IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received February 10, 2018, accepted March 20, 2018, date of publication April 3, 2018, date of current version May 2, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2822824

Niching Pareto Ant Colony Optimization Algorithm for Bi-Objective Pathfinding Problem

DONG LIN, LINGCHAO HE, XINXIN FENG, AND WEI LUO
College of Physics and Information Engineering, Fuzhou University, Fuzhou 350116, China

Corresponding author: Xinxin Feng (fxx1116@fzu.edu.cn)

This work was supported in part by NSF China under Grant 61601126, Grant U1405251, and Grant 61571129, and in part by the Foundation of Fujian Province under Grant 2016J01299.

ABSTRACT In this paper, we propose a niching Pareto ant colony optimization (NPACO) algorithm to solve the bi-objective pathfinding problem. First, based on a planar navigable data model, three different searching area restricted methods are proposed and compared. In addition, a node simplification strategy is introduced to simplify nodes that exist in network branch loops, eliminating the redundant search time in the branch loops. Afterward, we propose the elitist ants and weakened strategy for an ACO to overcome the problem caused by the impact of accumulated pheromone on the suboptimal path and apply the strategy to a PACO for urban city pathfinding. Finally, the niching method is adopted to simultaneously locate and maintain multiple optimal solutions to increase search robustness. The experimental results show that the NPACO with a restricted and simplified search area returns a Pareto optimal solution set that is uniformly distributed along the Pareto frontier with low computational complexity.

INDEX TERMS Pathfinding problem, Ant colony optimization, Pareto optimal solution.

I. INTRODUCTION

The popularity of cars has brought convenience to people's lives, but also brought a series of problems, such as traffic congestion, traffic accidents and exhaust pollution. To some extent, Intelligent Transportation Systems (ITS) [1] can alleviate these problems. The path planning, one of the popular fields studied in ITS, can help drivers to choose a better path for avoiding congestion. At the same time, it can also save travel time to reduce exhaust emissions. Two different types of path planning are most commonly discussed, which are single-objective and multi-objective. For example, if the people just consider one factor when choosing roads, such as the length of the path, or the number of red lights, it would be a single-objective path planning problem. On the other hand, when more than two factors are considered, it becomes to a multi-objective problem. Actually, multi-objective is an extension of bi-objective, so we focus on bi-objective in this work.[1]

The bi-objective shortest path (BSP) problem involves finding an efficient path that satisfies several conflicting conditions. In the pathfinding problem, the objectives are often the shortest path length, minimum fare and minimum accident rate. In general, there is no single solution that simultaneously accomplishes all the objectives of a bi-objective optimization problem. Sometimes, compromises must be made regarding some of the objectives.

The BSP problem is an NP-hard problem; thus, the number of efficient solutions and the amount of computation may exponentially increase with the number of nodes [2]. The BSP can be solved by utilizing the following [3]: (1) *Top-K* query [4], which typically involves assigning a weight to each objective and combining the values of the weighted criteria into a single value; (2) the lexicographic approach [5], the basic idea of which is to assign different priorities to different objectives and then to focus on the optimal solutions according to their priority; (3) the Pareto approach [6]–[8], a multi-objective algorithm that returns a set of non-dominated solutions to the user. In a number of Pareto approaches, PACO has three main advantages [9]. First, it can handle complex project interactions and constraints better than many other metaheuristics. Second, it is robust with respect to various problem characteristics. Third, heuristic information can easily be input into the algorithm. Also, the problem that arises because PACO has a tendency to converge to a single solution, which means that all solutions easily become nearly identical, can be overcome.

[1] We also discuss the exact differences between multi-objective and bi-objective problems in Section III, Subsection D.

We propose a niching Pareto ant colony optimization, which is an extension of the traditional ant colony optimization (ACO), to solve the bi-objective path planning problem for an urban city [10]. Our algorithm can achieve the Pareto optimal solutions in a reasonable amount of time and, simultaneously, can provide users with diverse choices, which means that the solutions are uniformly distributed along the Pareto frontier. Our main contributions are as follows.

- *A feasible algorithm for the bi-objective pathfinding problem*: We propose NPACO, which is implemented with a restricted search area and simplified road networks and combines the elitist ants and weakened strategy with niching methods. It can return the Pareto optimal solutions with sufficient diversity.
- *Thorough experiments on real transportation datasets*: We perform thorough experiments on the transportation networks of Fuzhou, China. The experimental results show that our NPACO outperforms the traditional methods with respect to both speed and validity for bi-objective pathfinding.

The remainder of this paper is organized as follows. In Section II, the problem is stated, and some existing works are reviewed. Section III presents the research motivation and provides details of the proposed methodologies. Computational experiments and analysis of the algorithms are discussed in Section IV. Section V concludes the paper and proposes future works.

II. RELATED WORK

In solving the multi-objective optimization problem, Freitas [3] addressed the problem of how to evaluate the quality of a model built based on the data of a multi-objective optimization scenario. The lexicographic approach and the Pareto approach are more often used to cope with multi-objective data mining problems than the conventional weighted-formula approach. al Chami *et al.* [5] used their model with a lexicographic approach to solve a bi-objective selective pickup and delivery problem with time windows and paired demands, while Doerner *et al.* [9] explored different heuristic approaches to solve the combinatorial optimization problem by applying PACO, Pareto simulated annealing and the non-dominated sorting genetic algorithm to 18 heterogeneous random problem instances and one instance involving real-world data. With regard to the multi-objective shortest path problem, Mora *et al.* [11] proposed a study on different coarse-grained distribution schemes dealing with multi-objective ant colony based on Pareto Set. PACO was found to be the most efficient. Then, Lopez-Ibanez and Stutzle [12] proposed a formulation of algorithmic components that sufficed to describe most multi-objective ACO algorithms.

Regarding PACO applications, Baran and Schaerer [13] used their model to solve the vehicle routing problem, using two ant colonies to minimize the total length. Mora *et al.* [14] proposed a family of multi-objective ant colony optimization to solve the military pathfinding problem. Jia *et al.* [15] used the PACO algorithm to address a bi-objective scheduling problem on parallel batch processing machines with dynamic job arrivals and non-identical job sizes to minimize the makespan and total electricity cost. Hou *et al.* [16] integrated remote sensing, GIS and the Pareto ant colony algorithm (PACA) to optimize the large-scale, multi-objective allocation of water resources. Pasia *et al.* [17] improved PACO by incorporating the path relinking mechanism and illustrated how different hybrid approaches could result in shorter computational times as well as more significant improvements in medium-size instances, serving as an exploration of the benefits of PACO incorporation. The next year, they [18] solved a vehicle routing problem with route balancing, the objectives of which were to minimize the tour length and balance the routes. Iredi *et al.* [19] used heterogeneous colonies in which the ants set different weights to the two objects in a colony so that they could be able to find more solutions along the Pareto front.

TABLE 1. Mathematical symbols.

Symbol	Meaning
$\boldsymbol{x}$	specific constructed nodes of a solution
$\boldsymbol{X}$	solution set comprising all $\boldsymbol{x}$
p	pheromone evaporation rate above the arc
$\eta(i, j)$	expectation from node N_i to N_j, defined as $\frac{1}{d_{ij}}$
$\tau_{t+1}^{k}(i, j)$	kth objective pheromone above (i, j) at time $t+1$
$\Delta\tau_{t+1}^{k}(i, j)$	pheromone increment above (i, j)

Niching methods have been extensively studied to maintain the diversity of solutions of both the genetic algorithm [20], [21] and the evolutionary algorithm [6], [22]. Angus [23] applied the niching technique to an ACO algorithm to allow the simultaneous location and maintenance of multiple areas of interest in a search space.

In this work, we propose a niching Pareto ant colony optimization to solve the bi-objective path planning problem. On the one hand, the search area of an urban transportation road network is restricted by three methods. On the other hand, ACO is improved by introducing the elitist ants and weakened strategy to alleviate the impact of accumulated pheromone above the suboptimal path. The introduction of the niching method yields solution diversity. Both the network restriction and ACO improvement cause the bi-objective path planning to return a Pareto optimal solution set that is uniformly distributed along the Pareto frontier with low computational complexity.

III. NICHING PARETO ANT COLONY OPTIMIZATION ALGORITHM

In this section, we will introduce both the bi-objective shortest path problem and our solution NPACO. For the sake of convenience, we list the main mathematical symbols used in this article in TABLE 1.

A. THE BI-OBJECTIVE SHORTEST PATH PROBLEM

Consider a directed network $G(\mathcal{N}, \mathcal{A})$, where $\mathcal{N} = \{N_1, N_2, \ldots, N_n\}$ is the set of nodes and

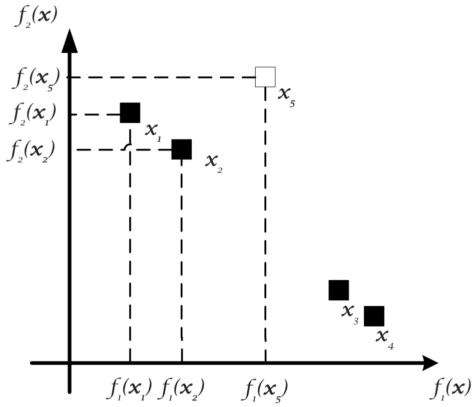


FIGURE 1. Pareto non-dominant solutions.

$\mathcal{A} = \{(i,j), (k,l), ..., (p,q)\}$ is the set of directed arcs joining the nodes in $\mathcal{N}$, and a cost vector denoted by (c^1_{ij}, c^2_{ij}) associated with each arc (i,j). In a road network, the costs c^1_{ij} and c^2_{ij} can represent the distance and accident rate for traversing arc (i,j), respectively. The objective is to find the efficient paths from a start node S to a terminal node T, which can be formulated as follows [2]:

$$\min f(\boldsymbol{x}) = \begin{cases} f_1(\boldsymbol{x}) = \sum\limits_{(i,j)\in\mathcal{A}} c^1_{ij} x_{ij}, \\ f_2(\boldsymbol{x}) = \sum\limits_{(i,j)\in\mathcal{A}} c^2_{ij} x_{ij}, \end{cases}$$

$$s.t. \sum_{(i,j)\in\mathcal{A}} x_{ij} - \sum_{(j,i)\in\mathcal{A}} x_{ji} = \begin{cases} 1, & if\ i = S \\ 0, & if\ i \neq S, T \\ -1, & if\ i = T \end{cases}$$
$$x_{ij} \in \{0,1\}, \forall (i,j) \in \mathcal{A} \quad (1)$$

where x_{ij} is a binary variable that equals to 1 if arc (i,j) exists in the efficient paths and equals to 0 otherwise. $\boldsymbol{x}$ denotes the specific constructed nodes of a solution; all solutions form a set denoted as $\boldsymbol{X}$. Users would like to choose the solution in which both $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ are minimal, but in general, one dimension is satisfied, while the other is not.

The set of solutions that contains all decision vectors for which the corresponding objective vectors cannot be improved in any dimension without degradation is considered *Pareto optimal* [6], and the solutions constitute the so-called *Pareto frontier*. This is the kind of solution that we are searching for. Mathematically, the concept of Pareto optimality is as follows:

Definition 1: Assume a minimization problem and consider two decision vectors $\mathbf{x}_1$, $\mathbf{x}_2 \in \boldsymbol{X}$. *Then,* $\mathbf{x}_1$ *is said to dominate* $\mathbf{x}_2$ *(also written as* $\mathbf{x}_1 \prec \mathbf{x}_2$*) iff* $\forall i \in \{1,2\} : f_i(\mathbf{x}_1) \leqslant f_i(\mathbf{x}_2)$ *and* $\exists j \in \{1,2\} : f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2)$.

Figure 1 presents an illustration of the Pareto non-dominant solutions. There are 5 solutions. For solutions $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, $f_2(\boldsymbol{x}_1) > f_2(\boldsymbol{x}_2)$, but $f_1(\boldsymbol{x}_1) < f_1(\boldsymbol{x}_2)$; therefore, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ do not dominate each other. For solutions $\boldsymbol{x}_2$ and $\boldsymbol{x}_5$, $f_2(\boldsymbol{x}_2) < f_2(\boldsymbol{x}_5)$, and $f_1(\boldsymbol{x}_2) < f_1(\boldsymbol{x}_5)$; therefore, $\boldsymbol{x}_2$ dominates $\boldsymbol{x}_5$.

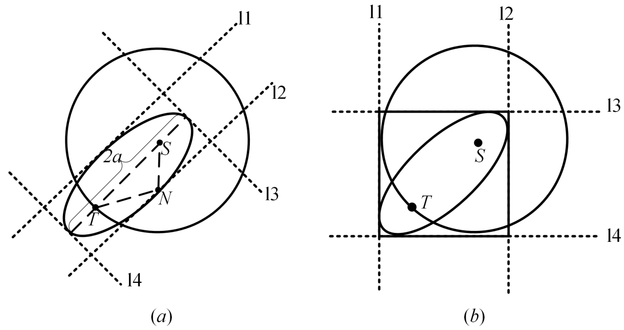


FIGURE 2. Two bounding rectangle strategies for a restricted search area.

B. RESTRICTED SEARCH AREA

Transportation networks have the apparent characteristics of a spatial distribution. Usually, there is more than one connection between two specific nodes, which is the basic difference between the sparse graphs describing the transportation networks and other planar graphs that describe the topological structure of a hierarchical structure. In the planar navigable data model, with the coordinates of S denoted as (S_x, S_y) and those of T denoted as (T_x, T_y), the shortest path nodes are approximately located within a circle, the center of which is S and the radius of which is the line segment d connecting S and T [24]. Furthermore, we define an ellipse, the foci of which are S and T and the semimajor axis of which is a. Each node N_n in the ellipse satisfies $|SN_n| + |N_nT| \leqslant 2a$. The physical meaning of this inequality is that we can ignore the case in which the path length is larger than $2a$. It is convenient to consider only the nodes within the approximate ellipse. The ellipse formula is expressed as follows:

$$\frac{[\cos\theta(x - \frac{S_x+T_x}{2}) + \sin\theta(y - \frac{S_y+T_y}{2})]^2}{a^2} + \frac{[-\sin\theta(x - \frac{S_x+T_x}{2}) + \cos\theta(y - \frac{S_y+T_y}{2})]^2}{b^2} = 1, \quad (2)$$

where $b = \sqrt{a^2 - \frac{(T_y-S_y)^2+(T_x-S_x)}{4}}$, and $\theta = \tan^{-1}(\frac{T_y-S_y}{T_x-S_x})$, $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2}]$.

The calculation of (2) to determine whether each node coordinate (x,y) is within the ellipse involves frequent use of the square algorithm and the trigonometric function, which slow down the search procedure. Applying the restrict method to the bounding rectangle can resolve this dilemma. There are two bounding rectangle strategies: minimum bounding rectangle and coordinate parallel bounding rectangle. The different restricted search area approaches are shown in figure 2.

In a minimum bounding rectangle, the 4 lines quadruple the amount of computation. Line l_1 in figure 2(a) can be expressed as follows:

$$y = \frac{T_y - S_y}{T_x - S_x} \cdot x + S_y - \frac{T_y - S_y}{T_x - S_x} \cdot S_x + \frac{b}{\cos[\tan^{-1}(\frac{T_y - S_y}{T_x - S_x})]}. \quad (3)$$

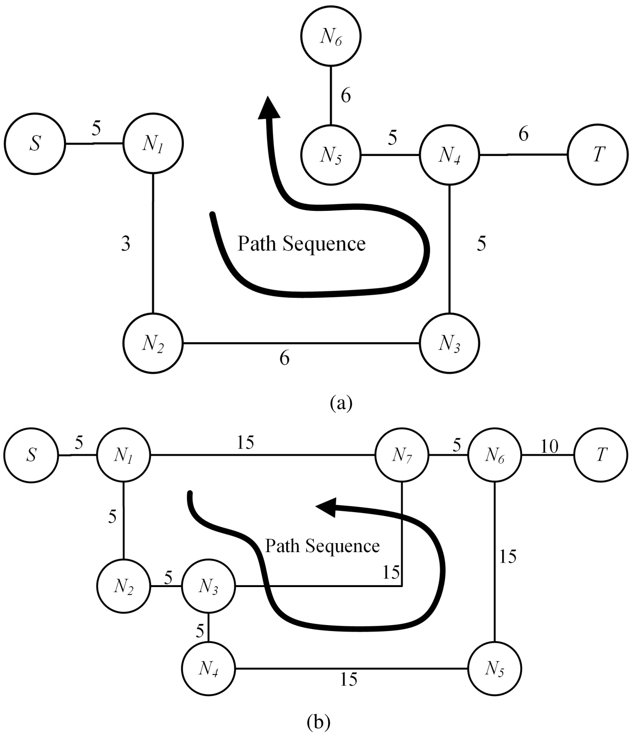


FIGURE 3. Simplified road network. (a) Scene 1: Redundant nodes in branch. (b) Scene 2: Redundant nodes in branch loop.

Line l_1 and line l_3 in figure 2(b) can be expressed as follows:

$$x = \frac{S_x + T_x}{2} - \sqrt{a^2 \cos^2 \theta + b^2 \sin^2 \theta}, \tag{4}$$

$$y = \frac{S_y + T_y}{2} + \sqrt{a^2 \sin^2 \theta + b^2 \cos^2 \theta}. \tag{5}$$

C. ROAD NETWORK SIMPLIFICATION

Although the GIS software preprocessing procedure can reduce most erroneous nodes of a road, some problems still remain, such as the existence of redundant nodes in branch loops. Road network simplification can largely reduce the pathfinding computation. It is worthwhile to take into account the risk of missing the optimal solution with the reward of saving precious time.

We show an example in figure 3, where a user plans to travel from node S to T by choosing the shortest node according to the greedy algorithm. The number on the line represents the weight. The larger the number is, the greater the weight is, which means a worse option for the user. As shown in figure 3(a), when we assume that the user stays at node N_4 and chooses N_5 as the next node, he will finally comes to node N_6, whose connected node N_5 has already been visited. Therefore, he fails to reach T in this attempt. Based on his first attempt, he should put node N_6 into the *forbiddenArray*, where the forbidden nodes are deposited. Then, node N_5 will be forbidden in the second attempt. The simplified nodes are shown in figure 4(a).

Figure 3(b) presents a more complex scene. For node N_3, the next node is N_4. For node N_6, the next node is N_7, after

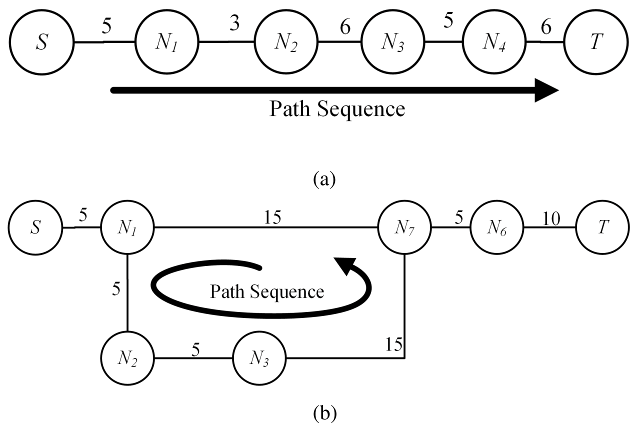


FIGURE 4. Simplified road network for complex scene. (a) Scene 1: Simplified redundant nodes in branch. (b) Scene 2: Simplified redundant nodes in branch loop.

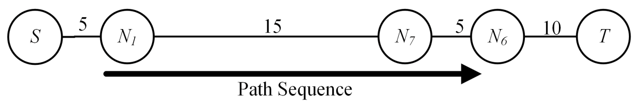


FIGURE 5. Final network of scene 2 after simplification.

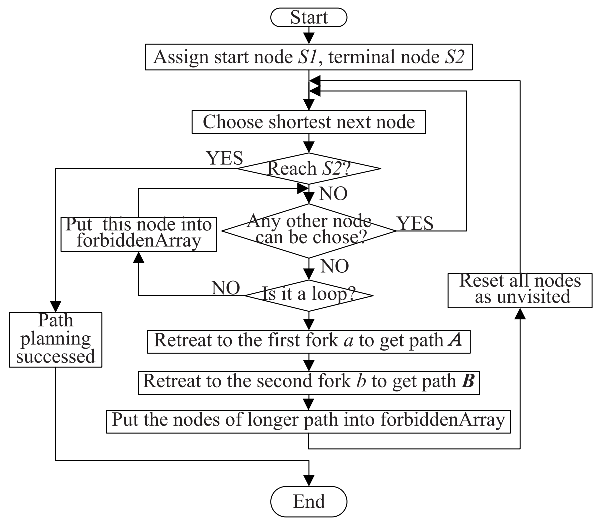


FIGURE 6. Flowchart of the road network simplification.

which the user has no way to advance further; thus, he will return to node N_1, which means that the route is a circle. In this situation, we treat the route as two paths: path $\boldsymbol{A} = \{N_3, N_4, N_5, N_6, N_7\}$ and path $\boldsymbol{B} = \{N_3, N_7\}$. We compare the lengths of the two paths and put the nodes of the longer one into the *forbiddenArray*. The simplified network is shown in figure 4(b).

In figure 4(b), there is another loop after the first simplification, which should be simplified again. The final network of figure 3(b) is shown in figure 5, and the flowchart of the road network simplification is shown in figure 6.

D. PARETO ANT COLONY OPTIMIZATION

In the early 1990s, Macro Dorigo proposed the ant system (AS) [25]; then, he presented ACO [10], a metaheuristic used to define a common framework for all versions of an AS. The goal of traditional ACO is to select the optimal path from a certain node to another node in a network. The main idea is to let the ants to choose their behavior according to the transfer probability between nodes, and each choice will strengthen the pheromone on the path. Finally, after several iterations, the traditional algorithm converges to a path that may not be globally but locally optimal.

It has some defects. First, assigning path ***A*** is identified as the current optimal path after n iterations. Then, a shorter path ***B*** is found; as a result, ***A*** is redesignated as a suboptimal path. The pheromone above is reinforced and accumulated n times, which affects the probability of choosing the next node. The pheromone above cannot be removed instantaneously, as it was evaluated in the previous search, neither can we reset it to the origin. We therefore introduce the elitist ants and weakened strategy (EAWS) to address this issue.

The EAWS defines the ant that finds the shortest path in one iteration as an elitist ant and records the shortest path length of the current iteration as L_k. Then, the EAWS reinforces the pheromone above. The times of the pheromone reinforces is recorded as n. Whenever a shorter path is found, the pheromone update rule is

$$\tau_{t+1}(i,j) = \tau_t(i,j)\cdot(1-p) + \Delta\tau_{(t+1)}(i,j), \tag{6}$$

where $\Delta\tau_{(t+1)}(i,j) = \begin{cases} \frac{Q}{L_k} & k_{th} \ ant \ passes \ arc \ (i,j) \\ 0, & otherwise \end{cases}$, and Q is the amount of pheromone ants released. The pheromone update rule above the suboptimal path is as follows:

$$\tau_{t+1}(i,j) = \tau_t(i,j)\cdot(1-p)^n. \tag{7}$$

The time complexity of the EAWS is:

$$T(n) = O(I_N \cdot M \cdot n^2), \tag{8}$$

I_N is the number of iterations, M denotes the number of ants, and n is the number of nodes.

By introducing the concepts of multiple objectives and Pareto into the EAWS-ACO, PACO was developed. The two critical problems in PACO are the following.

Critical problem 1: Guiding an ant colony exploration towards the Pareto optimal solutions;

Critical problem 2: Keeping the Pareto solutions as spread out as possible along the Pareto frontier, which is referred to as the solutions diversity problem.

For the k objectives optimization problem, the pheromone information above the arc (i,j) is stored in a vector $\tau_t^k(i,j)$, representing the current pheromone information. We address the above two problems by introducing some main functions of the suggested PACO as follows.

Decision rule. A feasible node N_j is selected to be added to the solution vector $\boldsymbol{x}$ when the user is at N_i according to a pseudo-random-proportional rule [9], which can be expressed as follows:

$$N_j = \begin{cases} \arg \max\limits_{j\in\mathcal{N}_{unvis}(i)} [\sum\limits_{k=1}^{2} R_k \cdot \tau_t^k(i,j)]^\alpha \cdot [\eta(i,j)]^\beta, & if \ q < q_0 \\ N_{next}, & otherwise \end{cases} \tag{9}$$

where q and R_k are random numbers uniformly distributed in [0, 1), q_0 is a parameter in [0, 1), which is set by the user, that represents the probability that a node is chosen. By using random factors q, Ants can not only lead within the optimal path, but also use the accumulated knowledge to find a better route.

The variable N_{next} is the next node selected according to the probability distribution given by:

$$P_{ij} = \begin{cases} \dfrac{[\sum_{k=1}^{2} R_k\cdot\tau_t^k(i,j)]^\alpha\cdot[\eta(i,j)]^\beta}{\sum_{j\in\mathcal{N}_{unvis}(i)}([\sum_{k=1}^{2} R_k\cdot\tau_t^k(i,j)]^\alpha\cdot[\eta(i,j)]^\beta)} & j\in\mathcal{N}_{unvis}(i) \\ 0 & otherwise \end{cases} \tag{10}$$

Where $\mathcal{N}_{unvis}(i)$ denotes the set of nodes that remain to be visited by the ant. And the meaning of other parameters in (10) can be referred to in TABLE 1. This probability distribution is biased by the parameters α and β, which determine the relative influence of the trails and the visibility, respectively.

Pheromone update rule. In addition to the EAWS, we perform a local pheromone update to maintain the diversity of the solutions. The local pheromone update is performed once an artificial ant has passed an edge. When an ant passes an arc (i,j), the amount of pheromone on the elements $\tau_t^k(i,j)$ of the pheromone vector is decreased for each objective k. The local pheromone update rule for these elements can be expressed as follows:

$$\tau_{t+1}^k(i,j) = (1-p)\tau_t^k(i,j) + \Delta\tau_{(t+1)}^k(i,j), \tag{11}$$

where $\Delta\tau_{(t+1)}^k(i,j) = \begin{cases} \frac{Q}{\min(f_k(x))} & ant \ passes \ arc \ (i,j) \\ 0, & otherwise \end{cases}$.

Because of the local update, ants prefer those combinations of orders that have not yet been chosen. When we update the pheromone above the specific solution $\boldsymbol{x}$, we update all the constructed arcs using (11). As a result, the diversity of the solution provided is enhanced.

Intuitively, the bi-objective problem can be changed into a multi-objective problem by transform $\sum_{k=1}^{2} R_k \cdot \tau_t^k(i,j)$ in (9) and (10) into $\sum_{k=1}^{N} R_k\cdot\tau_t^k(i,j)$. Accordingly, the multi-objective probability distribution should be considered, and we also need to update N objective pheromone in (11).

In the PACO algorithm, due to the accumulation of pheromones, most of the ants will tend to concentrate on the path having the most amount of pheromones, which may easily cause the problem of local convergence. For example, in figure 7, the solutions x_1 and x_2 is close to each other. If the axes represent path length and fare respectively, x_1 and x_2 represent almost the same choice to users, which consists of similar roads. The same situation happens to x_3 and x_4. In a word, the solution set along Pareto frontier contains only two types of choices.

To further spread out the solutions discovered as much as possible along the Pareto frontier, we adopt the niching methods, which are illustrated in the next subsection.

E. NICHING METHODS FOR PACO

Evolutionary algorithms (EAs) have the tendency to lose diversity within their population of feasible solutions and to converge to a single solution [20], [26], which is the common problem of metaheuristic approaches. Niching methods are techniques that promote the formation and maintenance of stable subpopulations. Holland [27] stated that a niche is associated with a fixed *payoff* at every timestep. If some niches become overcrowded, it is to the advantage of individuals occupying those niches to seek out less crowded niches. The methods that utilize this concept are called *sharing methods*. Goldberg *et al.* [28] introduced *fitness sharing* with the sharing functions.

The niching method has recently been shown to be highly effective. It can be applied to not only the EAs but also to a series of metaheuristic approaches, as well as PACO, to maintain the multiple optima in multi-objectives optimization. It is inevitable that after some number of iterations, the pheromone above some certain paths will affect the next node chosen. In this paper, in which every Pareto solution is considered as an individual, we incorporate the niching method into PACO by enforcing the pheromone concentration above the minimum niching count solution to increase its probability to be explored as well as by decreasing the pheromone above the maximum niching count solution. The niching count of an element $\boldsymbol{x}_i$ is

$$niche(\boldsymbol{x}_i) = \sum_{j=1}^{C_s} sh(d(\boldsymbol{x}_i, \boldsymbol{x}_j)) \tag{12}$$

where C_s is the number of solutions. $sh(d(\boldsymbol{x}_i, \boldsymbol{x}_j))$ is a function of the distance $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ between two solutions; it returns 1 when the elements are identical and 0 otherwise if they exceed some threshold of dissimilarity, which is specified by a constant σ_{shared}. If the distance between two population elements is greater than or equal to σ_{shared}, they do not affect each other's shared fitness. The most commonly used sharing functions are the following:

$$sh(d(\boldsymbol{x}_i, \boldsymbol{x}_j)) = \begin{cases} 1 - \frac{d(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sigma_{shared}}, & if\ d(\boldsymbol{x}_i, \boldsymbol{x}_j) < \sigma_{shared} \\ 0, & otherwise \end{cases} \tag{13}$$

where the power of $\frac{d(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sigma_{shared}}$ is a constant (typically set to 1) and is used to regulate the shape of the sharing function. The

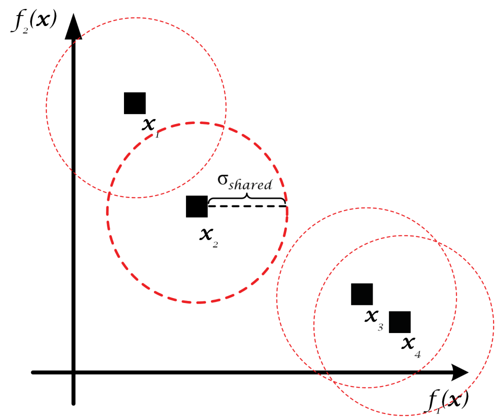


FIGURE 7. Niching shared radius σ_{shared}.

σ_{shared} constant can be defined as

$$\sigma_{shared} = \sum_{i=1}^{C_s} \frac{d_i}{C_s} \tag{14}$$

where we adopt the Hamming distance d_i as the distance measure, $d_i = \min d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \min_{i \neq j}(||\boldsymbol{x}_i - \boldsymbol{x}_j||)$. The niching shared radius σ_{shared} is shown in figure 7.

As is shown in figure 7, for solutions $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, there is only one solution within their own niching; therefore, for both of them, the niching count is 1. For $\boldsymbol{x}_3$ and $\boldsymbol{x}_4$, the niching count is 2. We can force the pheromone concentration above the minimum niching count solution to increase its probability to be explored. The theoretical definition of ACO convergence is that most ants choose the same solution during one iteration [29]. The pseudocode of the niching Pareto ant colony optimization (NPACO) algorithm is shown in Algorithm 1.

IV. SIMULATION EXPERIMENT AND ANALYSIS

In this section, we show the simulation results under different setting. The experiments in Subsection A show the performance of restricted and simplified area algorithm. And then we evaluate the elitist ants and weakened strategy on TSPlib data source. Finally, we perform our NPACO on the transportation networks of Fuzhou, China.

A. RESTRICTED AND SIMPLIFIED SEARCH AREA

Before restricting the search area, it is important to determine the semimajor axis of the ellipse mentioned above. Take the most representative area as the statistical sample. The abundant resources for vertical and horizontal transportation networks in the Gulou District reflect the superiority of the Fuzhou transportation networks. The sample area is shown in figure 8:

Divide the selected district, which includes 180 nodes, into two sets, and calculate their Cartesian product as set C. The ratio of the Euclidean distance E_{ab} to the shortest path

Algorithm 1 Algorithm of Niching Pareto Ant Colony Optimization

Input: *nodes information, number of ants M*
Initialization: *pheromone matrix*
  while Algorithm does not converge. **do**
    while All ants have not finished searching. **do**
      Reset nodes as unvisited;
      while Ant does not reached terminal node T. **do**
        if $q \leq q_0$ **then**
          Choose node ruled by (9);
        else
          Choose node ruled by (10);
        end if
        Set node as visited;
      end while
    end while
    if Find a better path $\boldsymbol{x}$ **then**
      Weaken the pheromone of the suboptimal path using (7);
    end if
    Perform pheromone local update using (11);
    Update Pareto solution set according to *Definition* 1;
    Reinforces pheromone of minimum niching count solution using (12);
  end while

FIGURE 8. Statistical sample in Gulou District.

length P_{ab} of each element in the Cartesian product is shown in figure 9.

According to figure 9, all the ratios are above the line $E_{ab} = P_{ab}$. In addition, 95% of the ratios are below the line $E_{ab0.95} = 1.3366P_{ab}$. 80% of the ratios are located below the line $E_{ab0.8} = 1.2008P_{ab}$, while half of the ratios are below the line $E_{ab0.5} = 1.1103P_{ab}$. The ratio depends on the actual distribution of the transportation network. If the network is dense and complex, E_{ab} will approach P_{ab}; therefore, most R_{ab} values will be approximately 1. Otherwise, the R_{ab} values will be scattered about. If the R_{ab} values are crowded together (dense network), most R_{ab} values will be less than a certain value, which is relate to the range of the restricted search area. Without loss of generality, we take the endpoints of a diagonal as the start node and terminal node, attempting to eliminate the influence of the scale as much as possible. Consider three different network scales: 7707 nodes, 19717 nodes and 57917 nodes; the time costs of the three kinds of restricted approaches at different confidence levels are shown in TABLE 2.

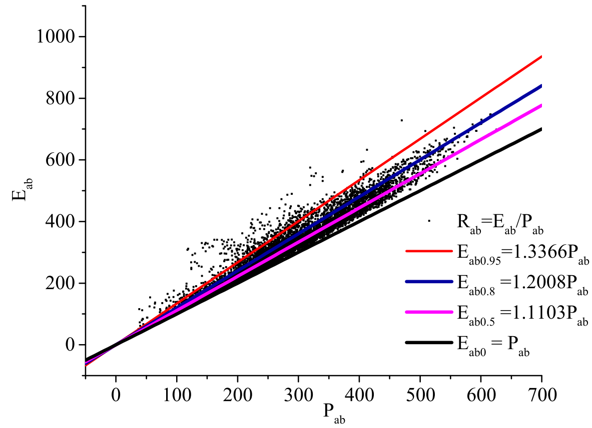


FIGURE 9. Ratio R_{ab} of the statistical sample.

TABLE 2. Restricted search area approaches at different confidence levels (0.95, 0.8, 0.5).

Restricted approach	Node scale	Restricted time cost (ms) 0.95	0.8	0.5	Restricted node scale 0.95	0.8	0.5
Ellipse	7707	4	4	4	6237	5393	3848
	19717	10	10	10	14857	11188	8170
	57911	31	33	30	42518	36181	30705
Minimum bounding rectangle	7707	2	3	2	6400	5717	4438
	19717	6	6	5	16302	12987	9479
	57911	11	11	10	44016	38798	32990
Coordinate parallel bounding rectangle	7707	1	1	1	7707	7630	6648
	19717	3	4	3	17661	14931	12494
	57911	9	9	8	46653	42922	39353

According to TABLE 2, the larger the transportation network, the more restricted the time cost. Take the ellipse approach for example. When the confidence level (CL) is 0.8, 7707 nodes require 4 ms, 19717 nodes require 10 ms and 57917 nodes require 33 ms; thus, the time cost is proportional to the node scale. For the same node scale, the time cost of the ellipse approach is the greatest, followed by that of the minimum bounding rectangle, while the coordinate parallel bounding rectangle approach is the most time-efficient. After restriction, by contrast, the number of nodes of the ellipse approach is minimum. The 7707 nodes are reduced to 69.9%, the 19717 nodes are reduced to 56.7%, and the 57917 nodes are reduced to 62.5%; the statistics of the coordinate parallel bounding rectangle approach are 99.0%, 75.0% and 74.1%, respectively.

Every approach has to judge whether a node is within the restricted area. The ellipse approach's judging standard requires frequent square and trigonometric function

TABLE 3. Road network simplification at different confidence levels (0.95, 0.8, 0.5).

Restricted approach	Node scale	Simplified time cost (ms)			Simplified node scale		
		0.95	0.8	0.5	0.95	0.8	0.5
Ellipse	7707	39	33	37	6625	5263	3828
	19717	82	120	71	14384	10755	7970
	57911	1206	1090	740	40663	34702	29632
Minimum bounding rectangle	7707	40	35	30	6733	5544	4318
	19717	74	48	34	15799	12790	9259
	57911	1284	1539	1417	43024	37961	31428
Coordinate parallel bounding rectangle	7707	47	44	39	7422	7315	6678
	19717	28	119	34	17101	14398	11590
	57911	1349	1813	1038	44522	40914	37710

operations, leaving no doubted that its time cost is maximum. On the contrary, the coordinate parallel bounding rectangle approach involves only the subtraction operation; therefore, it is the most time-efficient.

It is efficient to simplify the transportation network after restricting the search area. The transportation network nodes after simplification are shown in TABLE 3.

From TABLE 3, the node scale is reduced again. The time cost is again almost proportional to the node scale. In addition, the smaller the CL, the lower the node scale.

Compared to the pathfinding algorithm, both the restricted and simplified time costs are too tiny to affect the whole time cost. Focus should be placed on whether these prework time costs could compensate for the time cost reduction due to the decrease in node scale, as demonstrated in TABLE 5.

B. IMPROVED ANT COLONY OPTIMIZATION

In this subsection, we first evaluate the EAWS-ACO algorithm (denoted as EAWS for convenience) using the TSPlib data source and then apply it to a one-objective pathfinding problem.

The TSPlib data source is the universal criteria used to judge and evaluate approaches to solve the TSP(Traveling Salesman Problem) problem. In this paper, we choose pr136.tsp as the main data source. Without loss of generality, the verification process requires 500 iterations, the algorithm must be executed 100 times, and the average value is taken as the output. Whenever a solution remains unchanged over 200 iterations, we consider the algorithm as having converged. Figure 10 shows the simulation results for pr136.tsp.

As shown in figure 10, the EAWS drops-off a little bit faster than ACO, although they both collapse at the beginning. However, after the 200th iteration, the EAWS has a tendency to converge, while ACO still requires some time. Because of the introduction of weakened strategy, when the shortest path in a single iteration is constantly strengthened, the pheromone above is higher than that of the other path, affecting the transition probability, which makes the algorithm converge faster. Moreover, whenever a better path is found, the pheromone above the last suboptimal path found is weakened, alleviating the accumulated pheromone in the formal iteration. Adopting this strategy helps to speed up the convergence of the algorithm. TABLE 4 shows the results of comparison among different algorithms.

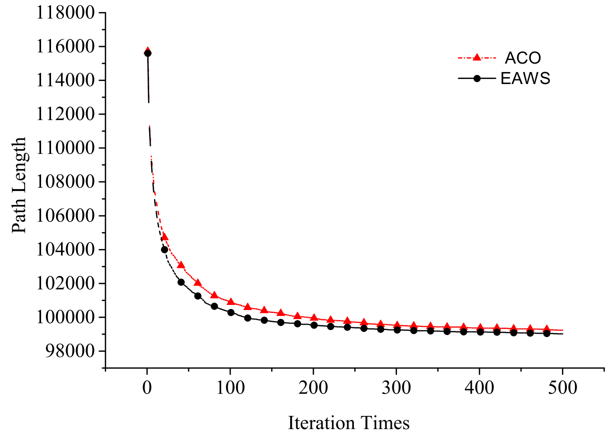


FIGURE 10. Simulation results for pr136.tsp.

TABLE 4. Results of comparison among different algorithms.

Algorithm	Item	Pr136	KroA100	Ch130
Official records	optimal value	96772	21282	6110
ACO	optimal value	97050	21335	6138
	average value	99083	21650	6224
	relative error (%)	0.29	0.25	0.46
EAWS	optimal value	96910	21285	6110
	average value	99017	21564	6219
	relative error (%)	0.14	0.01	0

TABLE 5. Average time cost of shortest pathfinding process using different restricted approaches of one ant.

Restricted & simplified	Confidence level (CL)					
	0.95	0.9	0.8	0.7	0.6	0.5
Original	1414.7					
Ellipse	876.5	682.9	428.4	358.2	189.7	98.1
Minimum bounding rectangle	953.3	867.4	650.8	405.3	338.7	227.1
Coordinate parallel bounding rectangle	1332.6	1262.4	1201.6	1094.5	949.1	560.8

The relative error is often used to compare approximations of numbers of widely different sizes; it is the absolute error divided by the magnitude of the exact value. We use the relative error as the evaluation index. According to TABLE 4, the EAWS performs better than ACO regardless of the data source. Compared to the TSPlib official record, the gap is quite small. The shortest path of Pr136 is 96910, which is close to the official record of 96772, and the relative error is 0.14%. Ch130 reaches its official record of 6110. The best path of KroA100 is 21285, with the relative error being 0.01%.

Applying the EAWS to the pathfinding problem, we take the 7707 node transportation network as the data source with 100 ants in one iteration; the average time cost of the shortest pathfinding process using different restricted EAWS approaches of one ant is shown in TABLE 5. The corresponding path is shown in figure 11. It is the final path from node 2600 to node 5800 under the different confidence intervals.

According to TABLE 5 and figure 11, as the CL decreased, the time cost decreased, and a lower confidence level results in a smaller search range, increasing the risk of missing the

FIGURE 11. Shortest path found with different restricted approaches under CL equaling to 0.9, 0.7, 0.5.

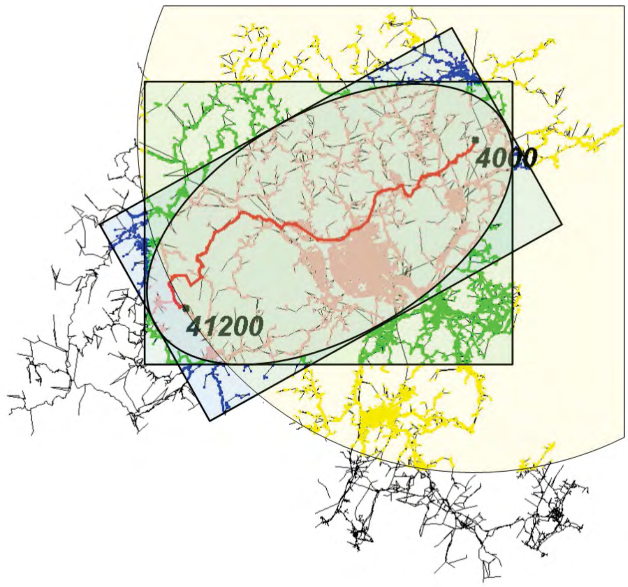


FIGURE 12. Shortest pathfinding process for 59717 nodes (CL=0.8).

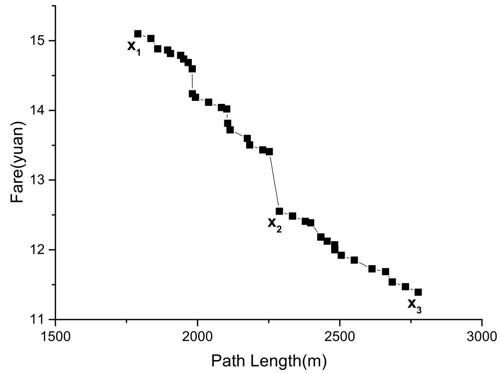
shortest path. The cost of the coordinate parallel bounding rectangle restricted approach is smallest at every confidence level in TABLE 2, but it takes more time for the ants to complete each pathfinding because it has the the largest scale of searching area. So it has the largest value in TABLE 5. Instances of other scales for CL=0.8 are shown in figure 12.

C. NICHING PARETO ANT COLONY OPTIMIZATION FOR BSP

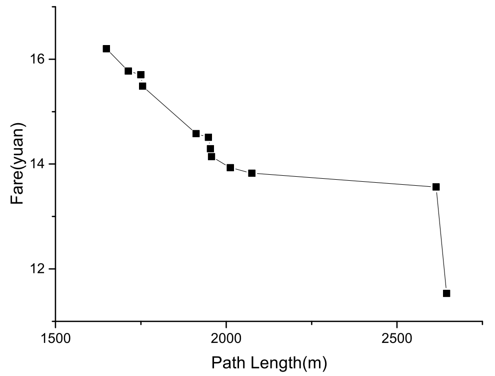
We adopt the original PACO [9] and BINAT [11] as comparison algorithms to evaluate our NPACO. The parameters setting in all the three algorithms are as follows. The iterations are set as 1000, the number of ants is 200, α is set as 2, β is set as 5, p is set as 0.5, and q_0 is set as 0.6.

We adopt different metrics, such as Hypervolume [6], Spread [30], PF [11] and Non-dominated Vector Generation Ratio(ONVGR) [13], to evaluate our NPACO from different scopes. The Hypervolume computes the volume covered by a series of non-dominated solutions in target space. The higher the value is, the better the result is. The Spread illustrates the level of a set of non-dominated solutions. It takes the Euclidean distance between consecutive solutions on average and extreme distances into account, and a small value stands for a excellent result. The PF is the number of non-dominated solutions gained by the pareto solutions. The ONVGR refers to the ratio of the number of solutions to the number of solutions in the true optimal Pareto set. And a bigger value denotes a better result in theory.

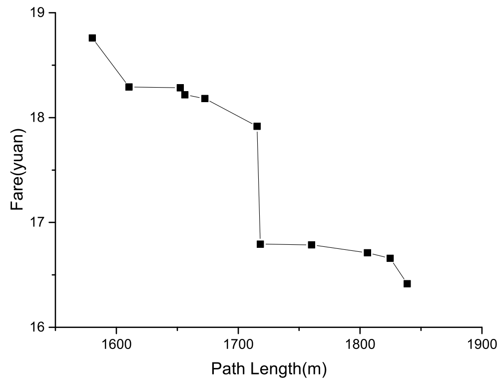
We executed the three algorithms 200 times to get the true optimal Pareto set, which is formed by all the non-dominated



(a)



(b)



(c)

FIGURE 13. Pareto frontiers of BINAT, PACO and NPACO. (a) Pareto frontier of NPACO. b) Pareto frontier of BINAT. (c) Pareto frontier of PACO.

TABLE 6. The metrics comparison among different algorithm.

CL	Algorithm	Spread	Hypervolume	PF	ONVGR
0.95	**NPACO**	0.8642	0.3063	11.75	0.7344
	BINAT	0.8947	0.2298	6.15	0.3844
	PACO	0.8916	0.2636	7.7	0.4813
0.80	**NPACO**	0.7941	0.4087	16.05	1.00
	BINAT	0.9436	0.3075	5.7	0.3563
	PACO	0.8163	0.3655	8.9	0.5563

FIGURE 14. Specific paths of Pareto frontier. (a) The shortest path length. (b) The moderate path length and fare. (c) The least fare.

solutions obtained during the execution. The jMetal software [31] is adopted to compute the first three metrics.

Without loss of generality, we take the 7707 node transportation network as the data source, and set 4000 as the starting node and 5800 as the terminal node. For each algorithm, we perform 20 times in two Confidence levels (0.95 and 0.8), and get the average value of every indicator. The results are shown in TABLE 6, and the best solutions from BINAT, normal PACO and NPACO are shown in figure 13.

From TABLE 6, it is easy to find that NPACO outperforms PACO and BIANT in each metric. For instance, when CL equals to 0.95, the Hypervolume of NPACO increases by 33.28% and 16.19% than BINAT and PACO, respectively. And the Spread decreases by 3.41% and 3.07%. Furthermore, the advantages in metrics are also reflected in figure 13. According to figure 13, none of the solutions dominate the others, and neither the fare nor the path length takes advantage of other solutions. The path length and fare cannot reach their minimum values at the same time. Obviously, the frontier of NPACO is distributed much more uniformly than that of PACO and BIANT. The distance ranges from 2000 to 2400, and only a finite number of PACO and BIANT solutions are explored, while NPACO yields a slightly continuous and compact solution set. In other words, both the PACO and the BINAT find as many non-dominated solutions as possible and return a set of non-dominated solutions to the user so that a final decision can be made. However, they cannot avoid becoming trapped in a local convergence. NPACO, which combines PACO with a niching uniform distribution mechanism, is not only an approach that supplies one or two solutions but also an absolutely multi-objective algorithm that offers a solution set at the largest scale.

In figure 14, we list some specific paths obtained by NPACO, which are the ones with (a) the shortest path length, (b) the moderate path length and fare, and (c) the least fare. The three paths are corresponding to the solutions x_1, x_2 and x_3 in figure 13(a).

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a niching Pareto ant colony optimization to solve the bi-objective path planning problem for an urban city. We first restrict the search area and simplify the road networks to greatly decrease the pathfinding computation complexity. Then, we combine the niching method and the Pareto ant colony optimization method to guide the ant colony towards the Pareto optimal solutions rapidly. We conduct a thorough experiment on the transportation networks of the city of Fuzhou. The experimental results show that NPACO, which has a restricted search area, returns a continuous, uniform distribution along the Pareto frontier within a short amount of computing time.

In future research, the transportation network simplification methods should be applied in more comprehensive ways to reduce the number of redundant nodes, as the number of nodes has a tremendous time effect on PACO. Moreover, it is important to guarantee diverse solutions over an efficient frontier; thus, the more mature niching methods used in the genetic algorithm or evolutionary algorithm could be modified for use in PACO. In addition, the successful experiences of other heuristic algorithms, such as an initial attempt to integrate the core idea of PSA (Pareto Simulated Annealing) into NPACO to keep solutions isolated from each other, should be considered.

REFERENCES

[1] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011.

[2] A. Raith and M. Ehrgott, "A comparison of solution strategies for biobjective shortest path problems," *Comput. Oper. Res.*, vol. 36, no. 4, pp. 1299–1331, 2009.

[3] A. A. Freitas, "A critical review of multi-objective optimization in data mining: A position paper," *ACM SIGKDD Explorations Newslett.*, vol. 6, no. 2, pp. 77–86, 2004.

[4] P. Cao and Z. Wang, "Efficient top-k query calculation in distributed networks," in *Proc. ACM Symp. Principles Distrib. Comput. (PODC)*, St. John's, NF, Canada, Jul. 2004, pp. 206–215.

[5] Z. Al Chami, H. Manier, and M.-A. Manier, "A lexicographic approach for the Bi-objective selective pickup and delivery problem with time windows and paired demands," *Ann. Oper. Res.*, vol. 2, pp. 1–19, Apr. 2017.

[6] 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.

[7] G. Moslehi and M. Mahnam, "A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search," *Int. J. Prod. Econ.*, vol. 129, no. 1, pp. 14–22, 2011.

[8] A. Saraswat, A. Saini, and A. K. Saxena, "A novel multi-zone reactive power market settlement model: A Pareto-optimization approach," *Energy*, vol. 51, no. 2, pp. 85–100, 2013.

[9] 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, 2004.

[10] M. Dorigo and G. Di Caro, "Ant colony optimization: A new meta-heuristic," in *Proc. IEEE Proc. Congr. Evol. Comput. (CEC)*, vol. 2. Jul. 1999, pp. 1470–1477.

[11] A. M. Mora, P. García-Sánchez, J. J. Merelo, and P. A. Castillo, "Pareto-based multi-colony multi-objective ant colony optimization algorithms: An island model proposal," *Soft Comput.*, vol. 17, no. 7, pp. 1175–1207, 2013.

[12] M. López-Ibánez and T. Stutzle, "The automatic design of multiobjective ant colony optimization algorithms," *IEEE Trans. Evol. Comput.*, vol. 16, no. 6, pp. 861–875, Dec. 2012.

[13] B. Baran and M. Schaerer, "A multiobjective ant colony system for vehicle routing problem with time windows," in *Proc. IASTED Int. Multi-Conf. Appl. Inform.*, 2003, pp. 97–102.

[14] A. M. Mora, J. J. Merelo, P. A. Castillo, and M. G. Arenas, "hCHAC: A family of MOACO algorithms for the resolution of the bi-criteria military unit pathfinding problem," *Comput. Oper. Res.*, vol. 40, no. 6, pp. 1524–1551, 2013.

[15] Z.-H. Jia, Y.-L. Zhang, J. Y.-T. Leung, and K. Li, "Bi-criteria ant colony optimization algorithm for minimizing makespan and energy consumption on parallel batch machines," *Appl. Soft Comput.*, vol. 55, pp. 226–237, Jun. 2017.

[16] J. Hou, W. Mi, and J. Sun, "Optimal spatial allocation of water resources based on Pareto ant colony algorithm," *Int. J. Geogr. Inf. Sci.*, vol. 28, no. 2, pp. 213–233, 2014.

[17] J. M. Pasia, R. F. Hartl, and K. F. Doerner, "Solving a Bi-objective flowshop scheduling problem by Pareto-ant colony optimization," in *Proc. Int. Workshop Colony Optim. Swarm Intell.*, 2006, pp. 294–305.

[18] J. M. Pasia, K. F. Doerner, R. F. Hartl, and M. Reimann, "Solving a Bi-objective vehicle routing problem by Pareto-ant colony optimization," in *Engineering Stochastic Local Search Algorithms. Designing, Implementing and Analyzing Effective Heuristics*. Berlin, Germany: Springer, 2007, pp. 187–191.

[19] S. Iredi, D. Merkle, and M. Middendorf, "Bi-criterion optimization with multi colony ant algorithms," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.*, 2001, pp. 359–372.

[20] S. W. Mahfoud, "Niching methods for genetic algorithms," *Urbana*, vol. 51, no. 95001, pp. 62–94, 1995.

[21] V. Sánchez-Anguix, S. Valero, V. Julián, V. Botti, and A. García-Fornes, "Evolutionary-aided negotiation model for bilateral bargaining in Ambient Intelligence domains with complex utility functions," *Inf. Sci.*, vol. 222, pp. 25–46, Feb. 2013.

[22] R. Mukherjee, G. R. Patra, R. Kundu, and S. Das, "Cluster-based differential evolution with crowding archive for Niching in dynamic environments," *Inf. Sci.*, vol. 267, pp. 58–82, May 2014.

[23] D. Angus, "Niching for population-based ant colony optimization," in *Proc. 2nd e-Sci. IEEE Int. Conf. e-Sci. Grid Comput.*, Dec. 2006, p. 115.

[24] L. Feng, L. Dongmei, and C. Weihong, "Time shortest path algorithm for restricted searching area in transportation networks," *J. Image Graph.*, vol. 4, no. 10, pp 849–853, 1999.

[25] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.

[26] D. B. Fogel, *Evolutionary Algorithms in Theory and Practice*. London, U.K.: Oxford Univ. Press, 1997.

[27] J. H. Holland, *Adaptation in Natural & Artificial Systems—An Introductory Analysis With Application to Biology, Control, and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

[28] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proc. 2nd Int. Conf. Gen. Algorithms Gen. Algorithms Appl.*, Hillsdale, NJ, USA, 1987, pp. 41–49.

[29] W. J. Gutjahr, "A generalized convergence result for the graph-based ant system metaheuristic," *Probab. Eng. Inf. Sci.*, vol. 17, no. 4, pp. 545–569, 2003.

[30] E. Zitzler and L. Thiele, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: Wiley, 2001.

[31] J. J. Durillo, A. J. Nebro, and E. Alba, "The jMetal framework for multi-objective optimization: Design and architecture," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1–8.

DONG LIN received the B.S. degree in radio technology and the M.S. degree in communication and information systems from Fuzhou University, Fuzhou, China, in 1991 and 1999, respectively, and the Ph.D. degree in signal and information processing from Shanghai Jiao Tong University, China, in 2006. He is currently an Associate Professor with the College of Physics and Information Engineering, Fuzhou University. His research interests include video signal processing, mobile communication, and information security.

LINGCHAO HE received the B.S. degree in electronic information engineering from the Henan University of Science and Technology, Henan, China, in 2015. He is currently pursuing the M.S. degree in communication and information system with Fuzhou University, Fuzhou, China. His research interests include big data analysis for Internet of vehicles, tensor analysis, and crowd and participatory sensing.

XINXIN FENG received the B.S. and M.S. degrees in communication engineering from the Nanjing University of Science and Technology, China, in 2006 and 2008, respectively, and the Ph.D. degree in information and communication engineering from Shanghai Jiao Tong University, China, in 2015. She is currently an Associate Professor with the College of Physics and Information Engineering, Fuzhou University, Fuzhou, China. Her research interests include incentive mechanism design in participatory sensing networks, dynamic spectrum sharing in cognitive radio networks, game theory, and market theory.

WEI LUO received the M.S. degree in information and communication engineering with Fuzhou University, Fuzhou, China, in 2017. His research interests include big data analysis and machine learning.