

A Survey on Particle Swarm Optimization Algorithms for Multimodal Function Optimization

Yu Liu¹, Xiaoxi Ling^{1,2}, Zhewen Shi¹, Mingwei Lv¹, Jing Fang¹ and Liang Zhang¹

¹School of Software, Dalian University of Technology, Dalian, China

²Civil Aviation Flight University of China, Guanghan, P.R.China

Email: yuliu@dlut.edu.cn

Abstract— Many scientific and engineering applications involve finding more than one optimum. A comprehensive review of the existing works done in the field of multimodal function optimization was given and a critical analysis of the existing methods was also provided. Several techniques in solving multimodal function optimization problems were introduced, such as clearing, deterministic crowding, sharing, species conserving and so on. And we summarized defects of existing algorithms: lacking of self-adaptive adjustment function, requiring setting some parameters according to different problems, lacking of unified theoretical and experimental system to guide algorithms design and not maintaining the diversity of swarm. Moreover, most of existing multimodal particle swarm optimization algorithms which include SPSO, MSPSO, ESPSO, ANPSO, kPSO, MGPSO, AT-MGPSO, rpso, and SDD-PSO were described and compared and advantages and disadvantages existing in these algorithms were pointed out. Therefore, some ideas to improve the performance of multimodal function optimization algorithms were proposed.

Index Terms—Multimodal Function Optimization, Evolutionary Algorithm, Particle Swarm Optimizer

I. INTRODUCTION

Multimodal Function Optimization (MFO) is the problems which have more than one optimum, or exist only one global optimum and several useful local optima in feasible solution spaces. There are a large number of such problems existing in the real world, such as Rule Discovery in Data Mining, Neutral Network Integration, Fuzzy System, Optimization, Multi-solution Investigation and so on [32]. In order to offer a variety of options and more information to decision makers, multimodal function optimization algorithms try to find all these global optima and useful local optima. However, conventional optimization algorithms can only find one optimum randomly, and show ineffectiveness in solving such problems. Evolutionary Algorithm (EA) using population search instead of individual search in the conventional algorithms has obvious advantages and great probability to locate multiple optima simultaneously in feasible search spaces. Evolutionary Algorithm (EA) is the method that simulates natural biological activities and evolutionary process, establishes computation model and solves complex optimization problems. The search

process only relies on fitness values, which is independent of gradient information. And that is widely used and not constrained by the differentiable and continuous limitations of problems, and has strong universality. Therefore, the application of Evolutionary Algorithm (EA) for solving multimodal function optimization problems, which has significantly theoretical value and practical significance, has aroused a widespread concern in the field of evolutionary computation and has become an important research field.

We can see the definition of Multimodal Function Problems from equation (1).

$$\begin{aligned} \min z = f(X) \text{ or } \max z = f(X), X = (x_1, \dots, x_n) \in R^n \quad (1) \\ \exists (X_1, X_2, \dots, X_k) \text{ When } X^* = X_1, X_2, \dots, X_k, \\ \forall(X) f(X) \leq f(X^*) \text{ or } \forall(X) f(X) \geq f(X^*), \text{ s.t. } \|X^*, X\| \leq r_k. \end{aligned}$$

Where r_k is a parameter that represents the radius of the k th optimum solution's territory. $\|X^*, X\|$ represents the Euclidean Distance between X^* and X .

In this paper, the formulation of multimodal function optimization was given. And we gave a comprehensive review of the existing works done in multimodal function optimization and analyzed several popular techniques, such as clearing, deterministic crowding, sharing, species conserving and so on. We also summarized major defects of most existing multimodal function optimization algorithms suffered. Moreover, we compared most existing multimodal particle swarm optimization algorithms which include SPSO [24], MSPSO [25], ESPSO [26], ANPSO [30], kPSO [27], MGPSO [28], AT-MGPSO [29], rpso [32], and SDD-PSO [31].

II. NICHING TECHNIQUES

Genetic Algorithm (GA), which has developed completely during a long time, is the most representative algorithm in the evolutionary computing. New techniques and theories are all based on Genetic Algorithm (GA). In the standard Genetic Algorithm, due to the random of selection and reorganization, genetic drift exists in the population [1], which makes the population converge to an absorbing state and finds only one optimum. In order to solve multimodal function optimization problems,

researchers restrict the search behaviors of the standard Evolutionary Algorithm (EA), and introduce niching techniques, which divides the swarm into several small swarms by some means. Algorithms select and evolve in the entire population, and at the same time save the best individual in each niche. The algorithms not only maintain the diversity of individuals, but also retain a variety of high-fitness individuals, which prevents a high-fitness individual from filling with the entire population quickly and can find more than one global optimum or local optimum. The representative niching techniques are crowding methods, fitness sharing, and sequential niche technique and species conservation.

Crowding technique maintains the diversity of population by appropriate replacement policy. It is based on the theory that a variety of creatures have to compete for a variety of limited resources to survive in a limited living space. Cavichio [2] first proposed niche implement based on pre-selection in 1970: if offspring's fitness value is higher than the poor individuals of parents, it will replace the parent. In 1975, Jong [3] extended Cavichio's pre-selection mechanism and proposed standard crowding mechanism: select $1/CF$ (Crowding Factor) individuals from the population to form a temporary sub-population, and then compare the similarity which is determined by the number of matched alleles between new individuals and the individuals in the temporary sub-population. Then a new individual replaces the individual that has the greatest similarity with it. Although standard crowding technique improves the diversity of population, it cannot solve the situation of more than two peaks due to replacement deviation and shows a limited ability to maintain niche in multi-peak problems [4]. Malifoud [5] extended standard crowding technique by including competition between parents and offspring in the same niche and proposed Deterministic Crowding (DC): divide population whose size is N into $N/2$ pairs and each pair of parent individuals generates two offspring individuals through crossover and mutation, and offspring and parents compete to become members of next population by a competition league. Deterministic Crowding technique has solved some multi-peak optimization problems and shows strong ability to maintain niches. However, replacement policy of Deterministic Crowding (DC) can lead to loss of low-fitness niches. In order to overcome the defect, Mengshoel [6] proposed Probability Crowding (PC): offspring compete with similar parents and replace parents according to fitness values.

Fitness sharing is based on the theory that individuals are punished as the presence of other individuals in the survival space. That means individuals compete for limited resources to survive in the same niche. An individual's allocated resources decreases as the presence of other individuals and the individual selects the niche which has the maximum average of resources rather than which has the maximum of resources. Goldberg and Reihardson [7] firstly proposed a niching technique based on fitness sharing in 1987. The mechanism defined a sharing function and the niche radius controls the shape of sharing function as a constant. Sharing Function is the

function on the similarity between two individuals (similarity of genotype or similarity of phenotype). When the similarity between two individuals is larger, the value of sharing function is larger; on the contrary, the value is smaller. This function determines each individual's sharing value. An individual's sharing value is equal to the sum of sharing function values between the individual and each of other individuals in the same population. Then each individual's fitness is adjusted according to its sharing value. After that, the algorithm can carry out selection operation based on the new sharing value to maintain the diversity of the population and create the niche's evolutionary environment in the process of population evolution. This model solved multimodal function optimization problems successfully and became a popular niching technique. Researchers proposed a number of improved algorithms in order to improve the effectiveness of fitness sharing genetic algorithm. Petrowski [8] proposed clearing improved fitness sharing technique. Clearing divides the population into q niches according to clustering analysis. However, due to the concept of limited resources, each niche only retains k better individual's fitness value and other individuals' fitness value is set to 0. Each niche only needs to compute sharing value for k better individuals, which reduces the cost of sharing technique's implementation. Dynamic Niche Clustering [9] (DNC) regards each individual in the initial population as a niche, and merges, decomposes and creates niches dynamically by fuzzy clustering analysis, and calculates sharing value in each niche.

Sequential Niche Technique [10] runs Genetic Algorithm (GA) repeatedly for the same problem and maintains the optima which have been found at the end of each run. In order to reduce the probability of repeatedly visiting the same optimum which has been found, this algorithm uses a similar method of calculating fitness sharing to reduce all the solutions' fitness values within the radius of optima that have been found during the next run.

Species Conserving [11] sorts the individuals according to fitness values at each generation. First, the algorithm selects the best individual as the first species seed of seed collection, and then each individual is compared with individuals in the seed collection sequentially. If the distance between an individual and all the species seeds is longer than specified radius, this individual will be added into seed collection as a new seed. The algorithm will not end until the worst individual has compared with the individuals in seed collection and formed final seed collection, which conserved to the next generation.

In order to solve multimodal function optimization problems, on the one hand, algorithms have to maintain the diversity of the population in order to converge on multiple solutions. On the other hand, algorithms must have better global search capability and can get rid of insignificant local optima and find better optima rapidly. They are not only essential, but also restricted to each other. Genetic Algorithm's defect of weak local search

capability makes it have better global search capability and can find vicinity of optima rapidly. However, if Genetic Algorithm (GA) wants to improve the solutions' qualities, it needs expensive computation cost. There are two kinds of thoughts to improve the search capability of Genetic Algorithm (GA). One is combining local search methods with genetic algorithm (GA) to overcome genetic algorithm's defects. Another one is using new principles of evolutionary search. Researchers try to use Differential Evolution [12], Artificial Immunity [13-14], Evolution Strategy [15] and Particle Swarm Optimizer (PSO) [17-25] to solve multimodal function optimization problems.

III. MULTIMODAL PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimizer (PSO) [16] is an evolutionary algorithm proposed by Kennedy and Eberhart in 1995. Particle Swarm Optimizer (PSO) imitates population's social behavior instead of relying on individual law of natural evolution. The existence of individual and individual, individual and population interaction and mutual influence behavior shows that sharing information exists in the population. Individuals rely on simple rules of particles to produce complex social behaviors and get better performance through sharing information and constant interaction. Particle Swarm Optimizer (PSO) develops very rapidly and becomes one of hot research field in evolutionary computation. Especially in the continuous variable function problems, Particle Swarm Optimizer (PSO) shows great potential and outstanding performance and stands out from a number of evolutionary algorithms.

Sharing information based on swarm intelligence provides a new thought for algorithm design. Particle Swarm Optimizer (PSO) is still in its infancy and most studies focus on the improvement of algorithm's performance. The use of Particle Swarm Optimizer (PSO) to solve multimodal function optimization problems has just started. Kennedy [17] introduced cluster technique to solve multimodal function optimization problems. Li et al [18] introduced sharing function used in Genetic Algorithm (GA) into Particle Swarm Optimizer (PSO). Parsopoulos et al [19] made use of compression and stretching techniques to solve multimodal function optimization problems. Brits [20] proposed NichePSO by using sub-swarm. Ozcan [21] improved the performance of NichePSO by using fanaticism mechanism and climbing mechanism, however, this algorithm need mutually set parameters associated with niche. Seo [22] developed a new strategy of selecting attractor *gbest* to solve multimodal function optimization problems; however, the algorithm needs to set the number of peaks artificially. Li [23] proposed multimodal function optimization based on Fitness Euclidean-Distance Ratio (FED). Particles select the individual's *gbest* whose fitness value has changed the most greatly in unit distance according to Fitness Euclidean-Distance Ratio (FED) as the attractor when particles update velocities and the algorithm has the similar problems with sharing function technique, which is that the algorithm needs to compute

N^2 (N is the population size) Fitness Euclidean-Distance Ratio. When optimizing complex problems, the algorithm needs a larger population size and the computation cost will increase sharply.

Most of existing multimodal particle swarm optimization algorithms are based on species conservation and fitness sharing. SPSO [24], MSPSO [25], ESPSO [26], ANPSO [30], kPSO [27], MGPSO [28], AT-MGPSO [29], and rpsO [32] are based on species conservation. SDD-PSO [31] is based on fitness sharing.

Algorithms based on species conservation can be divided into two classes, depending on niching parameters and parameters independent. SPSO and MSPSO depend on niching parameters. Users need to prespecify the niching parameter which determines how large a niche is. Depending on niching parameters is the greatest disadvantage of these algorithms. Users have to set different radius according to different problems. And this restricts these algorithms widely used in the real world. Although ANPSO, MGPSO and AT-MGPSO also need niching parameters, these algorithms adaptively determine niching parameters at each generation. ANPSO uses the intrinsic nature of the particles to converge on optima and create niches when they do so. And MGPSO and AT-MGPSO determine the size of niches according to the feasible solution space. ESPSO doesn't depend on niching parameters, but it introduces three new parameters s , m and δ . The first two have been shown to be robust across the test functions. The last is problem dependent; however, it is intuitive and easy to set. kPSO is also independent of niching parameters. However, depending on parameter c and the number of steps between clustering limits the algorithm widely used in the real world. rpsO doesn't depend on any niching parameter. And it shows that PSO with ring topology is able to induce stable niching behaviors.

Li proposed a Species-based Particle Swarm Optimization (SPSO) [24] in 2004. The algorithm needs to set a niching parameter r . At each generation, SPSO sorts all the particles descendingly by the current fitness values. Then the algorithm determines whether two particles are in the same niche by calculating the Euclidean Distance between them. If the distance is smaller than r , the two particles are in the same niche. If the distance is larger than r , they are in different niches. After the niches divided, the algorithm updates each particle's *gbest* in each niche by the best particle's *pbest* in this niche. Experiments show that SPSO has good performance in solving several standard multimodal test functions. However, depending on the parameter r which determines the size of a niche is the greatest weakness of SPSO. If r is too small, particles probably trap into local optima, or the algorithm may miss some global optima.

IWAMATSU and Masao [25] proposed a modified Particle Swarm Optimizer (PSO) called Multi-species Particle Swarm Optimization (MSPSO) to locate all the global optima of multimodal function optimization problems. MSPSO has the same idea as SPSO; however,

MSPSO sorts all the particles descendingly by their personal best fitness values at each generation.

Bird and Li proposed an adaptively niching parameters Particle Swarm Optimization algorithm (ANPSO) [30] in 2006. Due to most of existing algorithms need to set niching parameters to determine a niche's size. This greatly reduces the algorithms' usefulness and effectiveness. ANPSO adaptively determines niching parameters at each step. The method uses the intrinsic nature of the particles to converge on optima and create niches when they do so. When a particle discovers a local peak, its velocity will reduce and it will explore the area closely. If multiple particles explore the same area, it is likely that it is a peak of interest, so a niche is created with those particles as its initial members. Any other particles that later converge on the peak will also join the niche. In step 2, an undirected graph g is created containing a node for each particle. Initially there are no edges between any of the nodes. Step 3 finds every set of particles that have been close to each other for at least 2 steps. The algorithm finds the pairs of particles that are within r of each other. A counter is maintained for each pair – if the pair has been close for two or more steps, a niche is formed. After above three steps, all the particles are either in a particular niche, or are not in any niche. For each niche, using the best particle's $pbest$ updates other particles' $gbest$. The unniched particles are placed in a von Neumann Neighborhood which is updated at each step. Although ANPSO has good performance in handling some standard multimodal test functions, it has poor performance in solving high dimensional multimodal problems.

Bird and Li proposed an enhanced version of SPSO, ESPSO [26] in 2006. ESPSO enhances SPSO by greatly increasing the robustness of the niching parameter – to point that the algorithm is still effective even if it isn't used at all. ESPSO includes Detecting Convergence, Preventing Future Interactions and Removing Duplicate Species into PSO. It introduces three new parameters s , m and δ . The first two have been shown to be robust across the test functions. The last is problem dependent; however it is intuitive and easy to set. However, depending on the three parameters affects the algorithm's effectiveness and efficiency.

Passaro and Starita [27] proposed a new algorithm called k-means-based PSO (kPSO) to niching in PSO based on clustering particles to identify niches in 2008. kPSO employs the standard k-means clustering algorithm and Bayesian Information Criterion to adaptively identify the number of clusters. Through solving a set of multimodal test functions, kPSO shows better performance than some other existing algorithms. Although kPSO is a competitive solution in solving multimodal function optimization problems, the algorithm needs to set parameter c and the number of steps between clustering. Depending on parameter c and the number of steps between clustering limits kPSO widely used in the real world.

Seo et al [28] proposed a new algorithm solving multimodal function optimization problems called Multi-

grouped Particle Swarm Optimization (MGPSO). The algorithm gives every group a territory to avoid overlapping of discovered solutions. To encourage the individual particle they proposed the concept of repulsive velocity, located in territory of other group, to escape from the other group's territory in more efficient manner. In this algorithm, if a particle intrudes other group's territory, the group will push the particle out from its own territory by updating the particle's velocity according to equation (2).

$$V_{ij}^{k+1} = w_{ij}^k + C_1 R_1 (pbest_{ij}^k - X_{ij}^k) + C_2 R_2 (gbest_i^k - X_{ij}^k) + C_3 R_3 (X_{ij}^k - gbest_m^k) \quad (2)$$

Where X_{ij}^k and V_{ij}^k are the position and velocity of the j th particle in the i th group at k th iteration. w is inertia weight and is usually decreasing linearly from 0.9 to 0.4 throughout the simulation. $pbest_{ij}^k$ is the i th group's j th particle's personal best ($pbest$) at the k th iteration. $gbest_i^k$ is the i th group's global best ($gbest$) at the k th iteration. $gbest_m^k$ is the global best ($gbest$) of intruded group. C_1 and C_2 are acceleration factors which determine the relative pull for every particle toward personal best ($pbest$) and global best ($gbest$). C_3 is repulsively coefficient. However, it has a zero value if the j th particle in the i th group doesn't intrude other group's territory. R_1 , R_2 and R_3 are random numbers between 0 and 1. The forth term of the equation is a repulsive velocity component. And it is used to push the particle out from the territory of other group's intruded by the particle. The algorithm protects all the groups' territories from intruding by other group's particles. In MGPSO, all the groups have the same size of territories that means they have the same radius, and the radius decreases linearly in a decreasing order throughout the iteration. However, if the radius became too small before sufficient convergence level, some groups cannot find their own solutions and wandered around other groups' solutions. To overcome this defect, they proposed another algorithm based on MGPSO called Auto-tuning Multi-grouped Particle Swarm Optimization (AT-MGPSO) [29] in 2008. In AT-MGPSO, competition mechanism is invited and all the groups have different radius. When two groups' territories overlapped, the winner group which has the higher global fitness remains and the radius is increasing by dividing it by 0.95. And the loser group is expelled and reinitialized out of the existing groups' territories and searches other solutions. The algorithm is based on the idea that is such that a group with a broad scope of influence and high fitness has higher probability to be invaded by other groups. Although AT-MGPSO has good performance in handling simple multimodal test functions, it needs to set the niching parameter which determines how large a niche is. In this algorithm, r is set to 5% of the whole solution space. However, for simple problems, AT-MGPSO can get good performance, for complex problems, it cannot get so good performance. For example, for two-dimensional Shubert function, it has 18

global optima divided into 9 clusters; the distance between two optima in the same cluster is very small. And the niching parameter in AT-MGPSO is much larger than that distance, so we may miss some global optima.

Li proposed a new niching Particle Swarm Optimization based on ring topology in 2010 [32]. The algorithm doesn't depend on any niching parameter. He has demonstrated that the PSO algorithms with ring topology are able to induce more stable niching behavior. The PSO algorithms with overlapping are able to locate multiple global optima, given a reasonably large population size, whereas the PSO algorithms with a non-overlapping ring topology can be used to locate global as well as local optima, especially for low dimensional problems. Experiments show that the PSO algorithms with a ring topology can provide comparable or better, and more consistent performance, than some existing niching PSO algorithms. Even with a comparable or smaller population size, the proposed algorithms can outperform a niching algorithm using a fixed niche radius, in terms of success rate and the actual number of global optima found. Most importantly, one major advantage over existing niching algorithms is that no niching parameters are required. This should pave the way for more widespread use of this kind of niching algorithms in real-world applications. The algorithm is the first attempt showing that PSO algorithms with ring topology are able to induce stable niching behavior. The algorithm suggests that local memory and slow communication topology are two key elements for the success.

Most of algorithms based on fitness sharing introduce new mechanism to maintain diversity of swarm in order to locate all the global optima of multimodal functions. In SDD-PSO [31], a mutation operator is introduced to prevent premature convergence in high dimensional functions. This algorithm results in a superior performance and robustness to all other parameter configurations tested for some standard test functions.

Worasuchep [31] proposed a modified PSO with two techniques: a mutation operator to increase swarm diversity for high-dimensionality and an improved mechanism to detect and resolve the stagnation once it is found. In that paper, stagnation detection and dispersion (SDD) mechanism was modified and enhanced. A mutation operator is introduced to prevent premature convergence in high dimensional functions. Before the velocity and position update in each generation, the mutation method works as follow. For each particle, if a randomized value is below a mutation probability (mp), the particle's position is re-randomized within the range of each dimension. This applies to every dimension of the mutated particle. A few internal parameters for swarm dispersion are adjusted in order to increase the strengths of dispersion effects. This enhancement is more essential for asymmetric initialization (than in symmetric initialization). It should be beneficial for high-dimensionally as well. This configuration is based on a sensitivity analysis of the results from the preliminary experimentation using various parameter configurations on a set of test functions. This algorithm results in a

superior performance and robustness to all other parameter configurations tested for some standard test functions. However, this algorithm doesn't have good local search capability.

Due to evolutionary algorithms' bionic objects and mechanisms are different, algorithms' search mechanism, optimization efficiency and scope are different. In genetic algorithm (GA), the generation of new individuals is mainly driven by crossover; however, mutation is only as a supplement and cannot afford a decisive role. Therefore, genetic algorithm (GA) has better global search capacity and can locate the vicinity of optimal solution quickly; however, crossover hardly makes individuals have better local search capacity, and only rely on reducing the population converge to refine the solution. So genetic algorithm (GA) has weak capacity of local search and has low search efficiency. In evolution strategy and artificial immunity algorithms, the generation of new individuals is mainly driven by mutation. Individuals have strong local search capacity; however, due to lacking of interactions between individuals and worse global search capacity, individuals easily fall into valueless local optima. Particle Swarm Optimization's (PSO) powerful search capacities are from the advanced search driving principle (pattern of generating new individuals). Particles complete self-learning function and social-learning function by use of sharing information and have a good balance between global search capacity and local search capacity and have high search efficiency. Particle Swarm Optimizer (PSO) has better search capacity than other algorithms and has absolute advantages in the continuous variable function optimization problems. Furthermore, Particle Swarm Optimizer (PSO) has a distinct biological background of swarm intelligence. Sharing information provides a new thought for algorithms design. And algorithm's performance still has a large space to further improve. Species conservation applied on Particle Swarm Optimizer (PSO) has better performance than that applied on Genetic Algorithm (GA) based on previous research experiments, which shows Particle Swarm Optimizer's (PSO) advantages and great probability. However, Particle Swarm Optimizer (PSO) doesn't have good performance and high success rate in solving high dimensional and complex multimodal function optimization problems and most optimization algorithms based on Particle Swarm Optimizer (PSO) depends on prior knowledge of specified issues. And the algorithms can fail because of inappropriate parameters.

IV. APPLICATIONS TO MULTIMODAL FUNCTION OPTIMIZATION

In the real world, there are many multimodal function optimization problems about mathematical and project problems. Many advances in science, economics and engineering rely on numerical techniques for computing globally optimal solutions to corresponding optimization problems. These problems are extremely diverse and have many optimal solutions. For example, optimization problem about neural network's structure and weight, complex system's parameter, structure recognition,

economic modeling, neural networks training, image processing and engineering design and control and so on.

Scientific and engineering applications have many optimal solutions and regularly require algorithms to locate all the optimal solutions. This problem is NP-hard in the sense of its computational complexity even in simple cases. For example, nonlinear least squares problems, as well as feed forward neural network training and solving systems of equations [32], a comparatively simple task when only a few equations and unknowns are involved, grow notably more complex as the problem's dimensionality increases. The stochastic search algorithms are widely used in evolving artificial neural network (ANN) architecture and weights. As a rule, the best weights or architecture of an ANN are not exclusive. In fact, the different architecture of an ANN is very useful in different situations.

Multimodal function optimization is also widely used on magnetic. Interior Permanent-Magnet Synchronous Motor (IPMSM) [28] is a practical multimodal function optimization example which has four poles and 18 slots was selected. The objective of the optimal design was the maximum motor efficiency. However, if the value of the motor efficiency is similar, the solution with a lower winding temperature is preferred.

Multimodal function optimization is also widely used in the field of detecting intrusion [33]. The detecting intrusion technique based-rules are difficult to manage rule base and to establish statistical model. This technique also has higher false alarm rate and omission rate. However, multimodal function optimization algorithms can solve these problems.

In the field of cryptography [34], a number of hard and complex computational problems have been motivated. These problems have more than one optimum solution and the computational cost in solving them is very expensive. Such problems are the integer factorization problem related to the RSA cryptosystem; the index computation or the discrete algorithm problem related to the El Gamal cryptosystem and so on. We can use multimodal function optimization algorithms to solve these problems in an efficient way.

V. CONCLUSIONS

Evolutionary Algorithm has unique advantage in solving multimodal function optimization problems. Though multimodal function optimization algorithms based on Evolutionary Algorithm have been developed for a long time and varieties of techniques appeared and were used successfully in application, the design and theoretical research of algorithms are far from perfect and there are many issues worthy for further study. For instance:

(1) Due to the algorithms are lack of self-adaptive adjustment function and some parameters need be set mutually according to problems, which limit the algorithms' flexibility and scope and make the algorithm difficult to apply to practical applications widely. For example: a niche radius relies on prior knowledge of specified issues and inappropriate value will make the

algorithm fail; Fixed-size population can only maintain a certain number of equilibrium sub-population. For the specific applications, the algorithm is lack of estimation of population size parameters or the adaptively dynamic adjustment.

(2) Lack of unified theoretical and experimental system to guide algorithms design. It is difficult to analyze from theory and compare different algorithms' capacity of formation and maintenance niche.

(3) Niching techniques achieve the purpose searching multiple areas by increasing the diversity of population which is at the expense of decreasing local searching capability and convergence speed. If not improve the algorithm's search capability, though the algorithm can find multiple solutions, it need run a long time and even cannot find desirable solutions. And the algorithm will lose practical significance.

In the future, in order to solve the above problems, multimodal function optimization algorithms can introduce reinforcement learning mechanism to enhance the algorithms' search capacity. And we can use sharing information technique to make algorithms have a good balance between global search capacity and local search capacity. Other mechanisms can be introduced into multimodal function optimization algorithms to make the algorithms not rely on the prior knowledge of specified issues and self-learn the niching parameters. That can make the optimization algorithms have better practicality. We can study how to increase the search capability of small niches so that the performance of these niches will locate well with increasing dimensions. We can also develop techniques to adapt or self-adapt the population size, as the population size is a parameter that also needs to be supplied by users and focus on applying multimodal particle swarm optimization algorithms to tracking multiple peaks in a dynamic environment.

ACKNOWLEDGMENT

This work was funded by the Natural Science Foundation of China under grant No. 60803074 and the Fundamental Research Funds for the Central Universities (No. DUT10JR06).

REFERENCES

- [1] Mahfoud SW, "Genetic drift in sharing methods," IEEE World Congress on Computational Intelligence. 1994, pp. 67-72.
- [2] Cavicchio D.J, "Reproductive adaptive plans," Proceedings of the ACM annual conference, 1972, pp. 60-70.
- [3] De Jong and K.A, "Analysis of the behavior of a class of genetic adaptive systems," [Ph. D Thesis], Michigan: University of Michigan, 1975.
- [4] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," *IEEE Transl. J. Magn. Japan*, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetism Japan, p. 301, 1982].
- [5] SW Mahfoud, "Crowding and preselection revisited," Amsterdam Elsevier in Parallel Problem Solving from Nature, 1992, pp.27-36.

- [6] Mengshoel OJ and Goldberg D E, "Probabilistic crowding: Deterministic crowding with probabilistic replacement," Proceedings of the Genetic and Evolutionary Computation Conference, 1999, pp. 409-416.
- [7] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application, 1987, pp. 41-49.
- [8] Petrowski A, "A clearing procedure as a niching method for genetic algorithms," Proceedings of IEEE International Conference on Evolutionary Computation, 1996, pp.798-803.
- [9] Gan J and Warwick K, "Dynamic Niche Clustering: a fuzzy variable radius niching technique for multimodal optimisation in Gas," Proceedings of the 2001 Congress on Evolutionary Computation, 2001, pp. 215-222.
- [10] Beasley D, Bull DR and Martin RR, "A Sequential Niche Technique for Multimodal Function Optimization," Evolutionary Computation, 1993, pp. 101-125.
- [11] Li J.P., Balazs M.E., Parks G.T., et al, "A Species Conserving Genetic Algorithm for Multimodal Function Optimization," Evolutionary Computation, 2002, pp. 207-234.
- [12] Thomsen R, "Multimodal optimization using crowding-based differential evolution," Proceedings of the 2004 Congress on Evolutionary Computation, 2004, pp. 1382-1389.
- [13] LN De Castro and Timmis J, "An artificial immune network for multimodal function optimization," Proceedings of the 2002 Congress on Evolutionary Computation, 2002, pp. 699-704.
- [14] LN De Castro and Von Zuben FJ, "Learning and optimization using the clonal selection principle," IEEE Transactions on Evolutionary Computation, 2002, pp. 239-251.
- [15] Im C.H., Kim H.K., Jung H.K., et al, "A novel algorithm for multimodal function optimization based on evolution strategy," IEEE Transactions on Magnetics, 2004, pp. 1224-1227.
- [16] Kennedy J and Eberhart R, "Particle swarm optimization," IEEE International Conference on Neural Networks, 1995, pp. 1942-1948.
- [17] Kennedy J, "Stereotyping: improving particle swarm performance with clusteranalysis," Proceedings of the 2000 Congress on Evolutionary Computation, 2000, pp. 1507-1512.
- [18] Li T, Wei C and Pei W, "PSO with sharing for multimodal function optimization," Proceedings of the 2003 International Conference on Neural Networks and Signal Processing, 2003, pp. 450-453.
- [19] Parsopoulos KE and Vrahatis MN, "On the computation of all global minimizers through particle swarm optimization," IEEE Transactions on Evolutionary Computation, 2004, pp. 211-224.
- [20] Brits R, Engelbrecht AP and van den Bergh F, "A Niching Particle Swarm Optimizer," Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning, 2002, pp. 692-696.
- [21] Ozca and E., Yilmaz M, "Particle Swarms for Multimodal Optimization," ICANNGA 2007, Part I, LNCS 4431, 2007, pp. 366-375.
- [22] Seo, J.H., et al, "Multimodal Function Optimization Based on Particle Swarm Optimization," IEEE Transactions on Magnetics, 2006, pp. 1095-1098.
- [23] Li X, "A multimodal particle swarm optimizer based on fitness Euclidean-distance ratio," Proceedings of Genetic and Evolutionary Computation Conference, 2007, pp. 78-85.
- [24] Li X, "Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization," Proceedings of Genetic and Evolutionary Computation Conference, 2004, pp. 105-116.
- [25] IWAMATSU and Masao, "Multi-Species Particle Swarm Optimizer for Multimodal Function Optimization," I IEICE transactions on information and systems, 2006, pp. 1181-1187.
- [26] Bird S and Xiaodong Li, "Enhancing the robustness of a speciation-based PSO," IEEE Congress on Evolutionary Computation, 2006, pp. 843-850.
- [27] Alessandro Passaro and Antonina Starita, "Particle Swarm Optimization for Multimodal Functions: A Clustering Approach," Journal of Artificial Evolution and Applications, 2008, pp. 1-15.
- [28] Jang-Ho Seo, Chang-Hwan Im, Sang-Yeop Kwak, et al, "Multimodal Function Optimization based on Particle Swarm Optimization," IEEE TRANSACTIONS ON MAGNETICS, 2006, pp. 1095-1098.
- [29] Jang-Ho Seo, Chang-Hwan Im, Sang-Yeop Kwak, et al, "An Improved Particle Swarm Optimization Algorithm Mimicking Territorial Dispute Between Groups for Multimodal Function Optimization Problems," IEEE TRANSACTIONS ON MAGNETICS, 2008, pp. 1046-1049.
- [30] S. Bird and X. Li, "Adaptively choosing niching parameters in a PSO," Proceedings of the 8th annual conference on Genetic and evolutionary computation, Seattle, Washington, USA, 2006, pp. 3-10.
- [31] C. Worasuchee, "A particle swarm optimization for high-dimensional function optimization," Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010 International Conference on Electrical Engineering, 2010, pp. 1045-1049.
- [32] Xiaodong Li, "Niching Without Niching Parameters: Particle Swarm Optimization using a Ring Topology," IEEE Transactions on Evolutionary Computation, 2007, pp. 150-169.
- [33] Lizhong Xiao, Zhiqing Shao and Gang Liu, "K-means Algorithm Based on Particle Swarm Optimization Algorithm for Anomaly Intrusion Detection," The Sixth World Congress on Control and Automation in Intelligent, 2006, pp. 5854-5858.
- [34] E. C. Laskari, G.C. Meletioui, Y.C. Stamatiou, et al, "Evolutionary computation based cryptanalysis: A first study," Nonlinear Analysis, 2005, pp. 823-830.