IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 16, 2019, accepted May 14, 2019, date of publication May 20, 2019, date of current version June 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2917838

A New Evolutionary Multiobjective Model for Traveling Salesman Problem

XUEJIAO CHEN[1], YUXIN LIU[2], XIANGHUA LI[1], ZHEN WANG[3], SONGXIN WANG[4], AND CHAO GAO[1]

[1]College of Information and Computer Science, Southwest University, Chongqing 400715, China
[2]College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China
[3]Center for Optical Imagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xian 710072, China
[4]School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai 200433, China

Corresponding authors: Zhen Wang (zhenwang0@gmail.com) and Chao Gao (cgao@swu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant U1803263 and Grant 61762020, and in part by the Natural Science Foundation of Chongqing under Grant cstc2018jcyjAX0274.

ABSTRACT The traveling salesman problem (TSP) is one of the most classical NP-hard problems in the combinatorial optimization, as many practical problems, such as scheduling problems and vehicle-routing cost allocation problems can be abstracted. The introduction of multiobjective in the TSP is a very important research topic, which brings serious challenges to the TSP. Currently, genetic algorithms (GAs) are one of the most effective methods to solve the multiobjective traveling salesman problem (MOTSP). However, GA-based algorithms suffer the premature convergence, the insufficient diversity, and nonuniform distribution of solutions when solving the MOTSP, which further restrict the wide application of GA-based algorithms. In order to overcome these problems, this paper proposes an improved method for GAs based on a novel evolutionary computational model, named the *Physarum*-inspired computational model (PCM). Based on the prior knowledge of the PCM, the initialization of the population in the proposed method is first optimized to enhance the distribution of solutions. Then, the hill climbing (HC) method is used to improve the diversity of individuals and avert running into the local optimum. Compared to the other MOTSP solving algorithms, a series of experimental results demonstrate that our proposed method achieves a better performance.

INDEX TERMS Bi-objective traveling salesman problem, NSGA-II, hill climbing, *Physarum*.

I. INTRODUCTION

Multi-objective problems (MOPs) have attracted much more attentions mainly because of its wide application, such as the multiobjective network structure design [1] and multi-objective scheduling problems in the flow shop [2]. Many real-world problems can be designed as a multiobjective traveling salesman problem (MOTSP) [3], in which the presence of multiple objectives generates a series of optimal solutions (called as Pareto optimal solutions), instead of an optimal solution. Existing multiobjective optimization algorithms aim to find as many pareto optimal solutions as possible [4].

At present, most multiobjective optimization algorithms are based on the evolutionary algorithm, among which the most representative algorithms are the genetic algorithm (GA) and ant colony algorithm (ACO). Luo et al. have put forward an improved ant colony optimization algorithm for solving traveling salesman problem TSP [5]. Ye et al. have proposed an improved ACO algorithm with a strengthened negative-feedback mechanism for tackling CSPs [6]. However, there are still some limitations for ACO in practical application. For example, it is easily trapped into the local optimum and will take long executing time for finding an optimal solution, which limit the use of popularization and application of such algorithm [7], [8]. Meanwhile, different pheromone update rules will directly affect the efficiency and performance of the ants colony algorithm model. How to define an effective pheromone update rules is still an open question. Genetic algorithm has inherent implicit parallelism and powerful global optimization ability, which can provide a simple and effective method to solve complex problems. In particular, it requires few definitions of basic concepts and rules. This is the reason why GA-based

The associate editor coordinating the review of this manuscript and approving it for publication was Bora Onat.

algorithms have been widely used to solve various problems (e.g., overlapping community detection [9], multiobjective optimization [10], clustering problem [11], nash equilibria in electricity markets [12]). Therefore, this paper presents a new nature-inspired computational model based on the genetic algorithm with a faster convergence rate, better diversity, and more efficiency to solve the multiobjective traveling salesman problem.

Among various GA-based algorithms, one of the most well-known methods is the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [13], [14], which incorporates an elitism preservation strategy in evolutionary algorithms. The main idea of NSGA-II is to reproduce a population by a genetic operator and then sort them based on the non-domination rank and crowding distance [14]. Some studies have shown that NSGA-II can maintain a better spread of solutions and converge better in the obtained nondominated front in different test problems [15], [16]. NSGA-II for solving the bi-objective traveling salesman problem (BTSP), however, often cannot achieve a good trade-off solution or suffers the premature convergence, which results in falling into the local optimal solutions due to the disturbance of non-global optimal paths. Taking NSGA-II as a benchmark algorithm, a new nature-inspired computational model is applied in this paper to optimize this algorithm in order to help the original algorithm overcome the above problems and verify the validity of the nature-inspired algorithm framework.

The main part of nature-inspired computational model is based on the *Physarum*-inspired model. Currently, a series of biological experiments have demonstrated that a unicellular and multi-headed slime mold, *Physarum* shows an ability to solve mazes and construct efficient and robust networks [17]. In order to explore the inner mechanism of intelligent behaviors of *Physarum*, different scholars have established the bionic models from different perspectives. Tero et al. have captured and formulated the positive feedback mechanism of *Physarum* in foraging and have built a *Physarum*-inspired model which has been used for optimizing heuristic algorithms [17]. Gao et al. further have described the characteristics of *Physarum* from the bionic mechanism model and intelligent computation [18], [19]. Based on the positive feedback mechanism, the *Physarum*-inspired computational model (PCM) can generate the raw material pipes to link the maze with the shortest path. If the prior knowledge exists or can be generated at a low computational cost, the appropriate initial estimation can generate better solutions with a faster convergence. Based on such prior knowledge, the *Physarum*-inspired computational model is proposed to optimize the initialization of population of NSGA-II for solving multiobjective traveling salesman problems. Moreover, the hill climbing method (HC) is used to increase the diversity of individuals and avoid falling into the local optimum. What's more, compared to [20], this paper provides more detailed formulations, schematics and comparison for highlighting the performance of our proposed method. Besides, a series of detailed experiments, including carrying out parameter analyses and statistical analyses, which are implemented and compared with other algorithms in several different datasets, have shown our proposed computational model can achieve a better performance compared with compared algorithms.

The remaining of this paper is organized as follows: Sec. II illustrates some basic definitions and concepts, such as multiobjective optimization problems (MOOP), Pareto optimal solutions, BTSP and PCM. Sec. III presents the formulations of GA-based BTSP methods, and then proposes the *Physarum*-inspired NSGA-II algorithm based on the presented nature-inspired computational model. Some experiments and discussion are performed to show the effectiveness of algorithms and models in Sec. IV. Finally, some basic concluding remarks are discussed in Sec. V.

II. RELATED WORK

This section first introduces basic concepts of multiobjective optimization problems (MOOP) and Pareto optimal solutions, and then provides the problem formulation of BTSP. All of these concepts are given with respect to a minimization problem that is because each maximization problem can be transformed into a minimization problem.

A. BASIC CONCEPTS OF MOOP

A multiobjective optimization problem (MOOP) is designed to handle two or more objective functions simultaneously. As usual, a MOOP which satisfies p inequality constraints and q equality constraints can be formulated as Eq. (1).

$$MOOP = \begin{cases} min\ F(X) = (f_1(x), \ldots f_M(x)); & for\ M = 2, \ldots m \\ subject\ to\ G_i(X) > 0; & for\ i = 1, 2, \ldots p \\ H_j(X) = 0; & for\ j = 1, 2, \ldots q \end{cases} \quad (1)$$

Since different objectives in MOOP are usually in conflict with each other, it is very difficult to compare with solutions obtained by different objective functions. The goal of MOOP is to get these non-dominated solutions with good trade-offs among different objectives which are named as the Pareto set [21]. The related definitions are as follows.

Definition 1: $X_1 = (x_1^1, x_1^2, \ldots, x_1^n)^T$ is said to be dominated by $X_2 = (x_2^1, x_2^2, \ldots, x_2^n)^T$, denoted as $X_1 \prec X_2$, if and only if both conditions mentioned below are satisfied:

$$\begin{aligned} &\forall\, i \in (1, 2, \ldots, k) : f_i(X_1) \leqslant f_i(X_2) \\ &\exists\, i \in (1, 2, \ldots, k) : f_i(X_1) < f_i(X_2) \end{aligned} \quad (2)$$

Definition 2: The Pareto set (PS) is defined as the set of all Pareto optimal solutions. If a solution is not dominated by any other solutions in a feasible solution set, it is named as a Pareto optimal solution or non-dominated solution.

$$PS = \{X \in D | \nexists Y \in D, F(Y) \prec F(X)\} \quad (3)$$

Definition 3: The Pareto front (PF) is defined as an objective vector corresponding to PS in the objective space.

$$PF = \{F(X)|X \in PS\} \tag{4}$$

For MOOP instances, a true PS is always unknown [22]. Instead, the pseudo-optimal PS is defined as an approximation of true PS, which is obtained by fusing all PSs returned by all existing algorithms in several runnings [23].

B. PROBLEM FORMULATION

As an extension of a single objective traveling salesman problem (TSP), Bi-objective traveling salesman problem (BTSP) manages two objectives simultaneously, which can be described as follows. One objective is to minimize the distance, which can guarantee the shortest distance when traveling salesmen pass by N cities and return back if and only if one time. The second objective is to minimize the cost. The BTSP consists in finding a Hamiltonian cycle of N cities that optimizes the following minimization problem:

$$h_k(x) = C^k_{(m(N),m(1))} + \sum_{i=1}^{N-1} C^k_{(m(i),m(i+1))}, \quad k = 1, 2 \tag{5}$$

where $m(i)$ refers to the i^{th} city, $C^k_{(m(i),m(j))}$ represents the value factor between $m(i)$ and $m(j)$ for an objective k.

According to the definition of García-Martínez *et al.* [24], four typical measurements are used to estimate the performance of BTSP solution algorithms:

1) The graphical representation of PF is returned by an algorithm. These graphics provide a visual information for estimating the quality and distribution of solutions. It is an intuitive measurement of PF with a graphical representation. If there are two PFs, i.e., PFA and PFB, and results of PFA converge to the bottom-left region comparing with those of PFB, we can deduce that results of PFA are better than those of PFB.
2) M_1 metric represents the distance between results of an algorithm, denoted as Y, and the pseudo-optimal Pareto front ($\overline{Y}$). This metric is based on Eq. (6), in which $|Y|$ means the number of non-dominated solutions in the front of Y. The smaller M_1 metric is, the smaller difference between $\overline{Y}$ and Y is.

$$M_1(Y) = \frac{1}{|Y|} \sum_{p \in Y} min\{\| p - \overline{p} \|; \overline{p} \in \overline{Y}\} \tag{6}$$

3) M_2 metric evaluates the distribution of solutions in PF returned by an algorithm (denoted as Y). This metric is based on Eq. (7), in which σ is a positive constant. The larger M_2 metric is, the wider the coverage of obtained solutions is.

$$M_2(Y) = \frac{1}{|Y - 1|} \sum_{p \in Y} |\{q \in Y; \| p - q \| > \sigma\}| \tag{7}$$

4) M_3 metric is used to evaluate the diameter of PF returned by an algorithm (denoted as Y) based on Eq. (8), in which p_i denotes the solution value in p for an objective i. The larger M_3 metric is, the larger region of objective space of solutions locates.

$$M_3(Y) = \sqrt{\sum_{i=1}^{2} max\{\| p_i - q_i \|; p, q \in Y\}} \tag{8}$$

Many researchers have been using various methods to solve TSP with more than one objective. For examples, Florios et al. have proposed an improved version of augmented ε-constraint method to produce all Pareto optimal solutions in multiobjective traveling salesman problems [25]. Moraes et al. have presented a new approach called MOEA/NSM for bi-objective traveling salesman problems [26]. These algorithms are designed to find the Pareto-optimal set for MOTSP, and these methods are based on the elitism approach which preserve the best solution of population in each generation to the next generation. However, these techniques differ in strategies of preserving elitism and selecting subsets of solutions. Additionally, we note that some of these methods have some weaknesses of the balance between the convergence and diversity in combinatorial optimization problems with two or more objectives. In order to solve the imbalance between the diversity preservation and achieving convergence, this paper proposes a new nature-inspired computational model based on the genetic algorithm.

C. THE PHYSARUM-INSPIRED MODEL

Physarum is a large and single-celled amoeba whose vegetative state is composed of veins (i.e., pseudopodia) [18]. The flow in veins (tubes) is the carrier of chemical and physical signals, which makes up a supply network of nutrients for the whole organism. *Physarum* looks for food by expanding the network of veins, the flow of which increases or decreases with the location of food. In the biological experiment, the food source is placed in the entrance and exit of a maze, and it always generates the protoplasmic duct connecting the shortest path between the entrance and exit of the maze [27].

Based on the above biological phenomena, the basic assumption of *Physarum* model is that a graph follows the Poiseuille flow from an inlet node to an outlet node. The Poiseuille flow consists of two processes: 1) expansion process. In such process, fluxes expand across a network, as shown in Fig. 1 (a). Eqs. (9) and (10) describe the law of Poiseuille flow at each moment in the graph. 2) Shrinkage process. In this process, flows continuously concentrate on the optimal path determined by the *Physarum* model, as shown in Figs. 1 (b) and (c). The change of a linear between inlet and outlet in the graph indicates the increase or decrease of fluxes.

The *Physarum* model (PCM) can be used to find the shortest path between two points in a maze [27] or a road map [28], which is described in detail as follows. Taking Fig. 1 (a) as an example, each edge stands for a tube in a network. $D^k_{i,j}$ represents the conductivity of an edge e_{ij} at time step k, which denotes the transport capacity of flow. The greater

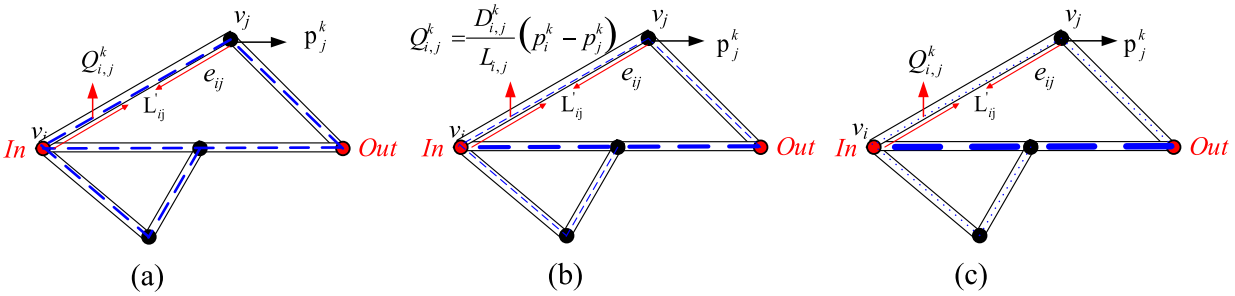


FIGURE 1. A network example for highlighting the working mechanism of PCM. The linear variation in thickness indicates the increase and decrease of the flux $Q_{i,j}^k$. The flux $Q_{i,j}^k$ is determined by the length L_{ij} of an edge e_{ij}, the conductivity $D_{i,j}^k$, and the pressure values p_i^k and p_j^k at time step k. (a) Physarum expands across a network as an initial network, (b) an intermediate network obtained by the evolutionary contraction of PCM, and (c) a final network obtained by the evolutionary contraction of PCM. The shortest path is reserved between *In* and *Out*, and other paths disappear.

the value of $D_{i,j}^k$ is, the stronger the transport capacity has. p_i^k represents the pressure value at a node v_i. Where L_{ij} is used to express the length of a tube (i, j), $Q_{i,j}^k$ stands for the flux on the edge e_{ij} at time step k. There is a positive feedback process between the flux $Q_{i,j}^k$ and conductivity $D_{i,j}^k$. Assuming that the flow in a tube is approximate to the Poiseuille flow, then the relationship among $D_{i,j}^k$, p_i^k, $Q_{i,j}^k$ and L_{ij} can be formulated as:

$$Q_{i,j}^k = \frac{D_{i,j}^k}{L_{i,j}}(p_i^k - p_j^k) \tag{9}$$

Suppose that the flow capacity of each node is 0, that is, if v_i is not an inlet node or an outlet node, then there is $\sum_i Q_{i,j}^k = 0$. According to the flow conservation law, the inlet in a network is set as I_0 and the flux input is equivalent to the flux output. Then the flux of whole network satisfies the formula Eq. (10).

$$\sum_i Q_{i,j}^k = \begin{cases} -I_0, & for\ j = In \\ I_0, & for\ j = Out \\ 0, & otherwise \end{cases} \tag{10}$$

By setting the basic pressure level p_2 as 0, all p_i^k can be calculated based on the Eq. (10). Then, we can get the flux $Q_{i,j}^k$ from Eq. (9). The value of $D_{i,j}^k$ changes with time based on the flux $Q_{i,j}^k$ of corresponding side, as shown in Eq. (11).

$$\frac{D_{i,j}^k - D_{i,j}^{k-1}}{\delta t} = \frac{\left|Q_{i,j}^k\right|}{1 + \left|Q_{i,j}^k\right|} - D_{i,j}^k \tag{11}$$

The value of $D_{i,j}^k$ is fed back to Eq. (10). If the constraint $\left|D_{i,j}^k - D_{i,j}^{k-1}\right| \leqslant 10^{-6}$ is satisfied, the iteration terminates. Fig. 1 displays the initial, intermediate and final states of a network, respectively. Obviously, the core mechanism of PCM is the positive feedback, i.e., the higher conductivity leads to the greater flux which increases the conductivity in turn [17]. The shorter the paths are, the higher the flux of tubes is. The tubes in the shorter paths tend to be wider, and will be remained in the evolutionary process of foraging network. Meanwhile, some longer tubes become narrower and disappear. Finally, the reserved tubes which are denoted as critical paths, are the solution to a path-finding problem.

In conclusion, due to the positive feedback mechanism during the foraging process of *Physarum*, the PCM model is applied to solve the shortest path problem or maze problem directly. Moreover, it is also used to solve some complex optimization problems, such as 0/1 knapsack problem [29] and network community detection [30], by combining nature-inspired algorithms. According to the type of a problem, the data set of such problem is taken as an input of the PCM model to form an initial network. Then, according to computational characteristics of the PCM model, the matrix of the flux $Q_{i,j}^k$ is obtained as an output, that is, the shortest path between specific inlet and outlet. Based on such matrix, some operations in nature-inspired algorithms can be optimized, such as the initial population in genetic algorithms or the pheromone matrix in ant colony algorithms.

III. PHYSARUM-INSPIRED NSGA-II FOR BTSP

GA-based algorithms are one type of effective methods for solving BTSP. However, GA-based algorithms suffer the premature convergence, the insufficient diversity and nonuniform distribution of solutions when solving BTSP. Based on the *Physarum*-based computational model, this paper proposes a novel modeling method for improving genetic algorithm, as shown in Fig. 2. This model consists of three parts: (1) the *Physarum*-based network computational model in Fig. 2 (a), (2) the main body of GA-based algorithms in Fig. 2 (b) and (3) the hill climbing method in Fig. 2 (c). To validate the efficiency of our proposed method, NSGA-II algorithm is selected as a benchmark algorithm and optimized based on our model, named as *p*NSGA-II.

In this section, previous GA-based methods for solving BTSP are first introduced in Sec. III-A. Sec. III-B takes the NSGA-II algorithm as an example to design an improved algorithm based on the *Physarum*-inspired computational model.

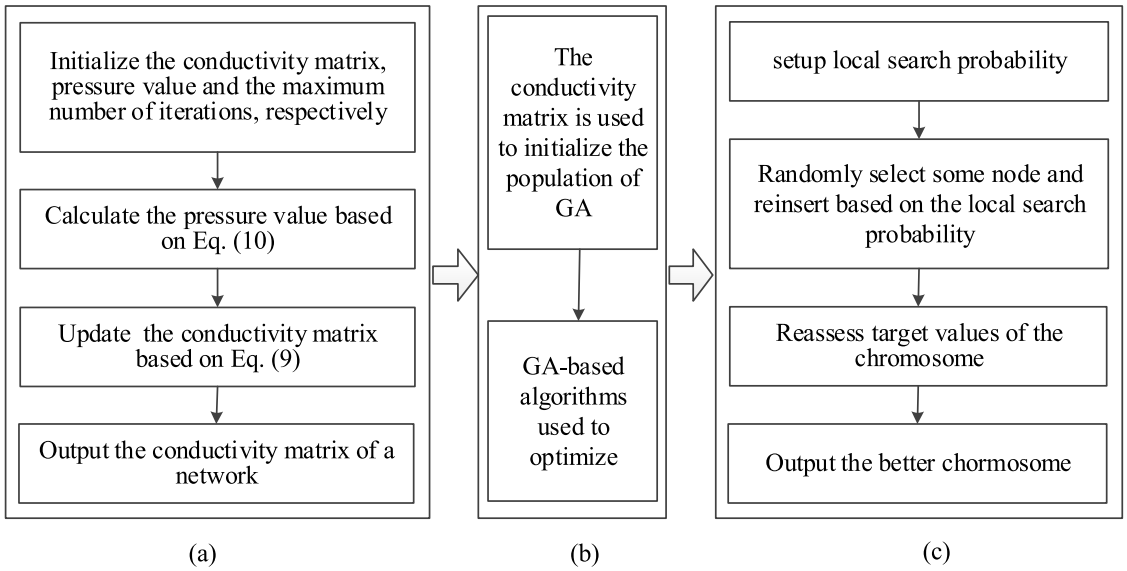


FIGURE 2. The *Physarum*-inspired computation model. This model consists of (a) *Physarum*-based network model, (b) the main body of GA-based algorithms, and (c) the hill climbing method.

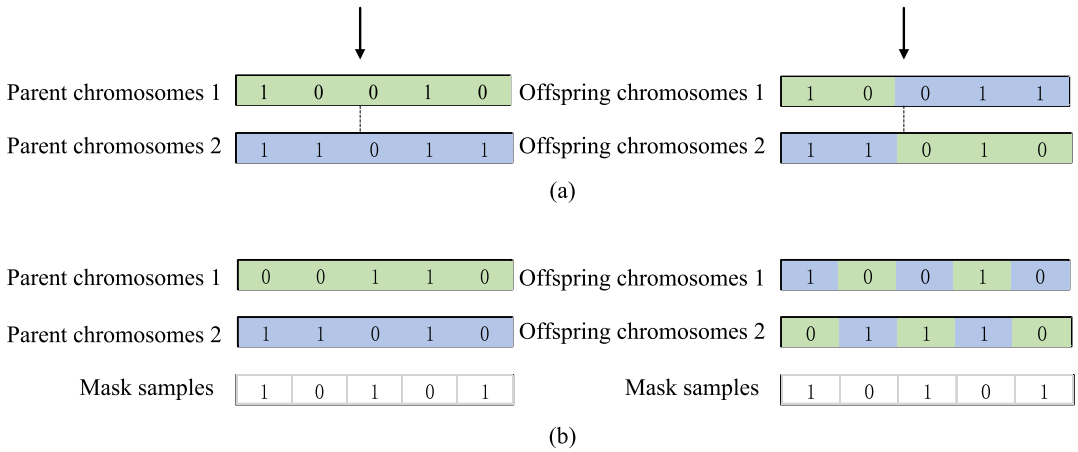


FIGURE 3. The illustration of different crossover operators in GA-based algorithms. An example of (a) the single point crossover and (b) the uniform crossover between chromosomes with 5 nodes.

A. THE FORMULATION OF GA-BASED BTSP METHOD

Genetic algorithms are optimization search algorithms that maximize or minimize given objective functions. GA-based methods are flexible methods that can be used to solve various types of problems which can be formulated as an optimization task [31].

Existing studies about GA-based methods focus on improving the encoding scheme and operator of selection, crossover and mutation. For examples, Albanyrak et al. have proposed a new mutation operator to solve TSP [32]. Elaoud et al. have proposed a dynamic selection scheme through optimizing multiple crossover and mutation operators [33]. During the optimization process of GA-based methods, these basic steps of GA-based methods are as follows:

(1) Identifying the appropriate selection technique is a critical step in genetic algorithms [34]. The selection process plays a key role in preventing the premature convergence because of a lack of diversity in the population. Therefore the selection of population in each generation is very important. A number of selection strategies have been developed and utilized for genetic algorithm optimizations.

(2) The crossover process is implemented in such a way, that children should have characteristics from parents. Usually the crossover operator consists of copying part of genotype from one parent and filling the missing genes from other parent. As show in Fig. 3, common crossover operators include single point crossover and uniform crossover. In this paper, we use the single point crossover for the following experiment.

(3) The task of mutation is to modify gene values in order to allow the exploration of search space for unexamined areas. However, the mutation should not be too

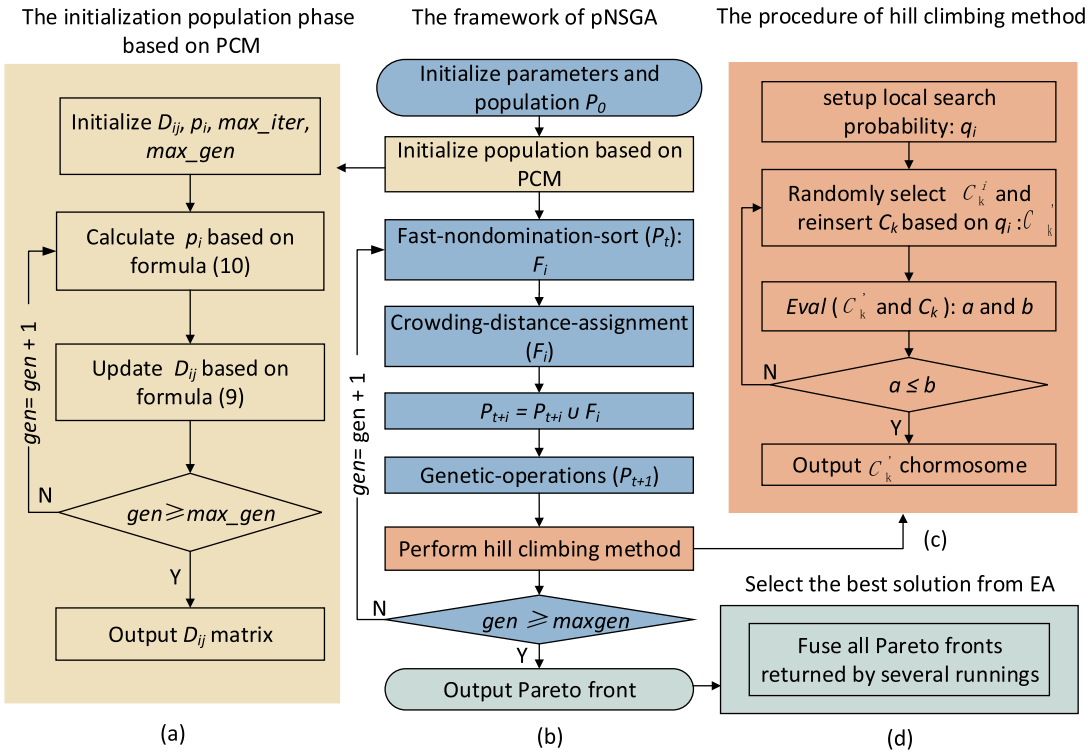


FIGURE 4. The flowchart of *p*NSGA algorithm. (a) The *physarum*-based network is mapped to the topology of a network. The conductivity matrix is calculated based on the relevant values to initialize the population of *p*NSGA-II algorithm. (b) The optimization algorithms based on the NSGA-II algorithm. (c) The hill climbing method is embed into the *p*NSGA-II algorithm framework.

destructive and invalidate the process of finding an optimal solution.

When these basic steps (i.e., population initialization, crossover or mutation) are optimized, the approach which involves both improved population initialization and local search operators is rarely considered in the existing literatures. In this paper, in order to increase the performance of GA-based methods, we take advantages of *Physarum*-inspired computational model and the hill climbing (HC) to improve the quality of population initialization and escape from the local optimum, respectively.

In all GA-based methods, NSGA-II is an effective algorithm which incorporates an elitism preservation strategy in evolutionary algorithms [14], which is widely used to solve various problems such as community detection and multiobjective optimization problems. However, it should be pointed that this algorithm tends to the premature convergence, the insufficient diversity and nonuniform distribution of solutions for solving BTSP. Therefore, taking NSGA-II as a benchmark algorithm, the *Physarum*-based network computational model (PCM) is applied to improve the performance of GA-based algorithms in order to validate the validity of proposed method.

B. FORMULATION OF pNSGA-II

Taking advantage of PCM in solving path-finding problems, we propose optimization method to improve the efficiency of NSGA-II when solving the BTSP. The proposed algorithm is denoted as *p*NSGA-II. Fig. 4 shows the flow chart of *p*NSGA-II. In *p*NSGA-II, a *Physarum* network is mapped to the topology of network. Food sources and tubes of *Physarum* network are defined as cities and paths connecting two different cities, respectively. We exploit the prior knowledge of *Physarum* network conductivity matrix to initialize the population. The optimized strategy can improve the search ability of algorithms, from two perspectives. First, the optimized method can reduce the searching space so as to speed up the convergence. Second, *Physarum* network conductivity matrices can promote the population diversity in the some way.

Compared with the original NSGA-II, we keep the main framework. The major distinction between the optimized and the original is the process of initialization population and genetic operator. During the initialization, the population is preset with the priori knowledge of PCM. Fig. 5 gives an example of initialization process of city agglomeration with 5 nodes.

In order to increase the diversity of individuals, the hill climbing method (HC) is added to the genetic operator of *p*NSGA-II. Fig. 6 presents an example of hill climbing procedure of 5 nodes. Given a chromosome $C_k = \{C_k^1, C_k^2, \ldots, C_k^n\}$, a node C_k^i is randomly selected from C_k and then we replace the location i with a random location j, where $j \neq i$. Compared with the original chromosome,

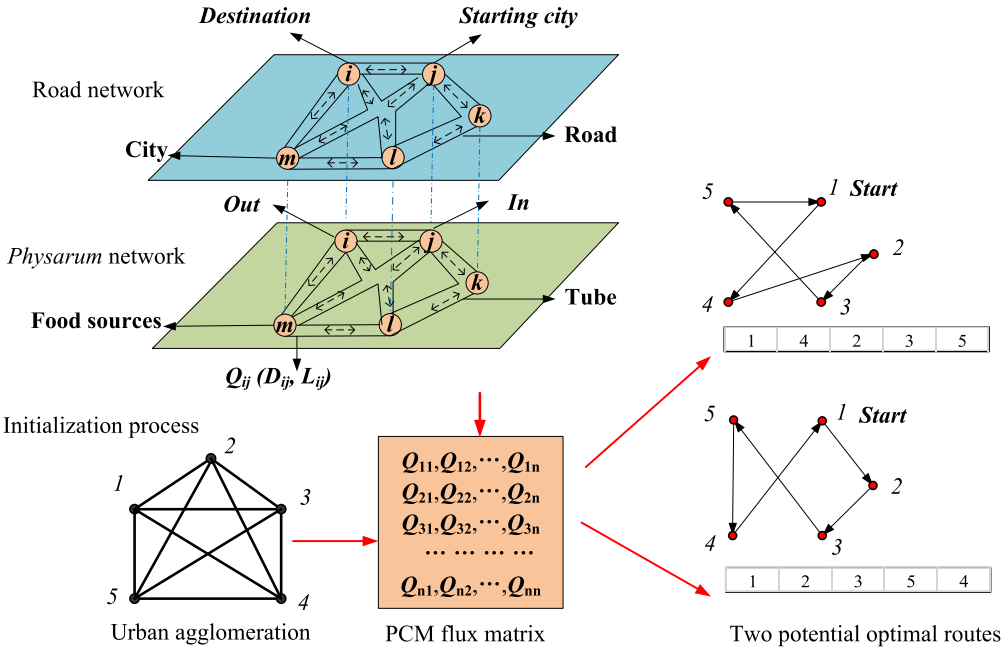


FIGURE 5. The illustration of initialization process of *p*NSGA-II. The figure shows food sources and tubes of *physarum* network which represent cities and paths or cost in a road network, respectively. Taking the city agglomeration of size 5, each objective matrix generates the corresponding flow matrix using the positive feedback mechanism of PCM, and then the initial solution is constructed according to the flow matrix of corresponding objective.

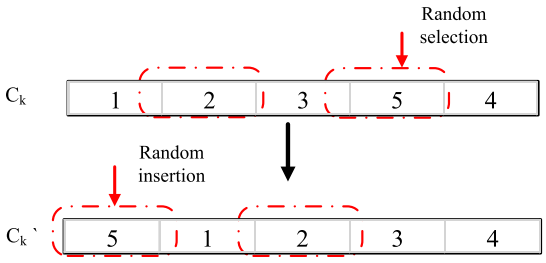


FIGURE 6. The illustration of operator of hill climbing. Our method chooses a node randomly and reinserts at a random location to generate a new chromosome.

Algorithm 1 Hill Climbing Procedure

Input: chromosome C_k, the local search probability p_i
$flag \leftarrow FALSE$
for $l = 1 : p_i * len(C_k)$ **do**
    Randomly select a node and location: $C_k^i \in C_k$ and j, where $j \neq i$
    Reinsert:$C'_k = C_k \leftarrow C_k^i$
if $Eval(C'_k) > Eval(C_k)$ **then**
    Assign:$C_k = C_k \leftarrow C'_k$
else
    Assign:$flag \leftarrow TRUE$

the new generated one is retained if it can achieve better solutions. The details of HC procedure are given in Alg. 1.

IV. EXPERIMENTS

This section first introduces bi-objective symmetric TSP instances and corresponding parameter settings in Sec. IV-A. Then experiments results and the statistical analysis are implemented in Sec. IV-B. The computation complexity analysis is carried out in Sec. IV-C. Finally, the effects of parameters are analyzed to verify the performance of *p*NSGA-II algorithm in Sec. IV-D.

A. DATASETS AND PARAMETERS

Bi-objective symmetric TSP instances are obtained from the web page.[1] Each of these instances is constructed from two different single objective TSP instances with the same number of nodes. For the sake of contrastive analysis, this paper uses eight bi-objective TSP instances, i.e., euclidAB100, kroAB100, kroAB150, kroAB200, kroDE100, kroCD100, kroAC100, and euclidEF100 to estimate our proposed method.

Some parameter settings are shown as follows. The initial value of conductivity of each tube is 1. The total number of runs affected by PCM are 30. Total steps of iteration are 500 or 1000, according to the scale of instance. The population-size, mutation, crossover and hill-climbing are set to 500/1000, 0.2, 0.6, 0.4, respectively, which are the same with the setting in NSGA-II [14]. More detailed parameter analyses are discussed in Sec. IV-D. All experiments are implemented on PC with 3.2 GHz CPU, 4 GB RAM and Windows 7 OS.

[1] https://eden.dei.uc.pt/~paquete/tsp/

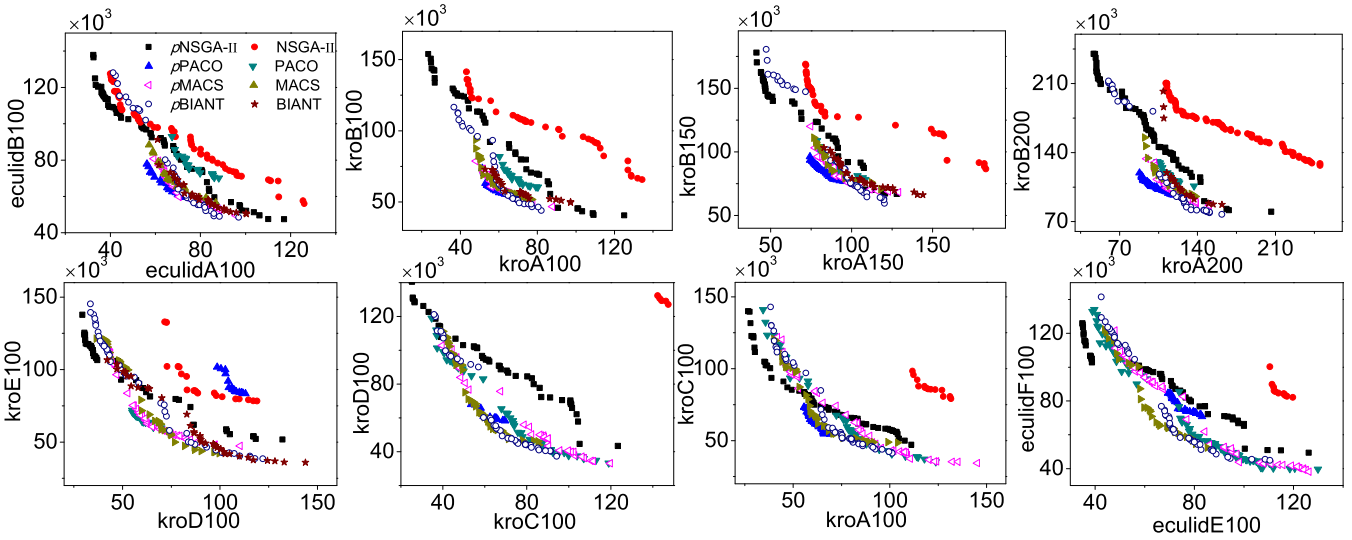


FIGURE 7. PFs returned by *p*NSGA-II, *p*PACO, *p*MACS, *p*BIANT, PACO, MACS, BIANT, and NSGA-II in eight bi-objective symmetric TSP instances. Results show that most of solutions generated by *p*NSGA-II in four instances can dominate solutions generated by NSGA-II, which means *p*NSGA-II can obtain the better PF than NSGA-II. Most of solutions generated by *p*NSGA-II converge to the upper-left region and locate a larger region of solution space than other algorithms. However, partial solutions generated by *p*NSGA-II are dominated by the most of solutions generated by other compared algorithms.

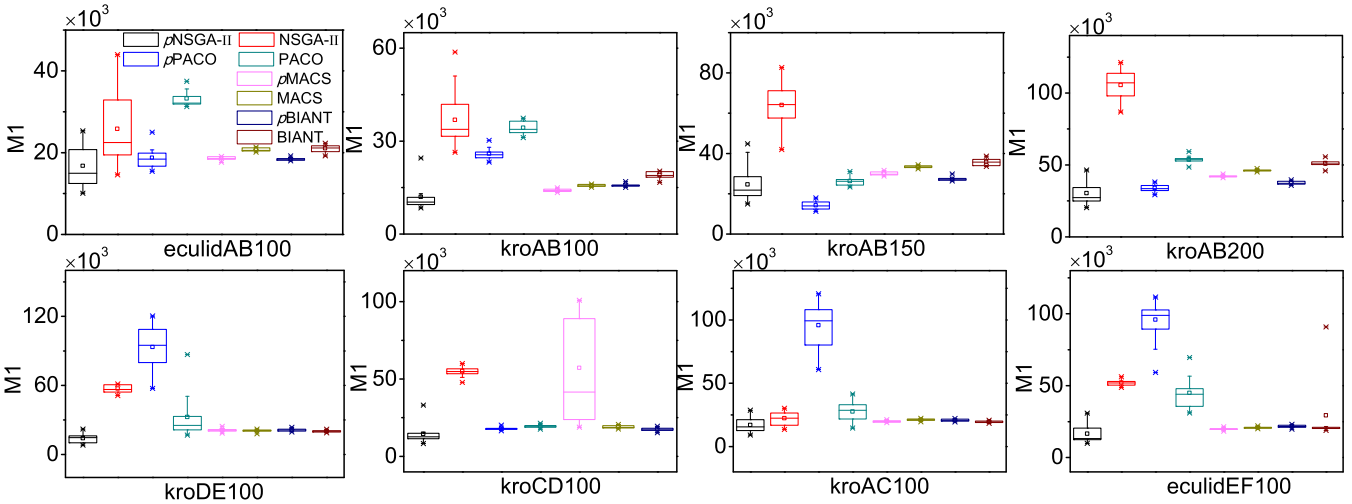


FIGURE 8. M_1 metric comparison among eight algorithms in eight instances. Results show that each corresponding M_1 value of *p*NSGA-II are much lower than those of optimized ant colony optimization (ACOs) and NSGA-II in seven instances, which means that solutions generated by *p*NSGA-II are much closer to the pseudo-optimal PFs. In other words, *p*NSGA-II algorithm works better than other algorithms about M_1 metric.

B. RESULTS AND ANALYSES

In this section, we present experimental results and carry out statistical analyses. All experiments are implemented in the same environment to enable fair comparisons between our algorithm and other algorithms including HYGA [35], NSGA-II [14], *p*PACO, *p*MACS, and *p*BIANT [36], where *p*PACO, *p*MACS, and *p*BIANT are enhanced algorithms of pareto ant colony optimization (PACO) [37], the multiple ant colony system (MACS) [38], the bicriterion ant algorithm (BIANT) [39], respectively. In order to wipe off the computational fluctuation, all results in our experiments are averaged over 30 times.

1) EXPERIMENTAL RESULTS

Figure 7 plots the graphical representation of PFs returned by eight algorithms in eight instances, where each coordinate represents an objective, and each point corresponds to a feasible solution for the instance, respectively. All PFs generated by each algorithm are fused into a single PF by removing dominated solutions. This result shows that the optimized strategy for NSGA-II can improve the quality and distribution of solutions, especially in kroAB200, kroAC100, euclidEF100 instances.

In order to further compare the performance among different algorithms, box-plots in Figs. 8-10 are used to estimate

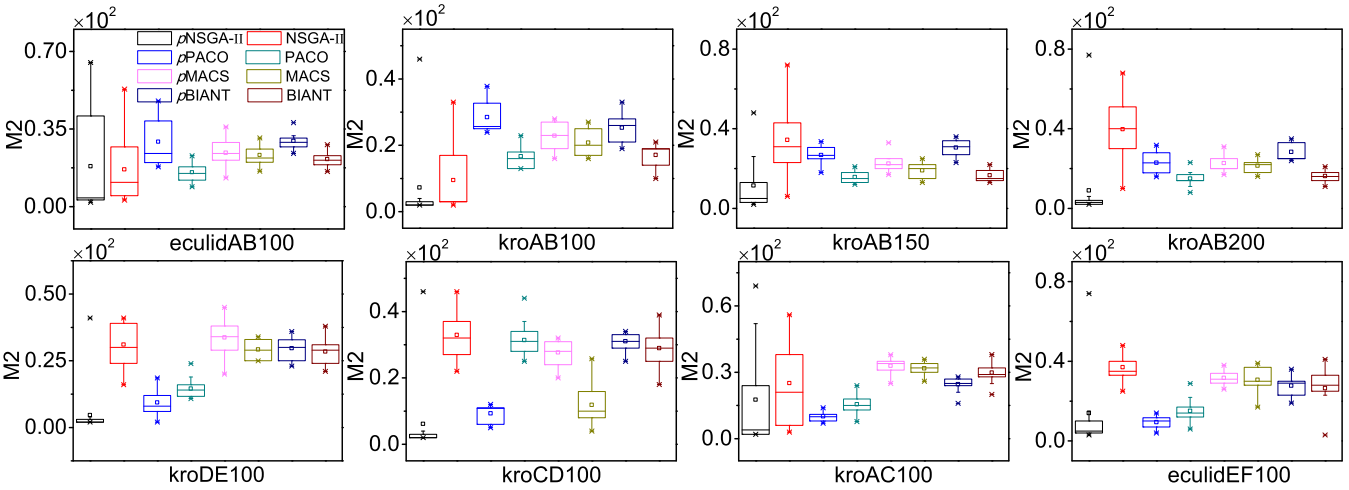


FIGURE 9. M_2 metric comparison among eight algorithms in eight instances. Results show that each corresponding M_2 values of *p*NSGA-II are slightly worse than those of optimized ACOs and NSGA-II in eight instances.

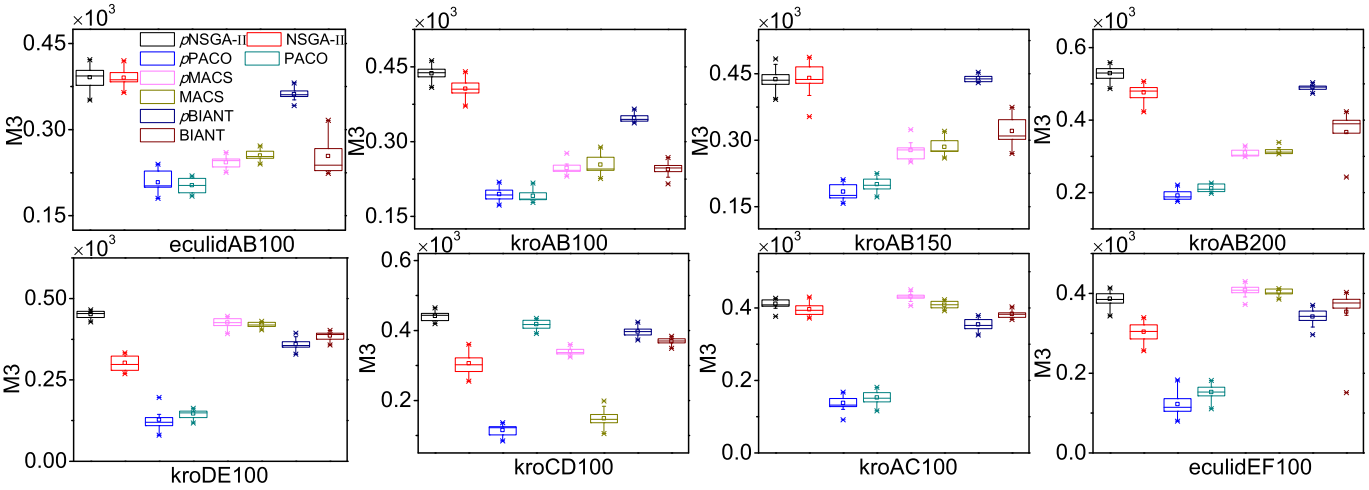


FIGURE 10. M_3 metric comparison among eight algorithms in eight instances. Results show that most of M_3 metrics of *p*NSGA-II are better than those of optimized MOACOs and NSGA-II.

the value of M_1, M_2 and M_3 metric. In each box, the highest and lowest lines represent the maximum value and minimum value with 30 runnings, respectively. The upper and lower of a box are the upper and lower quartiles, respectively. The line within a box means the median of solutions.

Figure 8 shows that PFs generated by the *p*NSGA-II are much closer to the pseudo-optimal PFs in eight instances. Fig. 9 evaluates the distribution of solutions in PFs returned by original algorithm (i.e., NSGA-II), compared algorithm (i.e., *p*PACO, *p*MACS, *p*BIANT, MACS, PACO, BIANT) and the optimized algorithm (i.e., *p*NSGA-II) according to M_2 metric. Results show that the distribution of solutions of *p*NSGA-II is less desirable while *p*BIANT and *p*MACS can obtain the better distribution of solutions in each Pareto front. Furthermore, we estimate the scalability of solutions by comparing the M_3 metric. As plotted in Fig. 10, the scalability of solutions of *p*NSGA-II is better than original and compared algorithms.

In order to further evaluate the performance of *p*NSGA-II comprehensively, this paper uses hypervolume metric proposed by Zitzler and Thiele [40] to further measure the differences of algorithms. The larger the value of hypervolume is, the better the algorithm performance is. Table 1 shows hypervolume comparison results among different algorithms in some instances. We can conclude that the performance of *p*NSGA-II is better than other algorithms.

Moreover, a typical GA-based method (i.e., HYGA [35]) is compared with our proposed method. Table 2 reports comparison results among *p*NSGA-II, NSGA-II and HYGA in four instances (i.e., euclidAB100, kroAB100, kroAC100, kroCD100). Results show that the performance of *p*NSGA-II algorithm is better than other algorithms.

TABLE 1. Comparison results of hypervolume among different algorithms in some instances.

	euclidAB100	kroAB100	kroAB150	kroAB200
*p*NSGA-II	2.51E+10	3.17E+10	3.54E+10	6.31E+10
NSGA-II	1.93E+10	2.41E+10	1.36E+09	5.96E+10
*p*PACO	5.59E+09	4.09E+09	6.26E+09	1.10E+10
PACO	5.71E+09	4.62E+09	3.23E+10	1.20E+10
*p*BIANT	1.75E+10	1.26E+10	1.42E+10	4.95E+10
BIANT	9.59E+09	6.75E+09	3.14E+10	3.65E+10
*p*MACS	7.05E+09	7.11E+09	1.17E+10	1.78E+10
MACS	7.07E+09	7.59E+09	9.09E+09	2.04E+10

TABLE 2. Hypervolume comparison results among *p*NSGA-II, NSGA-II, and HYGA in some instances.

	pNSGA-II	NSGA-II	HYGA
euclidAB100	2.51E+10	1.93E+10	1.69E+10
kroAB100	3.17E+10	2.41E+10	2.28E+09
kroAC100	2.54E+10	6.60E+09	2.23E+09
kroCD100	2.89E+10	7.57E+09	2.32E+10

2) STATISTICAL ANALYSES

In order to verify that the *p*NSGA-II algorithm is statistically superior to other algorithms, this section carries out statistical analyses according to [41]. Table 3 shows the average value of M_1 in different scales of instances. Table 4 gives the rankings of different algorithms on the various datasets based on results in Table 3.

In Table 4, symbols G_1 - G_8 denote eight instances, i.e., euclidAB100 (G_1), kroAB100, kroAB150, kroAB200, kroDE100, kroCD100, kroAC100, and euclidEF100. The values of Table 4 denote the ranking results of six algorithms from the best to the worst based on the averaged value of the valuation index M_1 in Table 3. R represents the average of all rankings on eight datasets for each algorithm. For example, the ranking R of *p*NSGA-II can be calculated as, $R = (1 + 1 + 2 + 1 + 1 + 1 + 1 + 1)/8 = 1.125$.

First, by Eq. (12), the statistics can be made in Friedman test (i.e., χ_F^2). R_j represents the rank of diverse algorithms. N and k express the number of algorithms and datasets, respectively. According to Eq. (12), the χ_F^2 of Table 4 is calculated to be 17.125. The degree of freedom of Table 4 can be obtained from the records in Chi-square table, i.e., $k-1=7$, and $\chi_{0.05}^2 = 9.488$. Because $17.125 > 9.488$, within the confidence interval of 95%, these algorithms in Table 4 show significant differences.

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{12}$$

Second, according to the significant differences among these algorithms, the Bonferroni-Dunn's test can be utilized to prove the specific distinction between two algorithms. This rule, whether the difference value between two algorithms in ranking is greater than the critical difference, is used as the evaluation criterion (denoted as *CD*). For a multiple comparison, α and q_α are the confidence level and threshold obtained by checking the Z table, respectively. Therefore, we can conclude that $q_{0.05} = 2.935$ (where $P = k(k-1)/2 = 15$) for Table 4. According to Eq. (13), we can get the critical values at the 95% confidence levels, i.e., $CD_{0.05} = 2.745$.

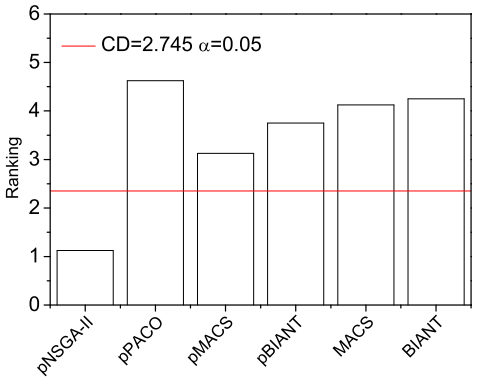


FIGURE 11. The Bonferroni-Dunn's graph corresponding to results of Table 4. The horizontal line represents the value which equals to the sum of ranking of control algorithm (i.e., *p*NSGA-II) and the corresponding CD. Those bars which exceed this line are the associated to an algorithm with worse performance than *p*NSGA-II.

$$CD_\alpha = q_\alpha\sqrt{\frac{k(k+1)}{6N}} \tag{13}$$

Since the performance of two algorithms is obviously different, and the ranking difference is larger than CD_α, it can be concluded that *p*NSGA-II is better than *p*PACO, *p*BIANT, NSGA-II, PACO, MACS, and BIANT with $\alpha = 0.05$ (95% confidence) based on Fig. 11.

$$z = \frac{R_i - R_j}{\sqrt{\frac{k(k+1)}{6N}}} \tag{14}$$

Finally, Holm's and Hochberg's methods are used to further compare the differences between two algorithms. To compare algorithm i and j, the statistic is computed (denoted as z value) by Eq. (14). Generally, the ranking result is listed in reverse order. More specifically, according to the z values, searching the normal distribution table can gain an *unadjusted p* (expressed as U_p). From $BD_{p_i} = min\{v_i; 1\}$, *Bonferroni-Dunn p* (expressed as BD_p) can be computed, where $v_i = (k-1)U_{p_i}$. From $H_{p_i} = min\{v_i; 1\}$, *Holm p* (denoted as H_p) can be computed, where $v_i = max\{(k-j)U_{p_j} : 1 \le j \le i\}$. From $HB_{p_i} = max\{(k-j)U_{p_j} : (k-1) \ge j \ge i\}$, *Hochberg p* (expressed as HB_p) can be computed.

According to the Holm's and Hochberg's procedures, Table 5 reports the statistical results of the data in Table 4. We can get the value of z by Eq. (14). Through searching and comparing the value of z and the α value in the normal distribution table, we can obtain the probabilistic error estimation of a comparison (i.e., *p-value*). *Unadjusted p* in Table 5 is *p-value* squared. However, it does not consider the remaining comparisons, when *p-value* is in multiple comparisons.

TABLE 3. The comparison of measure M_1 obtained by algorithms in different TSPLIB instances. From left to right, these TSPLIB instances are euclidAB100, kroAB100, kroAB150, kroAB200, kroDE100, kroCD100, kroAC100, and euclidEF100. These results show the superiority of *p*NSGA-II algorithm compared with other algorithms.

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8
*p*NSGA-II	16773.44	11970.53	24576.91	30356.82	14075.09789	14496.17348	16581.38	17024.68825
*p*PACO	18751.92	25936.78	14225.35	34011.15	93285.49	124638.35	95819.555	95765.945
*p*MACS	18635.8	14083.61	30077.55	42148.69	21047.97	17990.61	19858.5	19939.81
*p*BIANT	18411.75	15809.83	27301.62	37714.4	21303.37	19487.93	19858.5	20778.11
MACS	20773.23	15640.94	33406.69	46230.07	20606.13	19067.93	20773.23	21167.46
BIANT	21039.72	18997.61	35744.65	51209.66	20179.65	17529.45	29215.9	19607.585

TABLE 4. The ranking obtained based on Table 3. The value means the ranking result of these algorithms in each instance.

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	Ranking R
*p*NSGA-II	1	1	2	1	1	1	1	1	1.125
*p*PACO	4	6	1	2	6	6	6	6	4.625
*p*MACS	3	2	4	4	4	3	2	3	3.125
*p*BIANT	2	4	3	3	5	5	4	4	3.75
MACS	5	3	5	5	3	4	3	5	4.125
BIANT	6	5	6	6	2	2	5	2	4.25

TABLE 5. The *p*-value on datasets G_1 - G_8 (*p*NSGA-II is the control algorithm), which reports statistical results of Table 4. According to such comparison, it is easy to reach a conclusion that *p*NSGA-II is better than other comparison algorithms at the 95% confidence level.

*p*NSGA-II vs.	z	Unadjusted p	Bonferroni-Dunn p	Holm p	Hochberg p
*p*PACO	3.74	0.00018	0.0009	0.0009	0.0324
BIANT	3.34	0.00084	0.0042	0.00336	0.0324
MACS	3.21	0.00132	0.0066	0.00396	0.0324
*p*BIANT	2.81	0.0050	0.025	0.01	0.0324
*p*MACS	2.14	0.0324	0.162	0.0324	0.0324

Adjusted p-value (*APVs*) considers multiple tests and can be used directly as the assumptive *p-value* in the multiple algorithm comparison range. *Bonferroni-Dunn p* (BD_p), *Holm p* (H_p) and *Hochberg p* (HB_p) represent the three calculated *APVs*. According to such comparison, we can conclude that *p*NSGA-II is better than other comparison algorithms at the 95% confidence level.

C. COMPUTATION COMPLEXITY ANALYSIS

In the proposed *Physarum*-inspired algorithm framework, as shown in Fig. 4, there are three parts: (1) the *Physarum*-based network computational model, (2) the main body of GA-based algorithms and (3) the hill climbing method. The computational complexity of the above three parts are analyzed as follows.

For the *Physarum*-based model (PCM) in Fig. 4 (a), the Poiseuille flow has only a pair of inlet and outlet. The path from the inlet to other N nodes needs N operations, i.e., each iteration requires the solution of N-element equations (i.e., Eqs. (9) and (10)). According to the Gauss elimination, the time complexity of solving such equations is $O(N^3)$. Assuming that the maximum iteration number of the PCM is *max_gen*, in general, *max_gen* has little effect on the PCM and is usually set as 1. Therefore, the computational complexity of the first part is $O(N^3)$.

For the benchmark algorithm in Fig. 4 (b), the computational complexity of NSGA-II is $O(MN^2)$. Specifically, M represents the number of objective functions that is much less than N [14].

For the hill-climbing method in Fig. 4 (c), the computational complexity of generating new chromosomes based on the local search probability q_i is $O(N)$. The threshold parameter is a constant k. Therefore, the computational complexity of the hill-climbing method is $O(k * N)$, where k is much less than N.

According to above analyses, the computational complexity of the our proposed *Physarum*-inspired computation model is $O(N^3)$.

D. PARAMETER ANALYSIS

In *p*NSGA-II algorithm, there are four important parameters which are (1) total steps of iteration (*gen*),

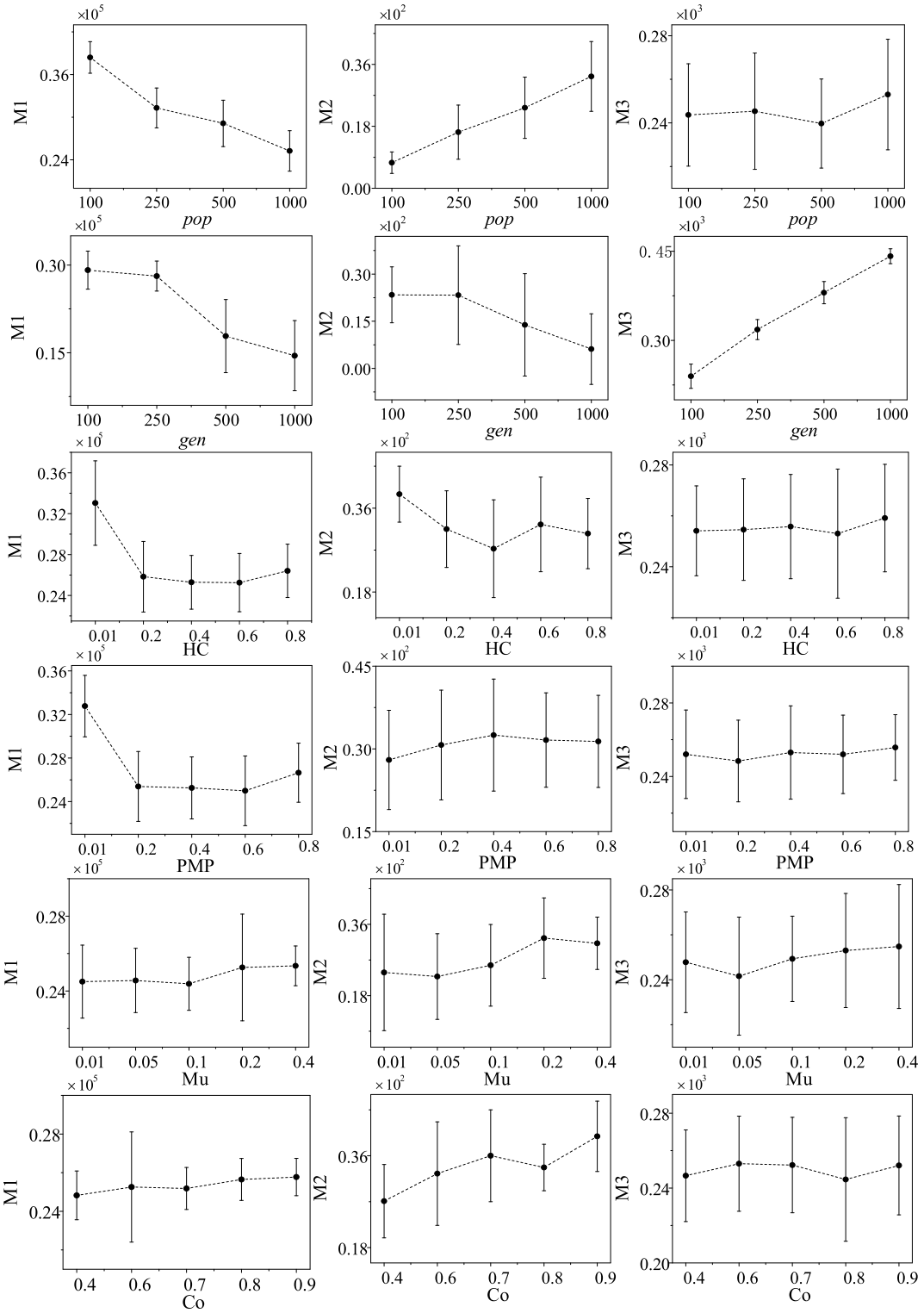


FIGURE 12. M_1, M_2, and M_3 values obtained by *p*NSGA-II with different *pop*, *gen*, *HC*, *PMP*, *Mu*, and *Co*, settings in kroCD100 instance. With the increase of parameters *pop*, *gen*, *Mu*, and *Co*, the overall performances generally increase. When *Mu* and *Co* take 0.2 and 0.7 respectively, the performance of *p*NSGA-II is best. When the parameters (i.e., *HC* and *PMP*) are within the range of 0.2 to 0.8, the performance slightly fluctuates with the increase of parameter values. Therefore, the improved algorithm *p*NSGA-II is less sensitive to the change of *HC* and *PMP* in kroCD100 instance in the whole.

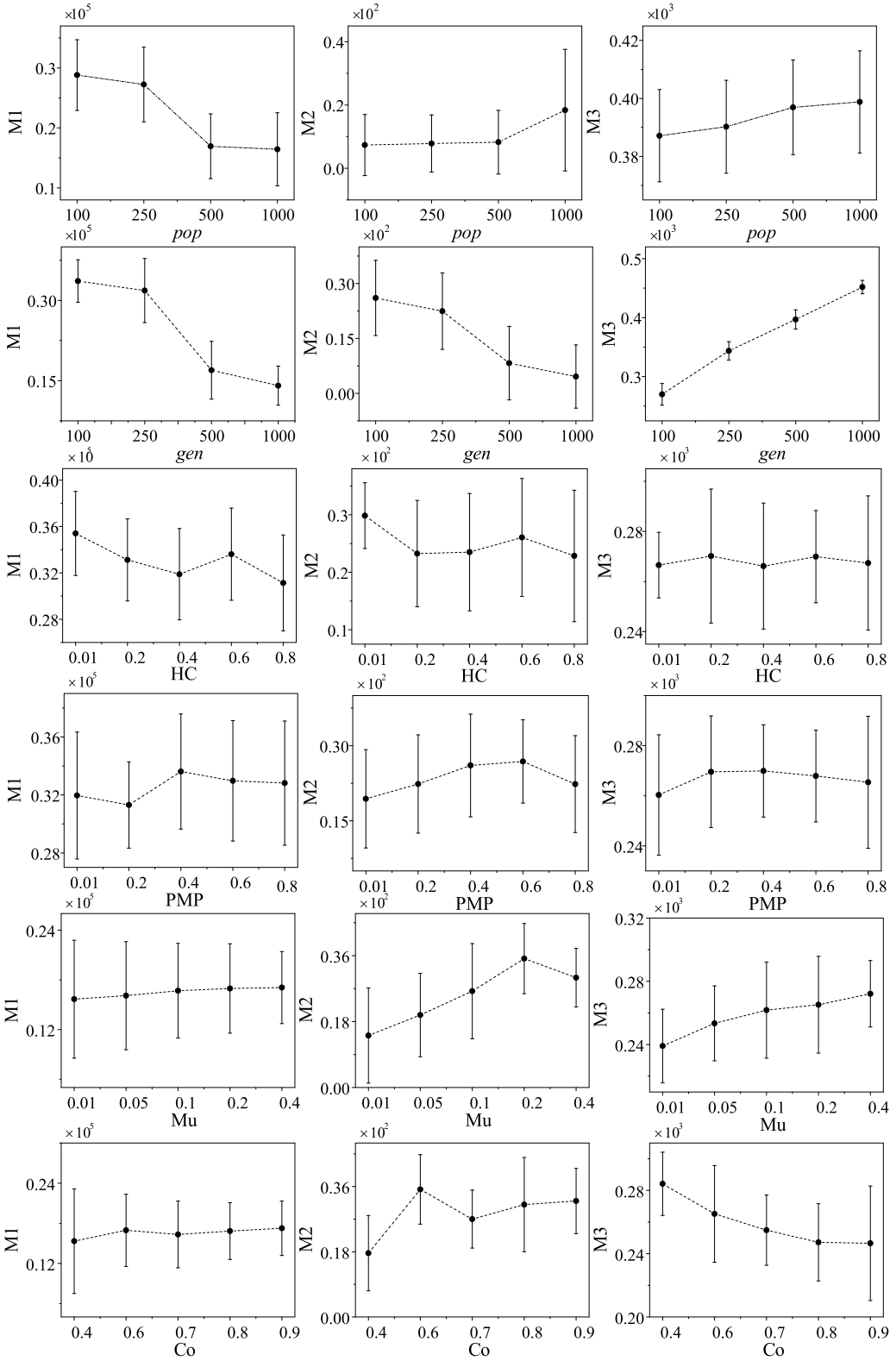


FIGURE 13. M_1, M_2, and M_3 values obtained by *p*NSGA-II with different *pop*, *gen*, *HC*, *PMP*, *Mu*, and *Co*, settings in kroDE100 instance. Similar to the results in kroCD100 instance, with the increase of parameters *pop*, *gen*, *Mu*, and *Co*, the overall performance of *p*NSGA-II generally increases. When *Mu* and *Co* take 0.2 and 0.6 respectively, the performance of *p*NSGA-II is best. When the parameters (i.e., *HC* and *PMP*) are within the range of 0.2 to 0.8, the performance slightly fluctuates with the increase of parameter values. Overall, the improved algorithm *p*NSGA-II is less sensitive to the change of *HC* and *PMP* in kroDE100 instance in the whole.

(2) population size (*pop*), (3) hill-climbing percentage (*HC*), (4) PCM percentage (*PMP*), (5) mutation probability (*Mu*) and (6) crossover probability (*Co*). The hill-climbing percentage (*HC*) represents the set local search probability, i.e., probability q_i in Fig. 4. The PCM percentage (*PMP*) represents the selection ratio of the PCM model output which is applied to the initial population. *gen*, *pop*, *Co* and *Mu* represent iteration times, population size, crossover probability and mutation probability, respectively. In this section, we mainly discuss the effect of these parameters on the algorithm. This section uses two datasets for experiments, which are kroDE100 and kroCD100. The setting range for each parameter is shown below. $gen = 100, 250, 500, 1000$. $pop = 100, 250, 500, 1000$. $HC = 0.01, 0.2, 0.4, 0.6, 0.8$. $PMP = 0.01, 0.2, 0.4, 0.6, 0.8$. $Mu = 0.01, 0.05, 0.1, 0.2, 0.4$. $Co = 0.4, 0.6, 0.7, 0.8, 0.9$. For the experiment of each parameter, the remaining parameters are the same as those set in the previous section and remain unchanged.

Figures 12 - 13 show the average value of M_1, M_2, M_3 obtained by *p*NSGA-II with different *pop*, *gen*, *HC*, *PMP*, *Mu* and *Co*, settings on the same instance, where the standard deviation is shown around the average value.

Some phenomena can be obtained from Figs. 12 - 13. First, all indicators are roughly consistent when they are used for different test instances. Second, it can be seen from parameter analyses of kroCD100 and kroDE100 that the performance of algorithm generally increases with the increase of *pop*, *gen*, *Mu* and *Co*. When *Mu* and *Co* take 0.2 and 0.6 respectively, the algorithm performance is better. In addition, according to Figs. 12 - 13, M_2 metric decreases with the number of generations. Due to the influence of selection operation, the difference between individuals gradually decrease with the increase of generations times, thus affecting the diversity of the population and the distribution of solutions. M_2 metric evaluates the distribution of solutions in Pareto front returned by an algorithm. So, the M_2 metric decreases with the number of generations.

However, the performance of *p*NSGA-II has only slightly fluctuated with the increase of *HC* and *PMP* from 0.2 to 0.8. The *p*NSGA-II algorithm is not sensitive to the change of parameter values in the whole. Therefore, in order to reduce the running time, both *HC* and *PMP* are generally set as 0.4. In conclusion, although two newly introduced parameters *HC* and *PMP* have a slight influence on the algorithm, the algorithm *p*NSGA-II is not sensitive to the value of parameters in general.

V. CONCLUSIONS

In order to solve the imbalance between diversity preservation and achieving convergence, a method using a new nature-inspired computational model based on the genetic algorithm has been presented. What's more, this method can be used in various GA-based algorithms for solving some practical optimization problems which can be translated into MOTSP and have several objectives to be optimized.

Taking advantages of prior knowledge of *Physarum*-inspired computational model and the hill climbing method, this paper proposes an optimization strategy to optimize the initialization and genetic operator of typical GA-based algorithms. NSGA-II is selected as a test algorithm, and the improved algorithm denoted as *p*NSGA-II is applied to solve bi-objective symmetric TSP (BTSP). Extensive experiments are implemented in BTSP instances. The results show that PFs obtained by *p*NSGA-II is closer to the pseudo-optimal PFs and wider extent comparing with PFs obtained by other algorithms. We can conclude that the quality of solutions generated by *p*NSGA-II is superior to others. Overall, the conclusions are as follows.

- The prior knowledge of *Physarum*-inspired computational model (PCM) is used to optimize the initialization process of GA-based algorithms. The PCM can increase the convergence speed to reach the optimal and improve the distribution of solutions, which can find the shortest route between two sources.
- The hill climbing method (HC) increases the diversity of individuals, which can enhance the exploration space and avoid falling into the local optimum.
- The *Physarum*-inspired algorithm framework based on the genetic algorithm takes into account the algorithm convergence rate, local search capabilities and diversity preservation, which makes the algorithm easier to find the global optimal solution.

This paper designs a new evolutionary multiobjective model to solve the traveling salesman problem. Because of the high computational complexity of the *Physarum*-inspired model, our future work aims to improve the efficiency of the algorithm through reducing the computational complexity. What is more, because of the multi-dimensional characteristics in multilayer networks, how to balance the heterogeneous features of community in each layer and the quality of composite community is an important issue. Therefore, we will expand our evolutionary multiobjective model to solve the problem of community detection on multilayer networks.

ACKNOWLEDGMENT

A preliminary version of this paper was presented at the 15th Pacific Rim International Conference on Artificial Intelligence, Nanjing, China, August 28–31, 2018.

REFERENCES

[1] C. Pizzuti, "Evolutionary computation for community detection in networks: A review," *IEEE Trans. Evol. Comput.*, vol. 22, no. 3, pp. 464–483, Jun. 2018.

[2] J. Xu, C.-C. Wu, Y. Yin, and W.-C. Lin, "An iterated local search for the multi-objective permutation flowshop scheduling problem with sequence-dependent setup times," *Appl. Soft. Comput.*, vol. 52, pp. 39–47, Mar. 2017.

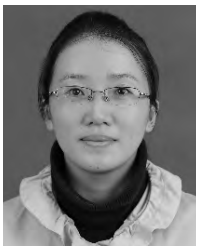
[3] S. Fereidouni, "Solving traveling salesman problem by using a fuzzy multi-objective linear programming," *Afr. J. Math. Comput. Sci. Res.*, vol. 4, no. 11, pp. 339–349, Oct. 2011.

[4] R. Z. Chen, K. Li, and X. Yao, "Dynamic multi-objectives optimization with a changing number of objectives," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 157–171, Feb. 2016.

[5] W. Luo, D. Lin, and X. Feng, "An improved ant colony optimization and its application on TSP problem," in *Proc. IEEE Int. Conf. Internet Things IEEE Green Comput. Commun. IEEE Cyber., Phys. Soc. Comput. IEEE Smart Data*, Dec. 2016, pp. 136–141.
[6] Y. Ke, Z. Changsheng, N. Jiaxu, and L. Xiaojie, "Ant-colony algorithm with a strengthened negative-feedback mechanism for constraint-satisfaction problems," *Inf. Sci.*, vols. 406–407, pp. 29–41, Sep. 2017.
[7] J. Ning, Q. Zhang, C. Zhang, and B. Zhang, "A best-path-updating information-guided ant colony optimization algorithm," *Inf. Sci.*, vols. 433–434, pp. 142–162, Apr. 2018.
[8] M. Tuba and R. Jovanovic, "Improved aco algorithm with pheromone correction strategy for the traveling salesman problem," *Int. J. Comput. Commun.*, vol. 8, no. 3, pp. 477–485, Apr. 2013.
[9] X. Wen *et al.*, "A maximal clique based multiobjective evolutionary algorithm for overlapping community detection," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 363–377, Jun. 2017.
[10] S. Jiang and S. Yang, "A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 65–82, Aug. 2017.
[11] R. Tinós, L. Zhao, F. Chicano, and D. Whitley, "NK hybrid genetic algorithm for clustering," *IEEE Trans. Evol. Comput.*, vol. 22, no. 5, pp. 748–761, Apr. 2018.
[12] F. Zaman, S. M. Elsayed, T. Ray, and R. T. Sarkerr, "Evolutionary algorithms for finding Nash equilibria in electricity markets," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 536–549, Aug. 2017.
[13] D. Cai, Y. Gao, and M. Yin, "NSGAII with local search based heavy perturbation for bi-objective weighted clique problem," *IEEE Access*, vol. 6, pp. 51253–51261, 2018. doi: 10.1109/ACCESS.2018.2869732.
[14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Jan. 2002.
[15] Y. Yuan, Y.-S. Ong, A. Gupta, and H. Xu, "Objective reduction in many-objective optimization: Evolutionary multiobjective approaches and comprehensive analysis," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 189–210, Feb. 2018.
[16] S. Wang, S. Ali, T. Yue, and M. Liaaen, "Integrating weight assignment strategies with NSGA-II for supporting user preference multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 3, pp. 378–393, Nov. 2018.
[17] A. Tero *et al.*, "Rules for biologically inspired adaptive network design," *Science*, vol. 327, no. 5964, pp. 439–442, Jan. 2010.
[18] C. Gao *et al.*, "Does being multi-headed make you better at solving problems? A survey of physarum-based models and computations," *Phys. Life Rev.*, May 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1571064518300605 and doi: 10.1016/j.plrev.2018.05.002.
[19] Y. Liu *et al.*, "Solving NP-hard problems with physarum-based ant colony system," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 14, no. 1, pp. 108–120, Jan./Feb. 2017.
[20] X. J. Chen, Z. P. Chen, Y. C. Xin, X. H. Li, and C. Gao, "Nature-inspired computational model for solving bi-objective traveling salesman problems," in *Proc. Pacific Rim Int. Conf. Artif. Int.*, vol. 11013, Jul. 2018, pp. 219–227.
[21] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, 2011.
[22] Y. T. Cao, B. J. Smucker, and T. J. Robinson, "On using the hypervolume indicator to compare Pareto fronts: Applications to multi-criteria optimal experimental design," *J. Stat. Planning Inference*, vol. 160, pp. 60–74, May 2015.
[23] M. Chica, Ó. Cordón, S. Damas, and J. Bautista, "Multiobjective constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem: ACO and random greedy search," *Inf. Sci.*, vol. 180, no. 18, pp. 3465–3487, Sep. 2010.
[24] C. García-Martínez, O. Cordón, and F. Herrera, "A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP," *Eur. J. Oper. Res.*, vol. 180, no. 1, pp. 116–148, 2007.
[25] K. Florios and G. Mavrotas, "Generation of the exact Pareto set in multi-objective traveling salesman and set covering problems," *Appl. Math. Comput.*, vol. 237, pp. 1–19, Jun. 2014.
[26] D. H. Moraes, D. S. Sanches, J. da Silva Rocha, J. M. C. Garbelini, and M. F. Castoldi, "A novel multi-objective evolutionary algorithm based on subpopulations for the bi-objective traveling salesman problem," *Soft. Comput.*, pp. 1–12, Jun. 2018. [Online]. Available: https://link.springer.com/article/10.1007/s00500-018-3269-8 and doi: 10.1007/s00500-018-3269-8.
[27] T. Nakagaki, H. Yamada, and A. Tóth, "Maze-solving by an amoeboid organism," *Nature*, vol. 407, p. 470, Oct. 2000.
[28] A. Tero, R. Kobayashi, and T. Nakagaki, "A mathematical model for adaptive transport network in path finding by true slime mold," *J. Theor. Biol.*, vol. 244, no. 4, pp. 553–564, Feb. 2007.
[29] C. Gao, S. Chen, X. Li, J. Huang, and Z. Zhang, "A *physarum*-inspired optimization algorithm for load-shedding problem," *Appl. Soft Comput.*, vol. 61, pp. 239–255, Dec. 2017.
[30] C. Gao, M. Liang, X. Li, Z. Zhang, Z. Wang, and Z. Zhou, "Network community detection based on the physarum-inspired computational framework," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 6, pp. 1916–1928, Nov. 2018.
[31] C. Han, L. Wang, Z. Zhang, J. Xie, and Z. Xing, "A multi-objective genetic algorithm based on fitting and interpolation," *IEEE Access*, vol. 6, pp. 22920–22929, 2018.
[32] M. Albayrak and N. Allahverd, "Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms," *Expert. Syst. Appl.*, vol. 38, no. 3, pp. 1313–1320, Mar. 2011.
[33] J. Wang, O. Ersoy, M. He, and F. Wang, "Multi-offspring genetic algorithm and its application to the traveling salesman problem," *Appl. Soft Comput.*, vol. 43, pp. 415–423, Jun. 2016.
[34] A. Shukla, H. M. Pandey, and D. Mehrotra, "Comparative review of selection techniques in genetic algorithm," in *Proc. Int. Conf. Futuristic Trends Comput. Anal. Knowl. Manage.*, Feb. 2015, pp. 515–519.
[35] M. Ma and H. Li, "A hybrid genetic algorithm for solving bi-objective traveling salesman problems," *J. Phys. Conf. Ser.*, vol. 887, Aug. 2017, Art. no. 012065.
[36] Z. Zhang, C. Gao, Y. Lu, Y. Liu, and M. Liang, "Multi-objective ant colony optimization based on the *physarum*-inspired mathematical model for bi-objective traveling salesman problems," *PLoS ONE*, vol. 11, no. 1, p. e0146709, Dec. 2016.
[37] K. Doerner, W. J. Gutjahr, R. F. Hartl, C. Strauss, and C. Stummer, "Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection," *Ann. Oper. Res.*, vol. 131, nos. 1–4, pp. 79–99, Oct. 2004.
[38] B. Barán and M. Schaerer, "A multiobjective ant colony system for vehicle routing problem with time windows," in *Proc. 21st IASTED Int. Conf. Appl. Inform.*, Aug. 2003, pp. 97–102.
[39] S. Iredi, D. Merkle, and M. Middendorf, "Bi-criterion optimization with multi colony ant algorithms," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.*, Jul. 2001, pp. 359–372.
[40] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.
[41] S. Garca, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization," *J. Heuristics*, vol. 15, no. 6, pp. 617–644, May 2008.

XUEJIAO CHEN is currently pursuing the M.S. degree with the College of Computer and Information Science, Southwest University, Chongqing, China. Her research interests include nature-inspired computing and complex computational problem solving.

YUXIN LIU received the bachelor's and master's degrees in software engineering and the Ph.D. degree in intelligent computing and complex system from Southwest University, Chongqing, China, in 2012, 2015, and 2018, respectively. She was a Visiting Researcher with the School of Engineering and Computer Science, Victoria University of Wellington, New Zealand, from 2016 to 2017. She is currently a Lecturer with the College of Information Engineering, Shanghai Maritime University. Her research interests include bio-inspired artificial intelligence, evolutionary computation, and combinatorial optimization problems.



XIANGHUA LI is currently an Associate Professor with the College of Computer and Information Science, Southwest University, Chongqing, China. Her research interests include nature-inspired computing and statistical analysis.

ZHEN WANG received the Ph.D. degree from Hong Kong Baptist University, Hong Kong, in 2014. He is currently a Professor with the Center for Optical Imagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University. He has authored/coauthored over 100 research papers and four review papers with over 4000 citations. His current research interests include complex networks, evolutionary game, and data science.

SONGXIN WANG received the Ph.D. degree from Jilin University, in 2002. He is currently an Associate Professor with the School of Information Management and Engineering, Shanghai University of Finance and Economics. His research interests include nature-inspired computing, complex network analysis, and knowledge representation.

CHAO GAO is currently a Professor with the College of Computer and Information Science, Southwest University, Chongqing, China. He has been a Research Fellow with the Humboldt University of Berlin, Germany, and a Postdoctoral Research Fellow with the Computer Science Department, Hong Kong Baptist University. His current research interests include complex social networks analysis, nature-inspired computing, and data-driven complex system modeling.