Multiple-solution Optimization Strategy for Multi-robot Task Allocation

Li Huang, Yongsheng Ding, *Senior Member, IEEE*, MengChu Zhou, *Fellow, IEEE*,
Yaochu Jin, *Fellow, IEEE*, and Kuangrong Hao

Abstract—**Multiple solutions are often needed because of different kinds of uncertain failures in a plan execution process and scenarios for which precise mathematical models and constraints are difficult to obtain. This work proposes an optimization strategy for multi-robot task allocation (MRTA) problems and makes efforts on offering multiple solutions with same or similar quality for switching and selection. Since the mentioned problem can be regarded as a multimodal optimization one, this work presents a niching immune-based optimization algorithm based on Softmax regression (sNIOA) to handle it. A pre-judgment of population is done before entering an evaluation process to reduce the evaluation time and to avoid unnecessary computation. Furthermore, a guiding mutation operator inspired by the base pair in theory of gene mutation is introduced into sNIOA to strengthen its search ability. When a certain gene mutates, the others in the same gene group are more likely to mutate with a higher probability. Experimental results show the improvement of sNIOA on the aspect of accelerating computation speed with comparison to other heuristic algorithms. They also show the effectiveness of the proposed guiding mutation operator by comparing sNIOA with and without it. Two MRTA application cases are tested finally.**

Index Terms—**Multi-robot task allocation, multimodal optimization, niching immune-based optimization algorithm, Softmax regression, guiding mutation operator.**

I. INTRODUCTION

MULTI-robot cooperative systems attract much attention due to their distributed parallel processing abilities in recent decades. They are considered to be promising and of wide applications, such as path planning [1], exploration [2], tracking [3], foraging [4], and transportation [5], and have been applied to many domains like social science [6] and industrial engineering [7]. With the increasing demand for multi-robot cooperation in complex applications, the significance of multi-robot task allocation (MRTA), which is to determine an efficient and intelligent task assignment to improve the system performance [8], is recognized by more and more researchers [9-13].

A single optimal allocation plan may be insufficient. First, uncertain emergencies occur occasionally during actual execution processes of multi-robot cooperative systems, such that the predefined optimal task allocation schemes cannot be realized as expected. In the light of tackling different types of failures, outstanding work has been done to handle them. For the contention of resources, the interference is modeled when multiple robots use the same resource [14]. For dynamic settings of uncertain costs, a probabilistic cost representation is introduced to incorporate uncertainty and interdependency via distributional models [15]. For the varying coupling relationships, an ontology-based behavior modeling and checking system is proposed [16]. However, these methods can only cope with specific types of events. There is a lack of methods able to handle all kinds of unexpected failures. Hence a group of optimal assignments are highly desired, such that the predetermined scheme can be switched to another one when it cannot continue anymore. Second, precise mathematical models and constraints with comprehensive considerations are difficult to be summarized in some realistic application environments. Nevertheless, decision-makers are surely conscious of the feasibility of given solutions. Multiple solutions are needed as candidates. Finally, for a redundant circumstance, more robots are available for limited tasks. A group of assignment schemes can all satisfy the task requirement. Based on the above analysis, we conclude the necessity to provide multiple solutions for an MRTA problem.

Multiple optimal solutions are common in industrial manufacturing processes as well and have highly valued practical applications. First, apart from uncertain failures of a system, physical or energy constraints have impact on the long-term execution of a unique optimum as well. Next, multiple solutions with same or similar quality are important to foster the robustness and flexibility of a system and are useful for the sensitivity analysis of a problem. Lastly, multiple optimal solutions, as calculated by a centralized controller, can act as the prior information of a distributed multi-robot system to guide cooperative work and decrease the chances of resource conflict and system deadlock.

In general, this kind of problems belongs to the field of multimodal optimization. Classical heuristic algorithms, because of their rapid convergence to an optimum, likely a local one, cannot be applied to multimodal optimization problems

Li Huang, Yongsheng Ding, Yaochu Jin and Kuangrong Hao are with the Engineering Research Center of Digitized Textile and Apparel Technology, Ministry of Education, together with the College of Information Science and Technology, Donghua University, Shanghai 201620, China. Yaochu Jin is also with the Department of Computer Science, University of Surrey, Guildford, Surrey GU27XH, United Kingdom. (e-mail: huanglili622@126.com, ysding@dhu.edu.cn, yaochu.jin@surrey.ac.uk, krhao@dhu.edu.cn). Mengchu Zhou is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07032, USA (e-mail: zhou@njit.edu).

directly. Yet as a base, they may properly be combined with other techniques to deal with such problems well.

Commonly used algorithms include Genetic Algorithm (GA) [17], Particle Swarm Optimization (PSO) [18], Ant Colony Optimization (ACO) [19] and Differential Evolution (DE) [20]. The Big Bang-Big Crunch (BB-BC) algorithm is improved for multimodal optimization because of its low computational cost [21]. The Estimation of Distribution Algorithms (EDAs) are improved to deal with multimodal one by considering their advantages in preserving high diversity [22].

Niching methods are widely taken as ideal techniques to maintain a balance between the exploration and exploitation of multiple solutions [23]. Related researches are done from the perspective of overcoming shortcomings and enhancing performances of basic methods. A classical niching method requires setting a niche distance threshold [24]. For this problem, a Twin-space Crowding (TC) is introduced to build a parameter-free paradigm to eliminate the effect caused by a parameter value [25]. A distance-based Locally Informed Particle Swarm (LIPS) optimizer is proposed in [18] to eliminate the need for specifying any niching parameter and enhance the fine search ability of PSO. A PSO algorithm using a ring neighborhood topology, which does not require any niching parameters, is described in [26]. In order to avoid the time complexity caused by pairwise distance calculations, the work [27] proposes a fast niching technique by introducing the locality sensitive hashing, which is an efficient algorithm for approximately retrieving nearest neighbors. An improved information-sharing mechanism among individuals is introduced in [28] to induce more stable and efficient niching behavior, and a newly proposed parent-centric mutation operator is combined with a synchronous crowding replacement rule in [29]. Besides niching methods, new paths to gain the multimodal optimization ability are established. For example, in Gaussian Classifier-based Evolutionary Strategy (GCES) [30], multimodal optimization problems are regarded as classification ones, and the locations and basins of optima are saved by using Gaussian mixture models. In the cluster-based differential evolution [31], the clustering partition is used to divide the whole population into subpopulations for locating different optima.

An Artificial Immune System (AIS) is a parallel distributed self-adaptive system including features of evolutionary learning, pattern recognition and associative memory. Clonal Selection Algorithm (CSA), which evolves multiple populations simultaneously, is suitable for solving multimodal optimization problems. A hyper-mutation operator appears in CSA often. However, it is semi-blind and inefficient to handle complex circumstances. To overcome such drawback, the work [32] embeds Baldwinian learning and orthogonal learning in CSA. Thus, the hybrid learning CSA can perform effectively and robustly as expected. An immune-inspired affinity model is studied for the container multimodal transport emergency relief to schedule a multimodal transportation flow of the chain efficiently and reliably [33]. A biological notion in vaccines is introduced into AIS to lead antibodies to unexplored areas. Consequently, the exploration can be promoted in a search space for solving multimodal function optimization problems

[34]. A fusion of the immune network and predication performs well in regulating local and global search and guiding the determinate direction of local search, thus improving search ability for dealing with multimodal problems [35].

Most of the research achievements inspired by AIS for multimodal optimization are able to improve search ability, convergence speed and solution quality. In spite of some work like [33] caring about time efficiency as well, the existing methods are based on the idea of accelerating search processes. Considering characteristics of AIS and niching techniques, the computational complexity remains at a much-the-same level. In regard to the causes of the computational complexity, one of them is an expensive fitness evaluation process in complex environments and realistic applications, for example, our concerned MRTA problem.

In this work, considering that the fitness evaluation process is one of reasons causing much time consumption, Softmax regression [36] embedded in a Niching Immune Optimization Algorithm (sNIOA) is proposed to filter newly produced antibodies via pre-judgment based on historical information and reduce the number of individuals who enter an evaluation process. The computational procedure can be accelerated through the way of decreasing the time otherwise required for fitness evaluation. Furthermore, a Guiding Mutation (GM) operator is proposed to enhance the effective generation of new antibodies in order to improve the search ability of sNIOA, thereby resulting in its extension called sNIOA-GM. In comparison with sNIOA, the proposed sNIOA-GM is able to search in a larger space while requiring less computational resource. In simulation experiments, values of parameters are analyzed for better performance; comparisons with other multimodal optimization algorithms on benchmarks are conducted; and two scenarios of MRTA problems containing multiple solutions are tested.

The rest of this paper is constructed as follows: Section II presents the proposed sNIOA. Section III introduces the GM operator and sNIOA-GM. Section IV analyzes the multimodal optimization ability of sNIOA and influences of its parameters on its performance and compares it with other heuristic algorithms. It also presents the performance of the GM operator and shows task assignment results for MRTA problems. The conclusion and future work are given in Section V.

II. Niching Immune-based Optimization Algorithm based on Softmax Regression

The flowchart of the proposed sNIOA is shown in Fig. 1. First, initial antibodies are generated according to an antigen recognized, which represent initial solutions for a problem. Their fitness values are calculated through an evaluation process to determine affinity to the antigen. Then, these antibodies and their fitness compose the training set of Softmax regression. A pre-judgment model is built by it. New population, produced by operators of selection, mutation and crossing, is prejudged by this regression model such that non-ideal (low-quality) antibodies can be eliminated before entering an evaluation process. In this way, the time of fitness evaluation can be reduced. The niche technology based on a niching distance is used to maintain distribution and multimodal performance of solutions.
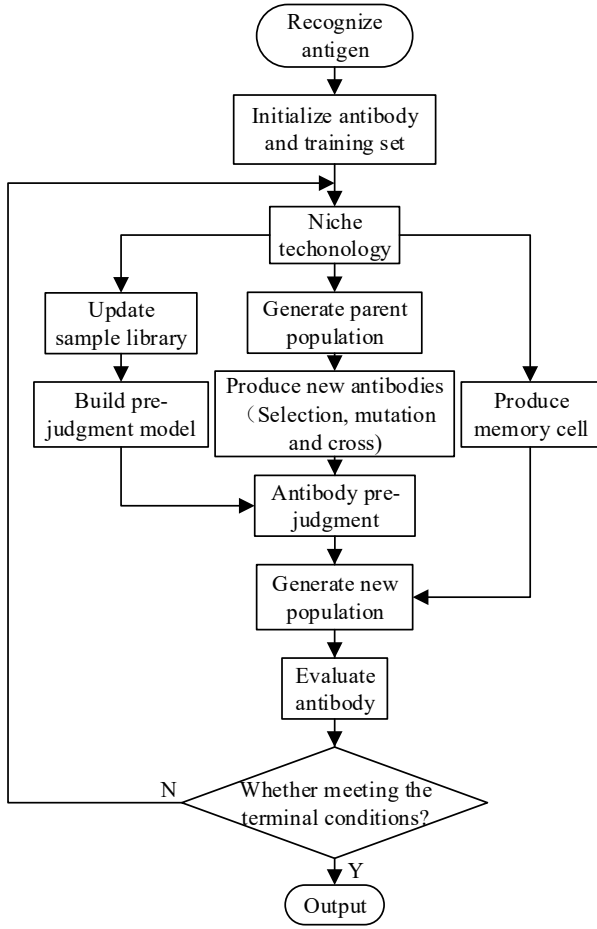
Fig. 1. The flowchart of sNIOA.

A. Immune-based Optimization Algorithm

AIS is described as the one that incorporates many properties of natural immune systems, including diversity, distributed computation, error tolerance, dynamic learning and adaptation, and self-monitoring [37]. Therefore, it offers effective and efficient inspiration for complex optimization problems. Based on the mentioned properties, a series of intelligent algorithms have been proposed. Their typical representative is an immune-based optimization algorithm (IOA).

IOA is a combination of algorithms based on an immune self-regulation mechanism and immune response [38]. It shares similarities with other immune algorithms and possesses unique and distinctive characteristics of keeping population diversity and converging to the optimum rapidly. Attributing to them, IOA is a preferable approach for multimodal optimization. Its implementation procedure is shown in Section A of the Supplementary File.

First, aiming at a recognized antigen, which corresponds to an objective function in an optimization problem, initial antibodies are generated randomly.

Then, individuals enter an evolution procedure based on affinity values with the antigen and keep diversity by considering concentration values among antibodies. This is an application of the immune system antibody concentration regulation principle, i.e., the antibody with worse values of fitness as well as concentration is suppressed during the process. Thus, the population diversity can be preserved along with the reservation of matched antibodies. More specifically, the evaluation phase is based on three indexes of individuals. The first one is fitness, which corresponds to the value of an objective function in optimization. The second one is concentration, which indicates population diversity. The third one is excellence, which is calculated based on their fitness and concentration:

$$I_{e_i} = I_{p_e} \frac{I_{f_i}}{\sum_{i=1}^{n} I_{f_i}} + \left(1 - I_{p_e}\right) \frac{I_{c_i}}{\sum_{i=1}^{n} I_{c_i}}, \qquad (1)$$

where I_{f_i}, I_{c_i} and I_{e_i} denote values of fitness, concentration and excellence of individual i, respectively; and I_{p_e} is the multiplicity evaluation parameter to balance fitness and concentration as a weight.

Next, excellent antibodies constitute the parent population and the best part of these is kept in the memory cell. The memory unit helps the corresponding antibodies for an antigen, which has come up, to be generated faster than before.

After that, new population is generated through three phases. The first one is selection that is based on the roulette selection. The individual with a greater value of excellence is chosen with a higher probability. The second one is mutation used to produce new individuals. It can search more space. The third one is crossing that contributes to reserving gene segments of good individuals and passing them on to future generations.

Two terminal conditions are used in this work. One is the limited number of iterations. The other is that the average fitness value of individuals reaches a predetermined threshold.

B. Niche Technology

The niche technology is an abstraction from a biological phenomenon. In an ecological environment, fierce competitions among intraspecific creatures may occur sometimes for limited resources. However, such a relationship does not appear among different species. In this way, in the process of population evolution, the fittest one can be preserved in a certain kind of creatures. At the same time, the species diversity can be maintained. The introduction of a niche technology to IOA is able to equip IOA with multimodal optimization ability while maintaining diversity of solutions.

There are a variety of ways to implement niche techniques. This work uses the one based on niche distance. First, a niche distance threshold is defined. Then, the distance between each pair of individuals is calculated. If the calculated distance is less than the threshold, the individual that has worse fitness is inflicted with a penalty. Thus it may be eliminated with a larger probability later on. In other words, there is only one satisfactory solution within the range of some niche distance. The pseudo-code of this niche technology based on niche distance for a maximization problem is shown in Section B of the Supplementary File.

The niche distance is an important parameter of the niche technology and has great impact on multimodal optimization performance. A large niche distance may lead to non-convergence of an algorithm. On the contrary, a tiny one can result in non-uniformity of solutions, especially for the function with continuous solution intervals, even missing the optima.

C. Softmax Regression

At the beginning of sNIOA, antibodies are initialized and their fitness values are computed. These values are contiguous. Hence, a fitness threshold should be given in order to replace fitness values by class labels. The fitness threshold t is called classification threshold and defined as:

$$\begin{cases} \hat{\xi} = \hat{f} - \hat{\beta}\left(\hat{f} - \overline{f}\right), & \textit{for } \text{maximization} \\ \breve{\xi} = \breve{f} - \breve{\beta}\left(\breve{f} - \overline{f}\right), & \textit{for } \text{minimization} \end{cases}, \quad (2)$$

where f is the fitness value of an antibody and β is the parameter of a classification threshold. For maximization problems, an antibody whose fitness is more than $\hat{\xi}$, is expected to be preserved. For minimization problems, an individual whose fitness is less than $\breve{\xi}$, is a better choice. In our concerned MRTA problems, the individuals with better fitness values are regarded as ideal ones. Others are called non-ideal antibodies. After being labeled, the initialized individuals and corresponding labels are input to a Softmax training process as a sample database and then a prediction model is obtained.

During iterations, when new antibodies are produced after selection, mutation and crossing operations, their qualities are determined via the mentioned prediction model. The non-ideal (low-quality) individuals are eliminated directly. The ones with predicted high qualities enter an evaluation process to obtain their precise fitness values.

The newly computed antibodies and their labels are embedded into the sample library of Softmax regression. The classification threshold is recomputed, and the individuals are relabeled with the new threshold to raise the standard of better individuals gradually. The description of Softmax regression is given in Section C in the Supplementary File.

III. Guiding Mutation Operator for sNIOA

In this section, the GM operator is proposed to strengthen the optimization ability of sNIOA, which is inspired by the theory of gene mutation [39], thus yielding sNIOA-GM. In accordance with the mutational pattern of a base, gene mutations can be divided into frameshift mutations and base substitutions. The former is caused by the insertion or deletion of a number of nucleotides. For example, a segment of genes "A T C T" changes to "A T A C T" by the insertion of "A" or "A T T" by the deletion of "C". The latter therein refers to a kind of mutations that one base pair is replaced by another different base pair. The GM operator is inspired by base substitutions.

Many of base substitutions originally derive from the mutation of a single base. When a single base is changed, another base in this base pair is also replaced by others in the next gene replication. For example under the influence of nitrites, a cytosine (C) mutates into a uracil (U) stimulated by the oxidative deamination. During the gene replication, the uracil (U) does not pair up with a guanine (G) but with an adenine (A). As a result, the C-G base pair mutates into a T-A base pair as shown in Fig. 2.
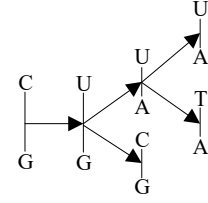

Fig. 2. Example of base substitution under the influence of nitrites.

With the inspiration mentioned above, dividing one antibody into several "gene pairs" is taken into consideration. A traditional gene pair consists of two bases. Herein, the number of genes in a "gene pair" may be more than two. In other words, the wording of "gene group" should be more exact. When a certain gene mutates, the other genes in a same group have a higher probability of mutation.

The definition of a gene group also makes sense to our MRTA problem. For example, a proper group among many available robots has been chosen but with the inferior connections with tasks as shown in Fig. 3(a), where triangles stand for tasks and circles represent robots. In this case, it is less helpful if there is only one assignment line being changed as shown in Fig. 3(b). When these tasks are regarded as a gene group, others have a higher probability of mutation when one of them is altered. Thus the combination has more chances to be modified comprehensively as shown in Fig. 3(c).
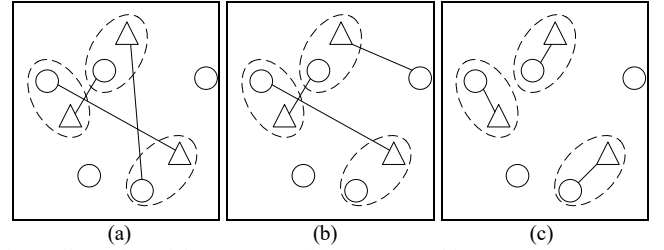

Fig. 3. Illustration of the gene group in a MRTA problem.

In the above-mentioned scenario, the formation of a gene group mainly depends on the locations of tasks. In other words, tasks which are relatively close to each other tend to be in a same gene group. So, the Mean-shift algorithm is used to do this herein. Some examples are shown in Section D of the Supplementary File. Different kinds of symbol shapes represent different gene groups. The results of clusters can adapt to different application demands by tuning the threshold in the Mean-shift algorithm.

After the creation of gene groups, individuals are determined to be whether mutated or not depending on mutation probability p_m. In the process of mutation, when one gene of an individual is chosen to be changed, each of the rest of genes in the same group continues to be decided with a higher mutation probability p_{m_h}. The GM operator is summarized as shown in Algorithm I.

Based on above discussions, the procedures of sNIOA-GM for MRTA problems are proposed as shown in Algorithm II.

Given population size N, sample library size S and gene group size G, the GM operator costs $O(N \times G) \approx O(N)$, where $G \ll N$. The niche technology takes $O(N^2)$. The training and predicting processes of Softmax regression spend

$O(S)$ and $O(N)$, respectively. In summary, the proposed sNIOA-GM's complexity is $O(N^2)+O(S)$.

Furthermore, the entire algorithm's complexity depends on the objective function being optimized as well. If it is a simple function, it takes $O(N)$ to evaluate the population. However, when it comes to a complex context, especially appearing in real applications, the evaluation process is complicated. When its complexity exceeds $O(S)$, the proposed algorithm can exhibit its great advantage than those without using Softmax regression.

ALGORITHM I. GUIDING MUTATION OPERATOR
1: Base groups creation;
2: For each individual in population, generate a random number r;
3:    If $r < p_m$
4:       Choose one base randomly and activate corresponding base group;
5:       Base mutation;
6:       For each of the rest of the base group, generate a random number r;
7:          If $r < p_{m_h}$
8:             Base mutation;
9:          End if
10:       End for
11:    End if
12: End for

ALGORITHM II. sNIOA-GM
Step 1: Generate initial solutions according to the problem statement. Calculate fitness values as the initial training set.
Step 2: Use the niche technique to obtain multimodal optimization performance.
Step 3: Evaluate the excellence e_i of each ideal individual by the combination of fitness f_i and concentration c_i through a multiplicity evaluation parameter p_e. Stop if the termination criterions are met.
Step 4: Update the training set and create the prediction model.
Step 5: Preserve good solutions in the memory cell.
Step 6: Generate new population through the way of selection, GM operator and crossing.
Step 7: Use Softmax regression to eliminate non-ideal solutions. Others are preserved and combined with the memory cell as the parent generation.
Step 8: Increase the iteration count by one and go to Step 2.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, the multimodal optimization ability of the proposed sNIOA is shown by comparing with other popular heuristic algorithms. This work provides the simulation results of sNIOA-GM for two MRTA application cases.

A. Multimodal Optimization Ability of sNIOA

In this phase, the multimodal optimization ability of sNIOA is tested to ensure the existence of alternative solutions. Schaffer's F6 function is a typical multimodal function described by

$$f(x,y)=0.5+\frac{\sin^2\left(\sqrt{x^2+y^2}\right)-0.5}{\left[1+0.001\left(x^2+y^2\right)\right]^2}. \tag{3}$$

In order to make the solution space including multiple equivalent optimal areas, $0.001(x^2+y^2)$ is set to 0. That is to say, the oscillations of function do not attenuate and each of the peaks is a continuous interval. With the purpose of a clear

observation, this optimization problem is considered to search for its maxima. The range of x and y is [-4, 4].

A group of random initial solutions are shown in Fig. 4(a). The optimization results of IOA and sNIOA are shown in Fig. 4(b) and Fig. 4(c), respectively.

(a) Solution space of test function and initial solutions

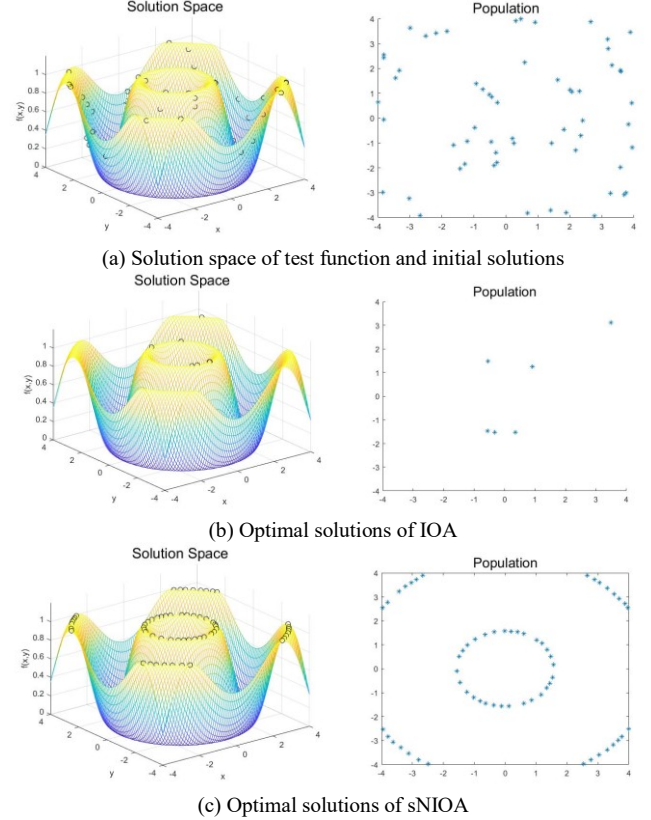(b) Optimal solutions of IOA

(c) Optimal solutions of sNIOA

Fig. 4. Illustration of multimodal optimization ability.

According to Fig. 4, both IOA and sNIOA can find the optimization solutions of the revised Schaffer's F6 function. The result obtained by IOA, however, converges to one optimal solution. To the contrast, obtained solutions of sNIOA can distribute over the entire optimal solution intervals uniformly. The theoretical reason is that, when an excellent antibody is once generated, other antibodies are influenced by this excellent one through the crossing operation during iterations. Thus, they all converge to the excellent one in the end of IOA. In sNIOA, a niche distance is defined. Because of this, there is a fixed distance between each pair of candidate solutions, such that solutions cannot gather to one point only. The training and predicting processes, which are introduced to reduce the evaluation time in sNIOA, do not have an effect on this performance. To sum up, the uniform multimodal optimization ability of sNIOA can be guaranteed.

B. Analysis of Parameters in sNIOA

The niche distance is a significant parameter of sNIOA. If it is too small, it is difficult for sNIOA to find all optimal solutions and leads to a premature convergence. On the other hand, large niche distance is unfavorable for functions that have steep peaks. In addition, the uniformity of optimal solutions in the same continuous interval is also related to the distance parameter.

The classification threshold as a parameter also impacts

optimization results. A loose threshold lets more ordinary antibodies enter the evaluation process such that the advantage of a fast computational speed cannot be shown dramatically. In contrast, a tight threshold may impose restrictions on population diversity.

These two parameters mentioned above have great influence on optimization performances of sNIOA. In this part, the impacts of niche distance and classification threshold are discussed through tests on the Branin function. Their values and resulting experimental results are shown in TABLE 1.

TABLE 1
PARAMETERS ANALYSIS EXPERIMENTAL RESULTS

Parameters		Experimental Results				
μ_n	$\breve{\beta}$	Deviation of Optimal Value	Deviation of Mean Value	Standard Deviation	Evaluation Time	Actual Peak Number
0.02	1/12	1.64E-03	0.0541	0.0309	22739.6	3
	1/3	7.38E-03	0.1884	0.1192	14616.2	3
	2/3	4.96E-03	0.2953	0.1744	9018	3
	11/12	5.80E-03	0.0696	0.0397	10635.4	3
0.05	1/12	1.78E-03	0.1337	0.0963	22432.4	3
	1/3	5.30E-03	0.2493	0.1530	16547.2	3
	2/3	7.20E-03	0.3719	0.2196	10296.2	3
	11/12	6.08E-03	0.2028	0.1179	10308	3
0.1	1/12	1.98E-03	0.2493	0.1739	22287.4	3
	1/3	5.20E-03	0.4675	0.4395	15172.2	3
	2/3	1.48E-03	0.5104	0.3103	10473.2	3
	11/12	1.25E-03	0.3817	0.2861	11317.4	3
0.2	1/12	3.78E-03	0.8018	0.5330	22890	3
	1/3	4.50E-03	0.9707	0.6108	15034.6	3
	2/3	5.10E-03	1.1528	0.6932	11694.4	3
	11/12	7.48E-03	1.0844	0.7796	9778	3

According to TABLE 1, all the experimental combinations can find all optimal peaks. The deviations of optimal value approach to 0 almost, i.e. sNIOA can find all optimal solutions of this test function and achieve the optimization goal. Furthermore, we use the prior computation time of NIOA as a reference, which is 24080 evaluation times. From TABLE 1, the times of fitness evaluation of all experimental combinations are less than 24080. It proves that sNIOA can reduce the evaluation time and show a satisfied computational performance.

Next, the effect of classification threshold is analyzed. According to above discussions, this parameter mainly has impact on the time of computation. This conclusion is verified by Section E in the Supplementary File, that the evaluation time decreases with the raise of classification threshold parameter. The reason for this is that, in the process of Softmax training for minimizing problems, an individual is regarded as a preferable antibody when its fitness is less than $\breve{\zeta}$. When $\breve{\beta}$ increases in the experiments, $\breve{\zeta}$ goes down, i.e. only antibodies whose fitness is much smaller than others can be preserved at the end of Softmax prediction. In consequence, fewer individuals go to the evaluation process, such that the evaluation time can be cut down drastically, e.g., 62.55%. The bigger value of this parameter, however, may not be better. When this value increases to more than 2/3, the evaluation time has a slight drop or remains unchanged till the end and even increases in some cases.

The niche distance mainly has effect on the deviation of

mean value and standard deviation of optimal solutions. Along with the niche distance's growth, both deviations are on the rise as shown in Section F of the Supplementary File. The explanation of this trend is that, when the niche distance increases, optimal solutions are more scattered around the peaks. Therefore, a bigger deviation is caused. Nevertheless, a smaller niche distance may lead to non-uniformity of solutions, especially for the problem which has continuous solution intervals, even missing optima sometimes.

To sum up, for different problems, the specific combination of parameters should be designed accordingly.

C. Comparison with other heuristic algorithms

In order to prove its efficiency, we compare sNIOA with other popular heuristic algorithms on test functions next.

These algorithms are all proposed for solving multimodal optimization problems including NIOA, Niching Genetic Algorithm (NGA) and Niching Particle Swarm Optimization (NPSO). They are selected because of their validated performance [40-42] and use of the niching technique. The parameters in each algorithm are shown in TABLE 2, where N is population size; L is the length of binary coding; I is the number of iterations; M is the capacity of memory; p_c and p_m are probabilities of crossing and mutation, respectively; s is the parameter that balances fitness and concentration of an antibody; $\hat{\beta}$ and $\breve{\beta}$ are parameters of classification threshold for maximization and minimization problems, respectively; μ_n is the niche distance; ω_{max} and ω_{min} are the ranges of the inertia weight; and c_1 is the learning factor.

TABLE 2
PARAMETERS IN COMPARISON EXPERIMENTS OF EACH ALGORITHM

sNIOA		NIOA		NGA		NPSO	
N	100	N	100	N	100	N	100
L	22	L	22	L	22	L	22
I	200	I	200	I	200	I	200
M	20	M	20	p_c	0.9	ω_{max}	1
p_c	0.9	p_c	0.9	p_m	0.05	ω_{min}	0
p_m	0.3	p_m	0.3	-	-	c_1	2
s	0.15-0.95	s	0.15-0.95	-	-	-	-
$\hat{\beta}$	$\dfrac{2}{3}$	μ_n	0.01-0.1	-	-	-	-
$\breve{\beta}$	$\dfrac{1}{3}$	-	-	-	-	-	-
μ_n	0.01-0.1	-	-	-	-	-	-

All parameters are selected carefully to ensure that each algorithm performs well. For example, the mutation probability p_m appears both in sNIOA, NIOA and NGA. However, the one used for NGA is 0.05 and is much less than 0.3 used in sNIOA and NIOA. We try several values of the mutation probability and choose the one that leads to the best performance of each algorithm by testing them on the revised Schaffer's F6 function. The explanation can be offered from the theoretical perspective. In NGA, since the final goal is to bring the population to convergence, selection and crossing happen very often. Mutation, attempting to occasionally break a number of individuals out of local optima as a way to maintain

diversity, should occur less frequently. In sNIOA and NIOA, there is a memory cell preserving well-behaved individuals in each generation inspired by AIS. Thus the excellent solutions are not eliminated in the following iterations when increasing the probability of searching more areas. As a result, a large p_m is preferred.

Test functions are shown in TABLE 3. Within specified limits, the revised Schaffer's F6 function includes multiple continuous global optimization intervals. Himmelblau function and Branin function have multiple global optimal peaks without local optima. Rosenbrock function is actually a unimodal function with a continuous gradual interval beside the peak. There are two optimal solutions and four local optima in Six-Hump Camel Back function. Cross-in-Tray function and Rastrigin function include four and one global optimal peak, respectively, both with many local optima. Shubert function includes two global optimal peaks, a small number of local optima and many slight fluctuation peaks. Note that when the minima of a function are negative, we move this function to upward. Thus, the lower boundaries are 0 for all test functions. These functions cover various kinds of characteristics of multimodal functions, such as continuous intervals and steep peaks, global optima with and without local optima and maximization and minimization problems. They are also chosen as commonly-used benchmarks by many researchers [42-44]. Therefore, we use them as our benchmarks.

The experimental results keep a record of the deviation from the optimal value, deviation from mean value, standard deviation, evaluation time and number of peaks found as shown in TABLE 4. Here the deviation from the optimal (mean) value is the difference between the optimal (mean) values obtained from optimization algorithms and theoretical optima. The standard deviation is used to measure stability of solutions. The evaluation time indicates the time performance. The number of peaks found is applied to evaluate the multimodal optimization

ability. In addition to the number of iterations, the termination condition of each algorithm also takes the deviation from the mean value into consideration. In this way, meaningless calculation does not proceed when the optimal value cannot be improved any more during iterations. All of the data are mean values of ten times repeated trials.

From this table, the excellent time performance of sNIOA can be seen in comparison with other heuristic algorithms. The proportion of evaluation time reduces on average by 41.01%, 38.38% and 39.34% compared with NIOA, NGA and NPSO, respectively. Its maximum time reduction can reach 53.78%, 68.74% and 69.63% in comparison with its three peers. Its other performance measures fall in normal ranges of optimization results. Some of its performances are the best. It is easy to conclude that sNIOA can speed up the computation dramatically while showing comparable optimization performance in comparison with its three peers.

From the point of view of multimodal optimization ability, sNIOA is able to find all optimal peaks. Antibodies do not converge to local optima in most of the functions due to the repulsion of the niche distance, except Rastrgin function with a large number of local extremums. However, the limitation of sNIOA cannot be denied, i.e., the non-uniformity of individuals is evident when peaks are steep. The experimental results show that NGA has strong convergence ability. Because of this, NGA's ability to search unknown optimal peaks is reduced as shown in Himmelblau and Branin functions in TABLE 4. The search ability of NPSO is uncertain and unstable. It sometimes results in convergence to local optima as shown in Branin, Six-Hump Camel Back and Shubert functions.

From the perspective of three deviation values, sNIOA performs noticeably well in the experiments of revised Schaffer's F6, Branin and Shubert functions. Other results are of the same order of magnitude and approach to the optimal values.

TABLE 3
TEST FUNCTIONS AND PARAMETERS

Test Function	Equation	Range	Max	Min
Revised Schaffer's F6	$f(x,y) = \sin^2\left(\sqrt{x^2 + y^2}\right)$	$-4 \leq x, y \leq 4$	1	0
Himmelblau	$f(x,y) = \left(x^2 + y - 11\right)^2 + \left(x + y^2 - 7\right)^2$	$-6 \leq x, y \leq 6$	2200	0
Branin	$f(x,y) = \left((y+10) - \frac{5.1(x+5)^2}{4\pi^2} + \frac{5(x+5)}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x+5) + 10$	$-10 \leq x, y \leq 10$	250	0
Rosenbrock	$f(x,y) = 100\left(y - x^2\right)^2 + \left(x - 1\right)^2$	$-2 \leq x, y \leq 3$	2200	0
Six-Hump Camel Back	$f(x,y) = \left(4 - 2.1x^2 + \frac{x^4}{3}\right)x^2 + xy + \left(-4 + 4y^2\right)y^2$	$-3 \leq x, y \leq 3$	5	0
Cross-in-Tray	$f(x,y) = -0.0001\left(\left|\sin x \sin y e^{\left|100 - \frac{\sqrt{x^2+y^2}}{\pi}\right|}\right| + 1\right)^{0.1}$	$-4 \leq x, y \leq 4$	1	0
Shubert	$f(x,y) = \left(1 + \cos(2x+1) + 2\cos(3x+2) + 3\cos(4x+3) + 4\cos(5x+4) + 5\cos(6x+5)\right)$ $\left(1 + \cos(2y+1) + 2\cos(3y+2) + 3\cos(4y+3) + 4\cos(5y+4) + 5\cos(6y+5)\right)$	$-2 \leq x, y \leq 2$	350	0
Rastrgin	$f(x,y) = 20 + \left(x^2 - 10\cos 2\pi x\right) + \left(y^2 - 10\cos 2\pi y\right)$	$-5 \leq x, y \leq 5$	80	0

TABLE 4
EXPERIMENTAL RESULTS ON TEST FUNCTIONS OF DIFFERENT HEURISTIC OPTIMIZATION ALGORITHMS

Test Function	Optimal Value	Optimization Algorithm	Deviation of Optimal Value	Deviation of Mean Value	Standard Deviation	Evaluation Time	Actual (Ideal) Peak Number
Schaffer's F6	1 (max)	sNIOA	1.69E-08	9.59E-05	**1.08E-04**	**6195**	5(5)
		NIOA	1.76E-08	**9.51E-05**	1.19E-04	7163	5(5)
		NGA	1.90E-13	4.81E-04	1.76E-03	19820	5(5)
		NPSO	**0**	1.27E-02	7.76E-02	20400	5(5)
Himmelblau	0 (min)	sNIOA	1.70E-03	1.3200	0.9378	**11553.9**	4(4)
		NIOA	4.54E-04	**0.9820**	**0.6493**	24080	4(4)
		NGA	**3.20E-04**	1.0500	0.7300	20100	3.5(4)
		NPSO	3.07E-03	4.0534	3.9987	20400	4(4)
Branin	0.397887 (min)	sNIOA	3.27E-03	**0.2525**	**0.1581**	**14346.1**	3(3)
		NIOA	1.77E-04	0.3044	0.2049	24080	3(3)
		NGA	**5.44E-05**	0.2553	0.1686	20100	2.8(3)
		NPSO	2.58E-03	1.3908	0.9989	20400	4(3)
Rosenbrock	0 (min)	sNIOA	7.72E-03	0.2574	0.1508	**14036.2**	1(1)
		NIOA	6.14E-02	0.3093	**0.0894**	24080	1(1)
		NGA	1.06E-02	**0.2066**	0.1431	20100	1(1)
		NPSO	**5.50E-04**	0.4576	0.4708	20400	1(1)
Six-Hump Camel Back	-1.0316 (min)	sNIOA	1.21E-03	0.0953	0.0655	**12280.5**	2(2)
		NIOA	8.61E-05	0.0739	0.0509	24080	2(2)
		NGA	**1.97E-05**	**0.0275**	**0.0185**	20100	2(2)
		NPSO	5.69E-04	0.4158	0.3252	20400	3.8(2)
Cross-in-Tray	-2.0626 (min)	sNIOA	4.29E-05	2.93E-03	2.03E-03	**11130.1**	4(4)
		NIOA	**1.19E-06**	2.05E-03	1.33E-03	24080	4(4)
		NGA	1.86E-05	**1.08E-03**	**8.33E-04**	20100	4(4)
		NPSO	1.71E-05	1.23E-02	1.54E-02	20400	4(4)
Shubert	-186.73 (min)	sNIOA	6.81E-02	**7.8764**	**5.0709**	**14071.5**	2(2)
		NIOA	1.18E-03	8.4319	5.2265	24080	2(2)
		NGA	**3.32E-04**	12.6556	8.5062	20100	2(2)
		NPSO	1.65E-01	106.8010	52.1314	20400	15.5(2)
Rastrgin	0 (min)	sNIOA	7.53E-02	3.2800	1.2984	**15389.1**	14.6(1)
		NIOA	1.94E-03	2.9933	1.2430	24080	13.5(1)
		NGA	**5.53E-09**	**2.2137**	**0.8807**	20100	9(1)
		NPSO	4.52E-01	6.8802	3.8715	20400	27.4(1)

Next, we illustrate if there are significant differences between sNIOA and other heuristic algorithms. The significant difference p value at a two-tail t-test is given with the significance level of 5%. The null hypothesis is that there are no significant differences between sNIOA and other algorithms. As the results of four indexes shown in TABLE 5 for the revised Schaffer's F6 function as an illustrative example, if the p value is smaller than 0.05, the null hypothesis is rejected, i.e. there is a significant difference between two algorithms on the corresponding index. The four indexes are divided into two parts. The first part describes the optimization ability, which consists of the deviation of optimal value, the deviation of mean value and the standard deviation.

TABLE 5
T-TEST RESULTS ON SCHAFFER'S F6 FUNCTION

Compared Optimization Algorithms	p value			
	Deviation of Optimal Value	Deviation of Mean Value	Standard Deviation	Evaluation Time
sNIOA vs NIOA	0.95554	0.66327	0.10946	**0.040021**
sNIOA vs NGA	0.15213	**0.0043328**	**0.037891**	3.4412e-19
sNIOA vs NPSO	0.15214	**5.1572e-05**	**1.3226e-07**	6.4315e-13

As shown in TABLE 5, some results obtained from sNIOA, for example the deviations from the mean value of sNIOA and NGA, show significant differences. However, accompanying with results shown in TABLE 4, the corresponding index of sNIOA is better than NGA's. Therefore, a conclusion can be obtained that sNIOA has at least the same optimization ability as other heuristic algorithms do. The second part describes the

time performance represented by the evaluation time. Regarding this index, sNIOA performs the best on computation speed as shown in TABLE 4. By means of the hypothesis testing, significant differences are verified between sNIOA and other algorithms as shown in TABLE 5. To sum up, considering all experimental results synthetically, sNIOA achieves the original intention of this work with competitive multimodal optimization performances, and clear advantage on computation speed over its three peers.

Next, two application cases are provided to illustrate the merits of sNIOA-GM in solving MRTA problems.

D. Application Case 1

Assume that there are q robots $R = \{r_1, r_2, \cdots, r_q\}$ and s tasks $T = \{t_1, t_2, \cdots, t_s\}$. At a certain moment each task needs to be executed by one robot and each robot can only perform one task. κ_{ij} is a binary variable and presents the allocation relationship between robot r_i and task t_j. c_{ij} is the execution cost of task t_j by robot r_i. Denote the priority level of task t_j as p_j. A smaller p_j implies a higher priority of task t_j. The MRTA problem can be stated as follows.

$$\min \; f = \sum_{j=1}^{s} c_{ij} \kappa_{ij} p_j \qquad (4)$$

$$s.t. \; \sum_{j=1}^{s} \kappa_{ij} = 1, \; \forall i = 1, 2, \cdots \qquad (5)$$

$$\sum_{i=1}^{q}\sum_{j=1}^{s}\kappa_{ij}=s, \ \ i=1,2,\cdots \qquad \cdots \qquad (6)$$

$$s \le q \qquad (7)$$

$$\kappa_{ij} \in \{0,1\} \qquad \forall i,j \qquad (8)$$

In this scenario, ten groups of simulations are done with the number of tasks increasing from 10 to 100 and the number of robots increasing from 20 to 200. The tasks have different priority levels from 1 to 5. Tasks and robots are dispersed over a 100×100 square area randomly. All the experimental combinations are repeated for 10 times. Then average values are taken as the final results.

The performances of sNIOA-GM are discussed from two perspectives: run time and obtained cost, respectively. For the former, due to the pre-judgment of Softmax regression, sNIOA and sNIOA-GM both show better time performance than the original IOA as shown in Fig. 5(a). Regarding the cost, the GM operator can account for the elimination of second half iterations of sNIOA-GM as shown in Fig. 5(b). When the number of tasks is small, optimal solutions can be found under the same conditions by all of these three methods. However when it comes to large-scale task allocation problems, in the case of same limited number of iterations which is 200, the GM operator can make antibodies mutate toward a better direction and help find the preferable solutions, while solutions given by IOA and sNIOA cannot converge to optimal values.

In closing, sNIOA-GM can realize effective optimization in the concerned MRTA scenarios by achieving rapid computation speed and better solutions, especially for large-scale problems.
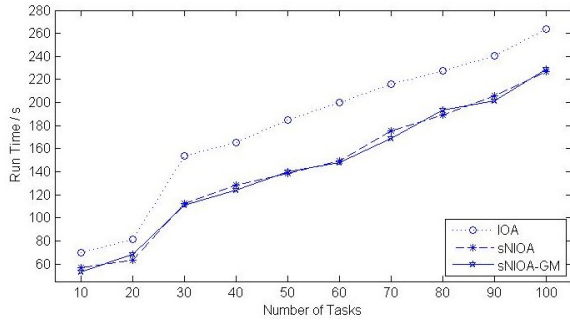
(a) Run time
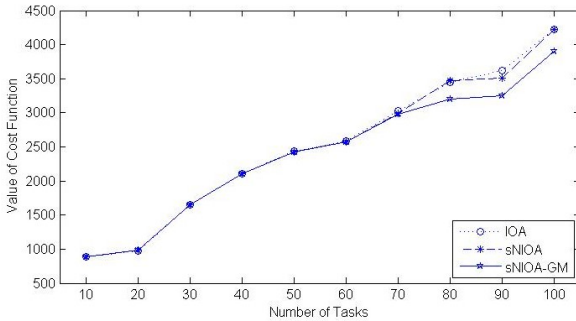
(b) Value of cost function

Fig. 5. Comparison between IOA, sNIOA and sNIOA-GM.

E. Application Case 2

Assume that there are w robot stations. For each station, g heterogeneous robots are expected to be available. The total number of robots is $q=wg$. All of the robots are labeled with number 1 to q. There are s tasks scattered on an area waiting for execution labeled with number 1 to s. Each task needs to be executed by one robot. One robot can perform multiple tasks in turn. The goal is to assign robots to perform tasks and design the execution sequence of each robot for the sake of minimizing total energy consumption, which is mainly related to the travelling distances of robots. The concerned context can be proved to be an NP-hard problem [45]. The mathematical model is built as follows.

Let A_{φ} denote the φth robot station. Given a set T, $|T|$ is the number of members in T. A robot r_i is one of the robots departing from station A_{φ} to accomplish a sequence of assigned tasks $T_i = \{t_i^1, t_i^2, \cdots \ t_i^{|T_i|}\}$ to r_i and return to the same place after finishing its jobs. Its destination set $D_i = \{A_{\varphi}, T_i, A_{\varphi}\}$ is labeled with number 1 to $|D_i|$. δ_{uv} is taken as the distance between destinations u and v. ζ_{uv} is defined as the binary variable representing the relationship that the path goes from u to v. Then the concerned MRTA problems can be formulated as follows:

$$\min \ f = \sum_{i=1}^{q}\sum_{u=1}^{|D_i|}\sum_{v\ne u,\, v=1}^{|D_i|}\delta_{uv}\zeta_{uv} \qquad (9)$$

$$s.t. \quad \bigcup_{i=1}^{q} \quad , \qquad (10)$$

$$\zeta_{uv} \in \{0,1\}, \forall u, v , \qquad (11)$$

$$\zeta_{A_i v} + \zeta_{u A_i} = 1 . \qquad (12)$$

For solving the above-mentioned problem by an immune-inspired method, an antibody is initialized as a vector whose length is s as shown in Fig. 6. Each position of the antibody represents the label of a robot being assigned to accomplish the task. One robot can be allocated to multiple tasks.
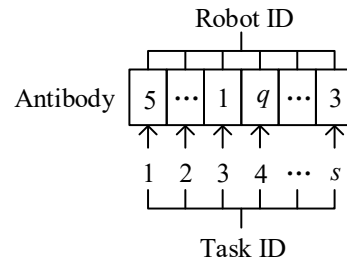

Fig. 6. An antibody of sNIOA-GM for MRTA.

For this MRTA problem, the procedure to compute the total cost is given as Algorithm III. The population size is N. In order to obtain the sequence that a robot performs a set of tasks, the Simulated Annealing Algorithm [46] is used to design a shortest path and its number of iterations is L. Therefore, the evaluation process needs $O(N\times q\times L)$, which is approximately equal to $O(N^2)$. Comparing with $O(N)$, which a simple objective function takes, it is reasonable to expect that the computational cost can be decreased when the time of fitness evaluation is shortened in this problem.

ALGORITHM III. FITNESS COMPUTATION FOR APPLICATION CASE 2
1: Population size is N ;
2: **For** $i=1:N$
3:    *Total distance* $=0$;
4:    Collect assigned tasks for each robot;
5:    **For** $j=1:q$
6:        Compute the shortest path with use of Simulated Annealing algorithm (Number of iterations is L);
7:        *Total distance* $=$ *Total distance* $+$ *Shortest Path* ;
8:    **End for**
9:    Return *Total distance* as the fitness value for i^{th} antibody;
10: **End for**

There are three robot stations A_1, A_2 and A_3 in an environment located on [0, 70], [50, 0] and [100, 30], respectively. Three robots at each station are used to deal with 20 tasks scattered on a 100×100 area. C_1^A, C_2^A and C_3^A denote the number of required robots from each station, respectively. Multiple solutions and related data are given in TABLE 6 and Section G of the Supplementary File. There are different kinds of allocation ways with different numbers of robots dispatched from different robot stations. According to these results, decision makers can choose one from them depending on their situations.

For example, in this application case, the number of robots is three at each station. However, when a set of tasks arrives, some robots may be charged or have been already dispatched to execute other jobs. It results in the reduction of available robots. In traditional approaches, managers collect data, build models and compute solutions at that time. This process consumes much time before starting execution and thus delays the completion of tasks. Besides, it needs to be done every time as long as tasks arrive. With the use of our method, a group of strategies covering different cooperative ways between robot stations and execution paths can be obtained previously according to the original information that each station has three robots. In such circumstances, decision makers can easily choose one from them regardless of the number of real-time available robots.

Furthermore, it is also helpful to avoid complex modeling of robot energy. For example, considering TABLE 6 and Section G of the Supplementary File, if a robot in station A_2 has insufficient power to finish a large-scale set of tasks, strategies (c), (e) and (f) are more appropriate than (a) even though the latter one has the shortest path length.

TABLE 6
EXPERIMENTAL RESULTS OF APPLICATION CASE 2

No.	Path Length	C_1^A	C_2^A	C_3^A
(a)	476.36	1	1	1
(b)	520.44	1	2	1
(c)	535.55	1	1	2
(d)	555.00	1	2	2
(e)	555.34	2	1	1
(f)	567.40	2	1	2
(g)	570.55	0	2	2
(h)	573.72	1	0	3
(i)	604.00	2	2	2
(j)	610.67	2	2	1

V.   CONCLUSION

In this work, an optimization strategy to provide multiple solutions with same or similar quality is proposed for MRTA problems. First, in order to decrease time of fitness evaluation and accelerate computation, sNIOA is designed. The new population is prejudged by the regression model such that non-ideal antibodies can be eliminated before entering their evaluation process and unnecessary computations are thus avoided. Then, the GM operator inspired by the base pair is proposed. Genes of an antibody are divided into several gene groups. When a gene mutates, others in the same gene group may mutate with a higher probability. The GM operator is helpful to find preferable solutions when the number of iteration is limited and the scale of a MRTA problem is large-sized. To verify the effectiveness and efficiency of the proposed algorithms, this work conducts a series of experiments on test functions and compares them with several heuristic algorithms. The results demonstrate the high speed of sNIOA and search ability of GM. Furthermore, application cases show that the proposed sNIOA-GM can well handle MRTA problems by providing multiple solutions.

Different from other task allocation methods, the major advantages of sNIOA-GM are: 1) it can provide multiple solutions with same or similar quality for decision makers to select and switch plans whenever such needs arise; 2) because of the pre-judgment by a regression model, sNIOA is able to reduce the evaluation time and computational cost; 3) the GM operator reveals a potential connection between genes and strengthens the optimization ability of sNIOA.

The further studies may include a selection mechanism and switching strategy given multiple candidate solutions. Besides, considering the balance between multimodal optimization ability and uniformity of solutions, the proposed sNIOA-GM still has some room for further improvement. Furthermore, preserved individuals should be judged more accurately to obtain different mutation probabilities according to their fitness values.

REFERENCES

[1] P. Rakshit et al., "Realization of an Adaptive Memetic Algorithm Using Differential Evolution and Q-Learning: A Case Study in Multirobot Path Planning," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 43, no. 4, pp. 814-831, July 2013.

[2] J. C. Elizondo Leal, J. G. Ramirez Torres, E. Rodriguez Tello and J. R. Martinez Angulo, "Multi-robot Exploration Using Self-Biddings under Constraints on Communication Range," in IEEE Latin America Transactions, vol. 14, no. 2, pp. 971-982, Feb. 2016.

[3] C. H. Chang, S. C. Wang and C. C. Wang, "Exploiting Moving Objects: Multi-Robot Simultaneous Localization and Tracking," in IEEE Transactions on Automation Science and Engineering, vol. 13, no. 2, pp. 810-827, April 2016.

[4] D. H. Lee, S. A. Zaheer and J. H. Kim, "Ad Hoc Network-Based Task Allocation With Resource-Aware Cost Generation for Multirobot

Systems," in IEEE Transactions on Industrial Electronics, vol. 61, no. 12, pp. 6871-6881, Dec. 2014.

[5] A. Garcia Barrientos, J. Lara Lopez, E. S. Espinoza, J. Hoyo and G. Valencia Palomo, "Object Transportation Using a Cooperative Mobile Multi-Robot System," in IEEE Latin America Transactions, vol. 14, no. 3, pp. 1184-1191, March 2016.

[6] M. Turduev, G. Cabrita, M. Kırtay, et al, " Experimental studies on chemical concentration map building by a multi-robot system using bio-inspired algorithms," in Autonomous Agents and Multi-Agent Systems, vol. 28, no. 1, pp. 72-100, 2014.

[7] H. Li, F. Karray, O. Basir and I. Song, "A Framework for Coordinated Control of Multiagent Systems and Its Applications," in IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 38, no. 3, pp. 534-548, May 2008.

[8] D. H. Lee, S. A. Zaheer and J. H. Kim, "A Resource-Oriented, Decentralized Auction Algorithm for Multirobot Task Allocation," in IEEE Transactions on Automation Science and Engineering, vol. 12, no. 4, pp. 1469-1481, Oct. 2015.

[9] D. Zhu, H. Huang and S. X. Yang, "Dynamic Task Assignment and Path Planning of Multi-AUV System Based on an Improved Self-Organizing Map and Velocity Synthesis Method in Three-Dimensional Underwater Workspace," in IEEE Transactions on Cybernetics, vol. 43, no. 2, pp. 504-514, April 2013.

[10] L. Liu L and D. A. Shell, "Large-scale multi-robot task allocation via dynamic partitioning and distribution," in Autonomous Robots, vol. 33, no. 3, pp. 291-307, 2012.

[11] L. Luo, N. Chakraborty and K. Sycara, "Provably-Good Distributed Algorithm for Constrained Multi-Robot Task Assignment for Grouped Tasks," in IEEE Transactions on Robotics, vol. 31, no. 1, pp. 19-30, Feb. 2015.

[12] Y. Cai, Z. Tang, Y. Ding, and B. Qian, "Theory and Application of Multi-robot Service-oriented Architecture," in IEEE/CAA Journal of Automatica Sinica, 3(1), 15-25, 2016.

[13] D. F. Wu, G. P. Zeng, L. G. Meng, W. J. Zhou, and L. M. Li, "Gini coefficient-based task allocation for multi-robot systems with limited energy resources," in IEEE/CAA Journal of Automatica Sinica, vol. 5, no. 1, pp. 155-168, Jan.2018.

[14] C. Nam and D. A. Shell, "Assignment Algorithms for Modeling Resource Contention in Multirobot Task Allocation," in IEEE Transactions on Automation Science and Engineering, vol. 12, no. 3, pp. 889-900, July 2015.

[15] C. Nam and D. A. Shell, "Analyzing the Sensitivity of the Optimal Assignment in Probabilistic Multi-Robot Task Allocation," in IEEE Robotics and Automation Letters, vol. 2, no. 1, pp. 193-200, Jan. 2017.

[16] C. Wang, L. Cao and C. H. Chi, "Formalization and Verification of Group Behavior Interactions," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 45, no. 8, pp. 1109-1124, Aug. 2015.

[17] J. Yao, N. Kharma and P. Grogono, "Bi-Objective Multipopulation Genetic Algorithm for Multimodal Function Optimization," in IEEE Transactions on Evolutionary Computation, vol. 14, no. 1, pp. 80-102, Feb. 2010.

[18] B. Y. Qu, P. N. Suganthan and S. Das, "A Distance-Based Locally Informed Particle Swarm Model for Multimodal Optimization," in IEEE Transactions on Evolutionary Computation, vol. 17, no. 3, pp. 387-402, June 2013.

[19] Q. Yang et al., "Adaptive Multimodal Continuous Ant Colony Optimization," in IEEE Transactions on Evolutionary Computation, vol. 21, no. 2, pp. 191-205, April 2017.

[20] A. Basak, S. Das and K. C. Tan, "Multimodal Optimization Using a Biobjective Differential Evolution Algorithm Enhanced With Mean Distance-Based Selection," in IEEE Transactions on Evolutionary Computation, vol. 17, no. 5, pp. 666-685, Oct. 2013.

[21] C. H. Yoo, D. K. Lim and H. K. Jung, "A Novel Multimodal Optimization Algorithm for the Design of Electromagnetic Machines," in IEEE Transactions on Magnetics, vol. 52, no. 3, pp. 1-4, March 2016.

[22] Q. Yang, W. N. Chen, Y. Li, C. L. P. Chen, X. M. Xu and J. Zhang, "Multimodal Estimation of Distribution Algorithms," in IEEE Transactions on Cybernetics, vol. 47, no. 3, pp. 636-650, March 2017.

[23] X. Li, M. G. Epitropakis, K. Deb and A. Engelbrecht, "Seeking Multiple Solutions: An Updated Survey on Niching Methods and Their Applications," in IEEE Transactions on Evolutionary Computation, vol. 21, no. 4, pp. 518-538, Aug. 2017.

[24] J. E. Vitela and O. Castanos, "A real-coded niching memetic algorithm for continuous multimodal function optimization," 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), Hong Kong, pp. 2170-2177, 2008.

[25] C. H. Chen, T. K. Liu and J. H. Chou, "A Novel Crowding Genetic Algorithm and Its Applications to Manufacturing Robots," in IEEE Transactions on Industrial Informatics, vol. 10, no. 3, pp. 1705-1716, Aug. 2014.

[26] X. Li, "Niching Without Niching Parameters: Particle Swarm Optimization Using a Ring Topology," in IEEE Transactions on Evolutionary Computation, vol. 14, no. 1, pp. 150-169, Feb. 2010.

[27] Y. H. Zhang, Y. J. Gong, H. X. Zhang, T. L. Gu and J. Zhang, "Toward Fast Niching Evolutionary Algorithms: A Locality Sensitive Hashing-Based Approach," in IEEE Transactions on Evolutionary Computation, vol. 21, no. 3, pp. 347-362, June 2017.

[28] S. Biswas, S. Kundu and S. Das, "Inducing Niching Behavior in Differential Evolution Through Local Information Sharing," in IEEE Transactions on Evolutionary Computation, vol. 19, no. 2, pp. 246-263, April 2015.

[29] S. Biswas, S. Kundu and S. Das, "An Improved Parent-Centric Mutation With Normalized Neighborhoods for Inducing Niching Behavior in Differential Evolution," in IEEE Transactions on Cybernetics, vol. 44, no. 10, pp. 1726-1737, Oct. 2014.

[30] W. Dong and M. Zhou, "Gaussian Classifier-Based Evolutionary Strategy for Multimodal Optimization," in IEEE Transactions on Neural Networks and Learning Systems, vol. 25, no. 6, pp. 1200-1216, June 2014.

[31] W. Gao, G. G. Yen and S. Liu, "A Cluster-Based Differential Evolution With Self-Adaptive Strategy for Multimodal Optimization," in IEEE Transactions on Cybernetics, vol. 44, no. 8, pp. 1314-1327, Aug. 2014.

[32] Y. Peng Y and B. L. Lu, "Hybrid learning clonal selection algorithm," in Information Sciences, vol. 296, no. 19, pp. 128-146, 2015.

[33] Z. Hu, "A container multimodal transportation scheduling approach based on immune affinity model for emergency relief," in Expert Systems with Applications, vol. 38, no.3, pp. 2632-2639, 2011.

[34] K. M. Woldemariam and G. G. Yen, "Vaccine-Enhanced Artificial Immune System for Multimodal Function Optimization," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 40, no. 1, pp. 218-228, Feb. 2010.

[35] Q. Xu, L. Wang and J. Si, "Predication based immune network for multimodal function optimization," in Engineering Applications of Artificial Intelligence, vol. 23, no.4, pp. 495-504, 2010.

[36] Y. Lei, F. Jia, J. Lin, S. Xing and S. X. Ding, "An Intelligent Fault Diagnosis Method Using Unsupervised Feature Learning Towards Mechanical Big Data," in IEEE Transactions on Industrial Electronics, vol. 63, no. 5, pp. 3137-3147, May 2016.

[37] S. A. Hofmeyr and S. Forrest, "Architecture for an artificial immune system," in Evolutionary Computation, vol. 8, no. 4, pp. 443-473, 2000.

[38] J. S. Chun, H. K. Jung and S. Y. Hahn, "A study on comparison of optimization performances between immune algorithm and other heuristic algorithms," in IEEE Transactions on Magnetics, vol. 34, no. 5, pp. 2972-2975, Sep 1998.

[39] D. N. Cooper, E. V. and M. Krawczak M, "The human gene mutation database," in Nucleic Acids Research, vol. 26, no. 1, pp. 285-287, 1998.

[40] R. Brits, A. Engelbrecht, and F. van den Bergh, "A niching particle swarm optimizer," In Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning 2002, pp. 692–696, 2002.

[41] B. Sareni, L. Krahenbuhl and A. Nicolas, "Niching genetic algorithms for optimization in electromagnetics. I. Fundamentals," in IEEE Transactions on Magnetics, vol. 34, no. 5, pp. 2984-2987, Sep 1998.

[42] N. Xu, Y. Ding, L. Ren and K. Hao, "Degeneration Recognizing Clonal Selection Algorithm for Multimodal Optimization," in IEEE Transactions on Cybernetics, vol. 48, no. 3, pp. 848-861, March 2018.

[43] Ran Cheng, Miqing Li and X. Yao, "Parallel peaks: A visualization method for benchmark studies of multimodal optimization," 2017 IEEE Congress on Evolutionary Computation (CEC), San Sebastian, 2017, pp. 263-270.

[44] G. P. Singh and A. Singh, "Comparative Study of Krill Herd, Firefly and Cuckoo Search Algorithms for Unimodal and Multimodal Optimization," in International Journal of Intelligent Systems and Applications, vol. 6, no. 3, pp. 35-49, Feb. 2014.

[45] G. Ayorkor Korsah, , Anthony Stentz and M. Bernardine Dias, "A comprehensive taxonomy for multi-robot task allocation," in The International Journal of Robotics Research, vol. 32, no. 12, pp. 1495-1512, Oct. 2013.

[46] H. Yuan, J. Bi, W. Tan, M. Zhou, B. H. Li and J. Li, "TTSA: An Effective Scheduling Approach for Delay Bounded Tasks in Hybrid Clouds," in

IEEE Transactions on Cybernetics, vol. 47, no. 11, pp. 3658-3668, Nov. 2017.

Li Huang (S'17) received the B.S. degree from Donghua University, Shanghai, China, in 2013, where she is currently pursuing the Ph. D. degree in control science and engineering. She is now a visiting student with the Helen and John C. Hartmann Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA, supported by the China Scholarship Council. Her research interests include cooperative control and dynamic optimization of multi-robot systems and intelligent automation.

Dr. Yongsheng Ding (M'00-SM'05) is currently a Professor at College of Information Sciences and Technology, Donghua University, Shanghai, China. He obtained the B.S. and Ph.D. degrees in Electrical Engineering from Donghua University, Shanghai, China in 1989 and 1998, respectively. He is a Changjiang Distinguished Professor appointed by the Ministry of Education, China. From 1996 to 1998, he was a Visiting Scientist at Biomedical Engineering Center, The University of Texas Medical Branch, TX, USA. From February 2005 to April 2005, he was a Visiting Professor at Department of Electrical and Computer Engineering, Wayne State University, MI, USA. From September 2007 to February 2008, he was a Visiting Professor at Harvard Medical School, Harvard University, MA, USA. He serves as Senior Member of Institute of Electrical and Electronics Engineers (IEEE). He has published more than 300 technical papers, and ten research monograph/advanced textbooks. His scientific interests include computational intelligence, network intelligence, intelligent Internet of things, big data intelligence, and intelligent robots.

MengChu Zhou (S'88-M'90-SM'93-F'03) received his B.S. degree in Control Engineering from Nanjing University of Science and Technology, Nanjing, China in 1983, M.S. degree in Automatic Control from Beijing Institute of Technology, Beijing, China in 1986, and Ph. D. degree in Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY in 1990. He joined New Jersey Institute of Technology (NJIT), Newark, NJ in 1990, and is now a Distinguished Professor of Electrical and Computer Engineering. His research interests are in Petri nets, intelligent automation, Internet of Things, big data, web services, and intelligent

transportation. He has over 700 publications including 12 books, 400+ journal papers (300+ in IEEE transactions), 11 patents and 28 book-chapters. He is the founding Editor of IEEE Press Book Series on Systems Science and Engineering and Editor-in-Chief of IEEE/CAA Journal of Automatica Sinica. He is a recipient of Humboldt Research Award for US Senior Scientists from Alexander von Humboldt Foundation, Franklin V. Taylor Memorial Award and the Norbert Wiener Award from IEEE Systems, Man and Cybernetics Society for which he serves as VP for Conferences and Meetings. He is a life member of Chinese Association for Science and Technology-USA and served as its President in 1999. He is a Fellow of International Federation of Automatic Control (IFAC), American Association for the Advancement of Science (AAAS) and Chinese Association of Automation (CAA).

Dr. Kuangrong Hao is currently a Professor at the College of Information Sciences and Technology, Donghua University, Shanghai, China. She obtained her B.S. degree in Mechanical Engineering from Hebei University of Technology, Tianjin, China in 1984, her M.S. degree from Ecole Normale Supérieur de Cachan, Paris, France in 1991, and her Ph.D. degree in Mathematics and Computer Science from Ecole Nationale des Ponts et Chaussées, Paris, France in 1995. She has published more than 100 technical papers, and three research monographs. Her scientific interests include machine vision, image processing, robot control, intelligent control, and digitized textile technology.

Yaochu Jin (M'98-SM'02-F'16) received the B.Sc., M.Sc., and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 1988, 1991, and 1996 respectively, and the Dr.-Ing. degree from Ruhr University Bochum, Germany, in 2001. He is a Professor in Computational Intelligence, Department of Computer Science, University of Surrey, Guildford, U.K. He was also a Finland Distinguished Professor funded by the Finnish Agency for Innovation and a Changjiang Distinguished Visiting Professor appointed by the Ministry of Education, China. His main research interests include data-driven surrogate-assisted evolutionary optimization, evolutionary multi-objective optimization, evolutionary learning, interpretable and secure machine learning, and evolutionary developmental systems. Dr. Jin is the Editor-in-Chief of the IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS and Co-Editor-in-Chief of Complex & Intelligent Systems. He is the recipient of the 2018 IEEE Transactions on Evolutionary Computation Outstanding Paper Award, the 2015 and 2017 IEEE Computational Intelligence Magazine Outstanding Paper Award. He is a Fellow of IEEE.