctoral Fellow with the School of Engineering, Murdoch University, Perth. He is currently a Professor with the School of Computer Science and Engineering, Guilin

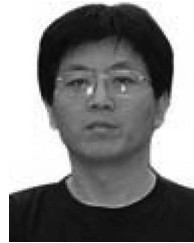
University of Electronic Technology, Guilin, China. His current research interests include formal methods, data and knowledge engineering, software engineering, and information security protocol.



**Yun Li** (S'87–M'90) received the B.S. degree in radio electronics science from Sichuan University, Chengdu, China, in 1984, the M.Eng. degree in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, in 1987, and the Ph.D. degree in parallel processing for control engineering from the University of Strathclyde, Glasgow, U.K., in 1990.

From 1989 to 1990, he was with U.K. National Engineering Laboratory and Industrial Systems and Control Ltd, Glasgow, U.K. He joined the University of Glasgow, Glasgow, U.K., as a Lecturer, in 1991, served as the Founding Director of the University of Glasgow Singapore, Singapore, from 2011 to 2013, and was an Interim/Founding Director of the university's first joint programme in China, in 2013, with the University of Electronic Science and Technology (UESTC), Chengdu, China. He established Evolutionary Computation workgroups for the IEEE Control System Society and European Network of Excellence in Evolutionary Computing (EvoNet) in 1998 and served on the Management Board of EvoNet from 2000 to 2005. He has been a Visiting Professor with Kumamoto University, Kumamoto, Japan, UESTC, and Sun Yat-sen University, Guangzhou, China. He has supervised over 20 Ph.D. students, and has over 200 publications.

Prof. Li is a Chartered Engineer in the U.K.



**Huaxiang Zhang** received the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2004.

He is currently a Professor with the School of Information Science and Engineering, Shandong Normal University, Jinan, China, where he was an Associate Professor with the Department of Computer Science, from 2004 to 2005. He has authored over 100 journal and conference papers and has been granted eight invention patents. His current research interests include machine learning, pattern recognition, evolutionary computation, and Web information processing.



**Jun Zhang** (M'02–SM'08) received the Ph.D. degree in electrical engineering from the City University of Hong Kong, Hong Kong, in 2002.

He is currently a Professor with the South China University of Technology, Guangzhou, China. His current research interests include computational intelligence, cloud computing, wireless sensor networks, operations research, and power electronic circuits. He has authored seven research books and book chapters, and over 50 IEEE TRANSACTIONS papers in the above areas.

Prof. Zhang was a recipient of the National Science Fund for Distinguished Young Scholars in 2011 and the First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009. He was also appointed as the Changjiang Chair Professor in 2013. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and the IEEE TRANSACTIONS ON CYBERNETICS. He is the Founding and Current Chair of the IEEE Guangzhou Subsection, the IEEE Beijing (Guangzhou) Section Computational Intelligence Society Chapters, and ACM Guangzhou Chapter.