Applied Intelligence (2023) 53:26690–26707
https://doi.org/10.1007/s10489-023-04772-5

Dual ant colony optimization for electric vehicle charging infrastructure planning

Junzhong Ji[1] · Yuefeng Liu[1] · Cuicui Yang[1]

Accepted: 4 June 2023 / Published online: 28 August 2023

Abstract

Charging infrastructure planning (CIPL) is key to popularizing electric vehicles and reducing carbon emissions. CIPL consists of two subproblems: charging station siting and charging pile allocation. The existing methods independently solve the two subproblems and ignore their interaction, which restricts the rationality of CIPL. To address this issue, this paper proposes a dual ant colony optimization for CIPL (DACO-CIPL). In each iteration, under the guidance of heuristic information and pheromones, the upper and lower ant colonies construct solutions for charging station siting and charging pile allocation in turn, respectively. Then, a global pheromone update strategy is performed to update the pheromones of each ant colony according to the historical best solutions, which realizes information transmission from the lower ant colony to the upper ant colony. In addition, whenever the upper ant colony finishes constructing solutions, a pheromone enhancement strategy is used to strengthen the pheromones of the lower ant colony according to the solutions of the upper ant colony, which realizes information transmission from the upper ant colony to the lower ant colony. DACO-CIPL is compared with several algorithms on multiple test instances. The experimental results show that DACO-CIPL has superior performance and more reasonable options for CIPL.

Keywords Charging infrastructure planning · Charging station siting · Charging pile allocation · Heuristic algorithms · Ant colony optimization

1 Introduction

With the excess emissions of carbon dioxide, the aggravation of environmental and climate problems has attracted increasing attention. Many countries have established plans to achieve carbon neutrality, but global carbon emissions are still growing [1]. Since the transportation industry is one of the largest carbon emitters, replacing fuel vehicles with electric vehicles (EVs) is a universally recognized way to reduce carbon emissions and protect the environment [2]. Thus, charging infrastructure construction becomes the foundation for EV development. However, there are still many problems in the charging infrastructure planning (CIPL), such as unreasonable charging station siting [3] and charging pile allocation [4], high economic cost [5] and unsatisfied charging demand [6, 7]. Therefore, it is very important to utilize CIPL for developing EVs and further improving the environment and climate.

In general, CIPL consists of two subproblems: the charging station siting problem and the charging pile allocation problem. The first subproblem aims to select charging station sites that maximize the coverage of charging demand, while the second subproblem aims to determine the charging pile number of each charging station that minimizes the charging station construction and operation costs. The charging demand determines the number of EVs waiting for charging at the stations, thereby indirectly impacting the operating cost. As such, the two subproblems are correlated, and some studies have started to solve them. However, the existing methods often solve these two subproblems independently, neglecting their correlation and only focusing on maximizing the coverage of charging demand by siting and then allocating charging piles.

Metaheuristic algorithms are optimization algorithms inspired by natural laws, and can be classified into four categories [8]: evolution-based, swarm-based, physics-based and human-based methods. Among them, swarm-based algorithms can split complex optimization problems into multiple parts and assign them to different populations for independent optimization, considering multiple factors and obtaining better solutions [9]. This mechanism of swarm-based algorithms is well suited for solving the two CIPL subproblems. Ant

✉ Cuicui Yang
yangcc@bjut.edu.cn

[1] Beijing Municipal Key Laboratory of Multimedia and Intelligent Software Technology, College of Computer Science, Beijing University of Technology, Beijing, China

colony optimization (ACO) [10] is also a swarm-inspired algorithm, and is promising for addressing CIPL due to the following two additional reasons: I) ACO has been successfully applied to solve planning problems related to the charging station sites [11], and 2) ACO guides ants in a colony to search for solutions using the positive feedback mechanism of pheromones. If this mechanism is used to transmit information between two ant colonies, it may achieve stronger population cooperation than other algorithms, comprehensively considering both CIPL subproblems.

Based on the above analysis, this paper proposes a dual ant colony optimization for CIPL (DACO-CIPL). The proposed algorithm first uses two ant colonies to separately solve the two CIPL subproblems, and then interacts with the two subproblems by two pheromone feedback mechanisms. More specifically, two ant colonies construct solutions by walking on their search graphs under the guidance of heuristic information and pheromones. In each iteration, the upper ant colony constructs solutions to the charging station siting problem. According to all the solutions obtained by the upper ant colony, the lower ant colony constructs corresponding solutions of the charging pile allocation. At the end of each iteration, a global pheromone update strategy is performed to update the pheromone of each ant colony according to the historical best solutions, which realizes information transmission from the lower ant colony to the upper ant colony. In addition, whenever the upper ant colony finishes the process of constructing solutions, a pheromone enhancement strategy is used to strengthen the pheromone of the lower ant colony according to the solutions obtained by the upper ant colony, which realizes information transmission from the upper ant colony to the lower ant colony. The main contributions of this paper are summarized as follows:

1. This paper proposes a novel dual ant colony optimization for CIPL, which addresses the issue of the existing methods (i.e., not considering the correlation between two subproblems: charging station siting and charging pile allocation).
2. 2. This paper proposes new pheromone adjustment strategies that achieve bidirectional information transmission between two ant colonies and better address the correlation between the two CIPL subproblems.
3. This paper compares DACO-CIPL with several competitive algorithms on multiple test instances. The experimental results show that DACO-CIPL has superior performance and is promising for providing more reason able CIPL options.

The rest of this paper is arranged as follows. Section 2 discusses the related work. Section 3 introduces two mathematical subproblem models. The proposed algorithm is described in detail in Section 4. Section 5 is dedicated to the experimental study of the proposed algorithm. Finally, Section 6 concludes this paper.

2 Related work

According to the literature, research on CIPL can be traced back to 2010. Pan et al. [12] created a charging station siting model by minimizing the charging load and used integer linear programming (ILP) to solve it, which first began re search on CIPL. Since then, there have been many studies on CIPL. However, most of the existing studies only focus on the charging station siting problem, while some recent studies have begun to simultaneously on the two subproblems. According to the realization mechanism, all the methods for the charging station siting problem can be di vided into two categories: ILP [13] and metaheuristic algorithms [14].

The ILP method models the charging station siting problem as an integer linear optimization problem and uses tradi tional deterministic optimization methods to select charging station sites. For example, He et al. [15] modeled the electric bus charging station siting problem as a mixed-integer ILP model with the objective of minimizing the construction cost of charging stations and the charging cost for buses. They used the CPLX solver to solve the model. Ahmed et al. [16] developed a stochastic ILP model considering the multiperiod decision horizon of charging stations and the uncertainty in charging demand. They used an exact solution method based on Benders decomposition to solve the model. Orner et al. [17] developed a mixed-integer ILP model with the objective of minimizing charging costs and designed two model variants to address the charging station location problem with or without charging station number constraints. They also proposed a Benders decomposition algorithm to solve the model. In summary, the ILP method can quickly solve the charging station siting problem, but it is only suitable for small-scale problems with fewer candidate charging station sites. Moreover, this category of method requires that the optimization objectives and constraint functions of the optimization model be linear.

With the development of charging infrastructure and EVs, there are more factors to be considered for the charging station siting problem. That is, the optimization model may involve more candidate charging station sites, which makes the scale become larger and no longer satisfy the linear requirement. Faced with such a situation, some researchers began to employ a metaheuristic algorithm to solve the charging station siting problem. For example, Hou et al. [18] considered the relationship between EV users and the power grid and established a problem model with the objective of minimizing social costs. They proposed an improved particle swarm optimization algorithm (PSO) to solve the model. Ma et al. [19] proposed a model to minimize the distance traveled by EVs to reach charging stations, taking into account parking queues, and utilized an immune algorithm to solve the problem. Zhou et al. [20] built a large-scale charging station siting model by minimizing the economic cost and adopted a genetic algorithm (GA) to solve it. Although the above methods can well address the charging station siting problem with various factors, they do not address how to allocate charging piles for each charging station.

Until recently, some researches began to simultaneously solve the charging station siting problem and the charging pile allocation problem. For example, Zhang et al. [21] first proposed a fast greedy algorithm to address the charging station siting problem by maximizing the charging demand, and then used M/M/N/N queuing theory to deal with the charging pile allocation problem by minimizing the queuing probability of EVs. Li et al. [22] proposed a PSO algorithm to deal with the two subproblems at the same time, where the objectives of the two subproblems are to minimize the operation cost and the time cost, respectively. Dai et al. [23] proposed an ascent heuristic algorithm (AHA) to deal with the two subproblems. AHA first determined the charging station sites to maximize the profit, and then allocated the charging piles for each charging station to minimize the demand loss rate. In summary, although these methods provide the solving methods for the two subproblems of CIPL, they deal with the two subproblems independently and ignore the interconnections and influences of the two subproblems, which may lead to unreasonable results. The reason is that there may be conflict between two optimization objects for the two subproblems. If the two subproblems can be comprehensively considered in a framework, the solution of CIPL may be more rational.

3 The proposed mathematical models of the two CIPL subproblems

3.1 Basic conceptions and notations

Charging stations are responsible for providing charging services for EVs in a given region. In a region, an EV track is numbered by an integer. All the EV track numbers form the set $T=\{1, 2, \ldots, t, \ldots, m\}$, where t represents any track number, and m represents the total number of tracks in the region. Any EV track t is a set that consists of an indefinite number of coordinates, i.e., $K^t=\{k^t, k, \ldots, k^t\}$. In K^t, each entry is a coordinate denoted by the form of (x, y), where x is the dimension and y is the longitude; ${k^t}_1$ is the starting coordinate of track t and ${k^t}_e$ is the ending coordinate of track t. All the candidate charging station site numbers form the set $\{1, 2, \ldots, i, \ldots, n\}$, where i represents any candidate charging station site number, and n represents the total number of candidate charging station sites. The coordinates of all the candidate charging station sites form the set $\{c_1, c_2, \ldots, c_i, \ldots, c_n\}$, where any entry c_i represents the coordinate of the candidate charging station site i. Each candidate charging station site may be selected to construct a charging station. All the charging station numbers form the set $\{1, 2, \ldots, j, \ldots, N\}$, where j represents any charging station number, and N represents the total number of charging stations to be constructed.

For convenience, we define two decision variable categories, x_j and y_j. When $x_j=i$, candidate charging station site i is selected as the charging station construction site j. When $y_j=r$, the number of charging piles allocated is r at charging station j. Any solution of the charging station siting problem is denoted by $X=(x_1, x_2, \ldots, x_j, \ldots, x_N)$. Any solution of the charging pile allocation problem is denoted by $Y=(y_1, y_2, \ldots, y_j, \ldots, y_N)$.

Figure 1 shows a simplified example of CIPL. In a given region, there are five candidate charging station sites, whose numbers form the set $\{1,2,3,4,5\}$. Among them, only three candidate charging stations, namely, candidate charging station 1, candidate charging station 3 and candidate charging station 4, are selected to construct charging stations, i.e., the charging station siting problem solution is $X=(x_1, x_2, x_3)=(1, 3, 4)$. According to the number of charging piles allocated shown in Fig. 1, the corresponding solution of the charging pile allocation problem is $Y=(y_1, y_2, y_3)=(4, 3, 2)$.

3.2 The mathematical model of the charging station siting problem

Since charging stations serve to satisfy the charging demand of EVs in a region that requires CIPL, the objective function of the charging station siting problem is naturally defined to maximize the total charging demand in this paper. The total charging demand is related to two factors: the number of tracks covered by all charging stations and the charging demand of each track.

A) Number of tracks covered by all charging stations: The number of tracks covered by all charging stations is related to the amount of charging demand in a region where CIPL is conducted. The more tracks covered, the larger the charging demand. The number of tracks covered by all charging stations is defined as:

$$T(X) = \sum_{j=1}^{N} \sum_{t\in T} s\left(x_j, t\right) \tag{3.1}$$

where $s(x_j, t)$ is a function to indicate whether the track t is covered by the charging station j and is defined as:

$$s\left(x_j, t\right) = \begin{cases} 1, & d_{x_j t} < 1km \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

where $dx_{\ j}\ t$ is the distance from charging station j to track t, i.e., the distance from charging station j to the straight line connecting the starting coordinate to the ending coordinate of track t. When $s(x_j, t)=1$, track t is

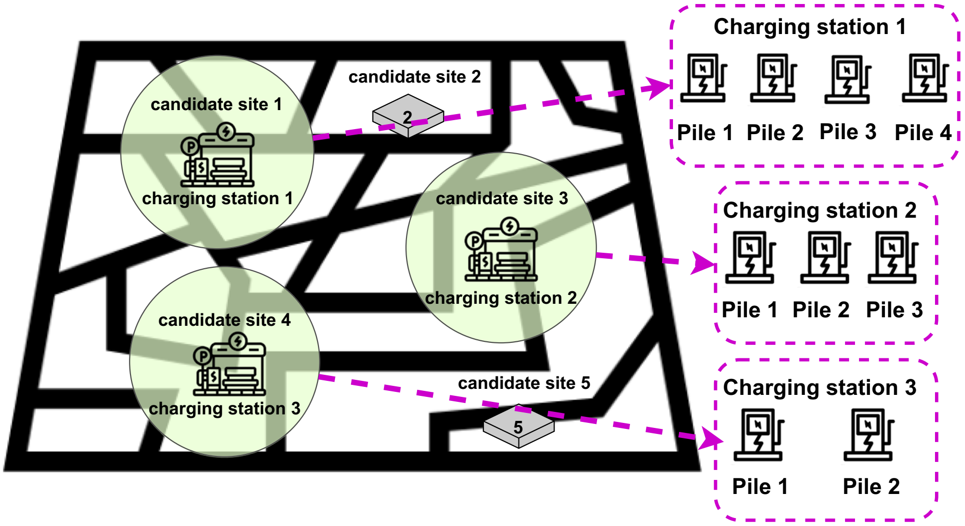

Fig. 1 A simplified example of CIPL in a region

covered by charging station j, and the EV on track t may drive to charging station j if this EV needs to charge.

B) Charging demand of each track: An EV drives away from the starting coordinate, and electric power will continue to be consumed. When the electric power of the EV is too low, it will generate charging demand. The larger the distance between a charging station and the starting coordinate of one track is, the larger the charging demand of the EV. The charging demand of track t is defined as:

$$D(t) = \begin{cases} g_{x_j t}, & x_j = min\{x | x \in X : s(x,t) = 1\} \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

where $g_{x_j t}$ is the distance between charging station j and the starting coordinate of track t and m is the total number of tracks. Thus, the total charging demand of all charging stations in a region that requires CIPL, i.e., the objective function of the charging station siting problem, is defined as:

$$\max F_d(X) = \sum_{t \in T} s\left(x_j, t\right) \bullet D(t) \tag{3.4}$$

where $X = (x_1, x_2 \ldots, x_j \ldots, x_N)$ is the decision vector, i.e., one solution of the charging station siting problem; N is the total number of charging stations to be constructed.

3.3 The mathematical model of the charging pile allocation problem

The solution to the charging pile allocation problem is to determine the number of charging piles for each station by considering some factors. Charging piles provide a charging service for EVs arriving at the charging station. For a charging pile to function normally, this requires costs that we label as economic costs. The lower the economic cost for satisfying the given total charging demand is, the better the charging pile allocation. Different studies usually use different economic costs. In this paper, the economic cost includes the operation cost and the charging cost.

A) Operation cost: The operation cost is determined by the construction cost and the maintenance cost. The construction cost is determined by the number of charging piles. The maintenance cost is reflected by the queuing probability of the charging stations. Charging stations that are often full will be more expensive to maintain. According to [20] and [23], the total operation cost of all charging stations is defined as:

$$Cost_1(X, Y) = \sum_{j=1}^{N} \frac{\sum_{i=1}^{j} y_i}{j \bullet r_{max}} \omega\left(x_j, y_j\right) y_j \bullet c \tag{3.5}$$

where r_{max} is the maximal number of charging piles for a charging station; c is the construction cost of a charging pile; and $\boldsymbol{\omega}$ (y_j, x_j) is the queuing probability at charging station j. According to [24], $\boldsymbol{\omega}(y_j, x_j)$ can be defined as:

$$\omega\left(y_j, x_j\right) = \frac{P_Q\left(\Lambda_{x_j}, y_j\right)}{\Lambda_{x_j}\left(1 - p_j\right)} p_j \tag{3.6}$$

where Λx_j is the demand generation rate of the tracks covered by charging station j and follows the Poisson distribution $\pi\left(\lambda_{x_j}^{max}\right)$. Λ_{x_j} is defined as:

$$\Lambda_{x_j} \sum_{t \in T} \lambda_{x_j}^{max} \frac{1}{1 + d_{x_j t}} \tag{3.7}$$

where $\lambda_{x_j}^{max}$ is the maximum EV charging demand rate at charging station j and $d_{x_j t}$ is the distance from charging station j to track t.

The probability $P_Q\left(\Lambda_{x_j}, y_j\right)$ that all charging piles of charging station j are busy is defined as:

$$P_Q\left(\Lambda_{x_j}, y_j\right) = \frac{\left(y_j p_j\right)^{y_j}}{\left(1-p_j\right)y_j!}\left(\sum_{k=1}^{y_j-1}\frac{\left(y_j p_j\right)^k}{k!}\right)^{-1} \tag{3.8}$$

where the utilization rate p_j of charging station j is defined as:

$$p_j = \Lambda_{x_j}/\sigma y_j \tag{3.9}$$

where σ is the number of EVs that a charging pile can charge at the same time.

B) Charging cost: The charging cost is determined by the distance from each charging station to each track. The farther this distance is, the higher the EV charging cost. The total charging cost of all charging stations can be defined as:

$$Cost_2(X) = \sum_{t=1}^{m}\sum_{j=1}^{N} d_{x_j t} s\left(x_j, t\right) \tag{3.10}$$

where N is the total number of charging stations; $s(x_j, t)$ is a function to indicate whether track t is covered by charging station j, as shown in Eq. (3.2); $d_{x_j t}$ is the distance from charging station j to track t.

In summary, the economic cost of all charging piles in a region that requires charging infrastructure planning is defined as:

$$F_e(X, Y) = Cost_1(X, Y) + Cost_2(X) \tag{3.11}$$

where $X = (x_1, x_2\ldots, x_j\ldots, x_N)$ is the decision vector of the charging station siting problem and $Y = (y_1, y_2\ldots, y_j\ldots y_N)$ is the decision vector of the charging pile allocation problem.

The goal of charging pile allocation is to minimize the economic cost while satisfying the charging demand covered by the charging station sites. To maintain consistency with the maximization form of the objection function of the charging station siting problem, the CIPL model can be defined as:

$$\max\ F(X, Y) = F_d(X)/F_e(X, Y) \tag{3.12}$$

$$\text{subject to } \sum_{j=1}^{N} e\left(x_j, i\right) = 1,\ fori = 1, 2, 3, \cdots, n \tag{3.13}$$

$$\sum_{i=1}^{n} e\left(i, x_j\right) \le 1,\ forj = 1, 2, 3, \cdots, n \tag{3.14}$$

$$\sum_{j=1}^{N} s\left(x_j, t\right) \le 1,\ fort = 1, 2, 3, \cdots, m \tag{3.15}$$

$$\sum_{r=1}^{r_{max}} e\left(r, y_j\right) = 1,\ forj = 1, 2, 3, \cdots, N \tag{3.16}$$

$$1 \le x_j \le n,\ forj = 1, 2, 3, \cdots, N \tag{3.17}$$

$$1 \le y_j \le r_{max},\ forj = 1, 2, 3, \cdots, N \tag{3.18}$$

where $e(x_j, i)$ is a Boolean function. When x_j is equal to i, $e(x_j, i) = 1$; otherwise, $e(x_j, i) = 0$. Constraint (3.13) guarantees that each charging station must be constructed on a candidate charging station site. Constraint (3.14) guarantees that no more than one charging station be constructed at each candidate charging station site. Constraint (3.15) guarantees that each track can only be covered by one charging station. Constraint (3.16) guarantees that each charging station is allocated a certain number of charging piles. Constraint (3.17) and Constraint (3.18) guarantee the range of variables x_j and y_j.

From formula (3.11), the solution to the charging station siting problem also affects the economic cost. To satisfy the corresponding charging demand, different charging station siting options would lead to different charging pile allocation economic costs. Therefore, the charging station siting problem and the charging pile allocation problem should be considered comprehensively and systematically. This is specifically the reason why we defined the objective function in Eq. (3.12). This objective function merges the two factors, charging demand and economic cost, which can evaluate the solutions of two CIPL subproblems comprehensively.

4 Proposed method

In this section, we first introduce the overall framework of DACO-CIPL. Then, we introduce the key strategies of DACOCIPL in detail.

4.1 Overall framework

Figure 2 shows the overall framework of DACO-CIPL. First, a certain number of track starting coordinates and ending coordinates are randomly selected as candidate charging station sites. Then, the heuristic and pheromone information associated with the search graph of each ant colony are initialized to begin the search process. In each iteration, the upper ant colony and the lower ant colony construct solutions for the charging station siting problem and the charging pile allocation problem in turn by walking on their respective search graphs. For all the ants in the two ant colonies, their walking trajectories are determined by a state transition rule according to the heuristic and pheromone information. Every time they select a feasible vertex from the search graph, a local pheromone update strategy is executed to modify the pheromone values on the vertex. When the upper ant colony finishes the process of constructing solutions in each iteration, the pheromone enhancement strategy is executed based on its solution to reinforce the pheromones of the lower ant colony to accelerate the convergence. This pheromone enhancement strategy plays the role of information transmission from the upper ant colony to the lower colony. In addition, when all the ants of two colonies finish their walking in each iteration, a global pheromone update strategy is performed on the vertices in the walking trajectories of the historical best solutions of the two ant colonies. Since the historical best solutions are determined by the optimization objective of the lower ant colony, this global pheromone update strategy plays the role of information transmission from the lower ant colony to the upper ant colony. The two ant colonies repeat the above process of constructing solutions until the termination criterion is satisfied. At this time, the historical best solutions of the two ant colonies is outputted.

Algorithm 1 DACO-CIPL

```
1: Input: Candidate charging station sites, track data
2: Output:Best charging station siting and best charging pile allocation
3: Initialization:
4: Set parameters $\alpha, \beta, \gamma, \rho, q_0, \mu$, itermax
5: Initialize heuristic information matrix $H_1, H_2$ and pheromone matrix $P_1, P_2$
6: iter=0;
7: while iter<itermax do
8:     for k=1 to $\mu$ do
9:         for j=1 to $N$ do
10:            Upper ant k select a feasible vertex
11:            from search graph $G_1$;
12:            Perform local pheromone update
13:            on this vertex;
14:    Perform pheromone enhancement for
15:    the lower ant colony;
16:    for k=1 to $\mu$ do
17:        for j=1 to $N$ do
18:            Lower ant k select a feasible vertex
19:            from search graph $G_2$;
20:            Perform local pheromone update
21:            on this vertex
22:    Get the best solutions of the two
23:    subproblems in this iteration: $X^{iter}_{best}, Y^{iter}_{best}$
24:    if $F(X^{iter}_{best}, Y^{iter}_{best}) \geq F(X^{h}_{best}, Y^{h}_{best})$ then
25:        $X^{h}_{best}, Y^{h}_{best} = X^{iter}_{best}, Y^{iter}_{best}$;
26:    Perform global pheromone update for
27:    the two ant colonies;
28:    end
29:    iter = iter + 1;
30: end while
31: Return: Best charging station siting $X^{h}_{bes}$, Best charging pile allocation $Y^{h}_{bes}$
```

Algorithm 1 provides the pseudocode of DACO-CIPL.

Lines 3–6 are the initialization of the algorithm, where two matrix initializations are described in Section 4.4. Lines 10–11 and lines 18–19 represent the solution construction process of the upper and lower ant colonies, respectively, which are explained in Section 4.5. Lines 12–13 and lines 20–21 perform local pheromone updates, while lines 26–27 perform global pheromone updates, which will be discussed in Section 4.6. Lines 14–15 are the pheromone reinforcement strategy, which is introduced in Section 4.7. Lines 22–25 determine the current iteration's best solution based on the composite objective function F and update the historical best solution.

4.2 Encoding scheme of ants

In the upper ant colony, each ant is represented as a candidate solution of the charging station siting problem in the form of an N-permutation of the set $\{1, 2\ldots, n\}$, where N is the number of candidate charging station sites and n is the number of charging stations to be constructed. In the lower ant colony, each ant is represented as a candidate solution of the charging pile allocation problem in the form of an N-permutation with repetition of the set $\{1, 2, \ldots, r_{max}\}$, where r_{max} is the maximal number of charging piles allocated for a charging station. According to the description in Section 2, each ant of the upper ant colony is denoted by $X=(x_1, x_2, \ldots, x_j, \ldots, x_N)$. If $x_j=i$, it indicates that candidate charging station site i is to construct charging station j. Each ant of the lower ant colony is denoted by $Y=(y_1, y_2, \ldots, y_j, \ldots, y_N)$. If $y_j=r$, the number of charging piles is r at charging station j.

4.3 Search graphs of two ant colonies

In DACO-CIPL, the search space of each ant colony is a search graph. The search graphs of two ant colonies are denoted by $G_1=(V_1, E_1)$ and $G_2=(V_2, E_2)$, respectively,where $V_i(i=1, 2)$ are two sets of vertices. Each vertex of V_1 represents a possible assignment of a charging station being constructed on a certain candidate charging station site. Each vertex of V_2 represents a possible number of charging piles being allocated for a certain candidate charging station site if it is selected to construct a charging station. $E_i(i=1, 2)$ are two sets of edges connecting the vertices in V_i. Figures 3 and 4 illustrate the search graphs of the two ant colonies. The G_1 vertex is shown in Fig. 3, and there are $n \times N$ vertices, where any vertex v_{ij} represents the charging station j constructed on the candidate charging station site i. In G_2 shown in Fig. 4, there are $r_{max} \times n$ vertices, where any vertex w_{ri} represents the number of charging piles allocated in candidate charging station site i is r.

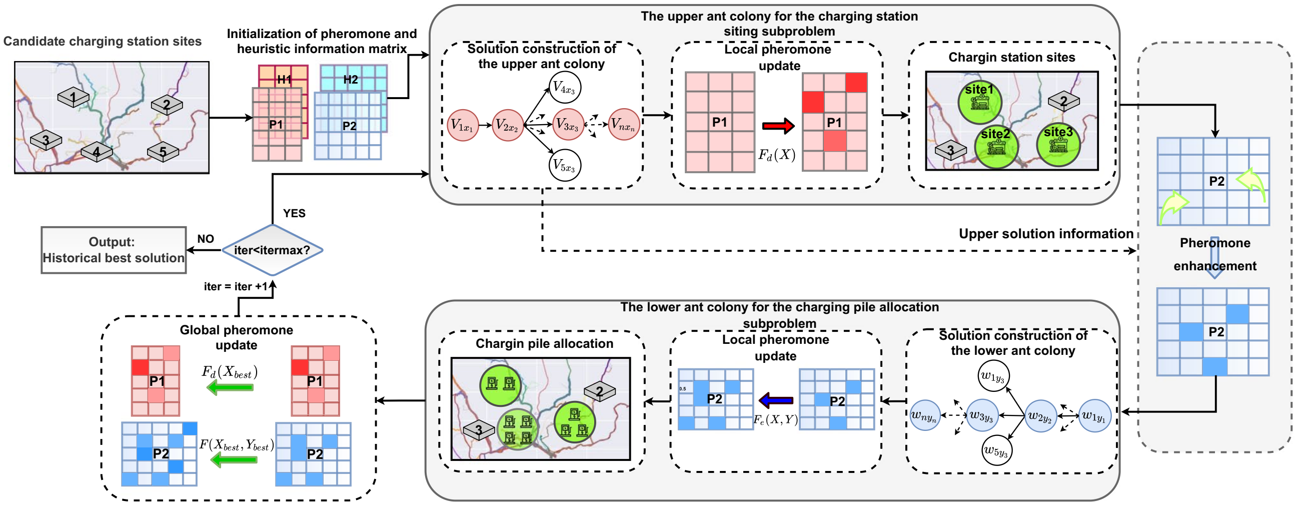

Fig. 2 The overall framework of DACO-CIPL

4.4 Initialization of heuristic information and pheromone

Corresponding to the definition of the search graphs G1 and G2, DACO-CIPL needs to initialize two heuristic information matrices, H_1, H_2, and two pheromone matrices, P_1 and P_2. H_1 and P_1 are $n \times N$ matrices related to the upper ant colony. $P_1(i, j)$ and $H_1(i, j)$ denote the pheromone and heuristic values on vertex v_{ij} of G_1, respectively. H_2 and P_2 are $r_{max} \times n$ matrices related to the lower ant colony. $P_2(r, i)$ and $H_2(r, i)$ denote the pheromone and heuristic values on vertex v_{ri} of G_2, respectively. The heuristic information matrix contains the prior knowledge related to the problem and can help the ant colony to effectively explore the search space. The two heuristic information matrices are not updated while the two ant colonies are searching. The pheromone matrix records the search experience of the ant colony and can guide the ant colony to explore the search space in the right direction. The two pheromone matrices are updated in each iteration since they play roles in the information transmission of the two ant colonies.

The initialization of H_1 is based on the number of tracks covered by the candidate charging station sites. The heuristic value $H_1(i, j)$ on any vertex v_{ij} is calculated as:

$$H_1(i,j) = \sum_{t=1}^{m} s(i,t) \tag{4.1}$$

where m is the total number of tracks and $s(i,t)$ is a function to indicate whether track t is covered by candidate charging station site i. H_1 is designed to increase the probability of constructing charging stations that cover more tracks.

The initialization of H_2 is based on the operation cost of the candidate charging station site when it is selected to construct the charging station. The heuristic value $H_2(r,i)$ on any vertex Vri is calculated as:

$$H_2(r,i) = \frac{1}{\omega(r,i) \cdot c \cdot r} \tag{4.2}$$

where c is the construction cost of a charging pile; r is the number of charging piles; and $\boldsymbol{\omega(r,i)}$ is the queuing probability of candidate charging station site i shown in Eq. (3.6). The heuristic information estimates the inverse of the operation cost of a specific charging pile allocation for a candidate charging station site i. It is designed to increase the probability of selecting the charging pile allocation scheme that has a low operation cost.

The initialization of the pheromone matrices P_1 and P_2 is based on the result of a greedy algorithm, which provides early search guidance to the ant colony. Suppose the solu- tions of the greedy algorithm for the charging station sit- ing problem and the charging pile allocation problem are denoted by X_{gd} and Y_{gd}, respectively. All the entries of the matrix P_1 are initialized to the same value τ_1:

$$\tau_1 = F_d\left(X_{gd}\right)/N \tag{4.3}$$

where $F_d(X_{gd})$ is the objective value of the charging station siting problem shown in Eq. (3.4); N is the number of charging stations to be constructed.

All the entries of the matrix P_2 are initialized to the same value τ_2:

$$\tau_2 = F\left(X_{gd}, Y_{gd}\right)/N \tag{4.4}$$

where F (Xgd, Ygd) is the objective value of the charging pile allocation problem shown in Eq. (3.12). It is designed to en- courage the lower ant colony to allocate more charging piles for charging stations covering high charging demand.

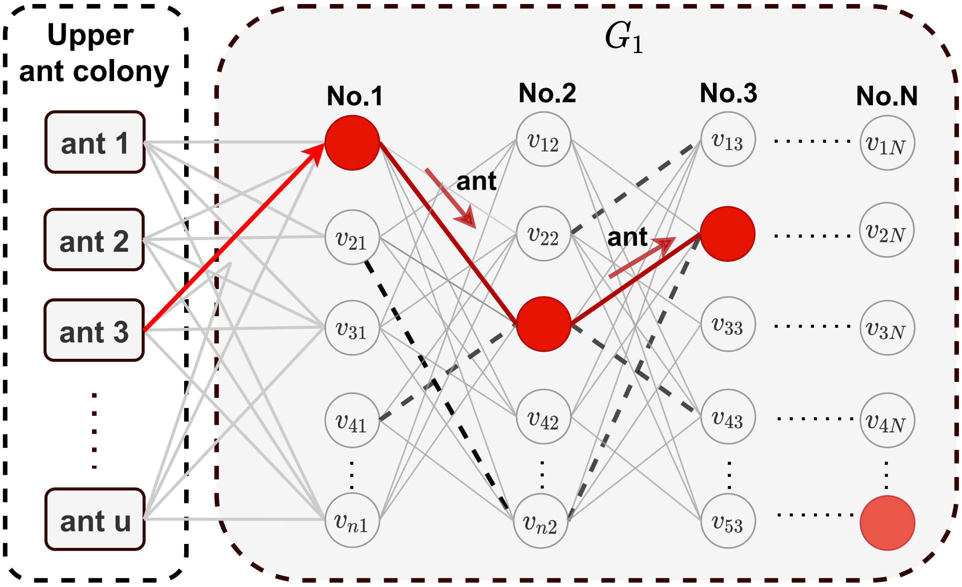

Fig. 3 The search graph G1 of the upper ant colony

4.5 Solution construction

DACO-CIPL employs two ant colonies to solve the charging station siting problem and the charging pile allocation problem of CIPL. Each ant colony contains u ants. The upper ant colony constructs the solution of the charging station siting problem by walking on the graph G1 based on the heuristic matrix H_1 and the pheromone matrix P_1. The lower ant colony constructs the solution of the charging pile allocation problem by walking on the graph G2 based on the heuristic matrix H_2 and the pheromone matrix P_2.

Each ant in the two colonies constructs a solution from an empty path list. At each step, the path list of each ant is expanded by adding a feasible vertex that satisfies the constraints (3.13–3.18). In the upper ant colony, suppose that an ant is walking on the graph G1 and the last vertex it just visited is $v_{x_{j-1}}j-1$. At this time, the trajectory of this ant can be denoted by the path list $V_X = \left(x_{x_1}1, v_{x_2}2, \cdots, v_{x_{j-1}j-1}\right)$. According to the constraints defined in Eq. (3.5) and Eq. (3.6), the next vertex can be selected from the set $V_j = \left\{v_{ij} | i \in \{1, 2, \cdots, n\} \backslash \{x_1, x_2, \cdots, x_{j-1}\}\right\}$, which is the set of the re- maining available candidate charging station sites. The state transition rule to the next vertex is as follows:

$$v = \begin{cases} \text{argmax}_{v_{i,j} \in V_j} \left\{P_1(i,j)^{\alpha} \cdot H_1(i,j)^{\beta}\right\}, & q \geq q_0 \\ \text{Random}\left(V_j\right), & \text{otherwise} \end{cases} \tag{4.5}$$

where α is the weight for the pheromone trail, β is the weight for the heuristic information, q is a random number in the interval [0, 1], and $q_0 \in (0, 1)$ is an initial

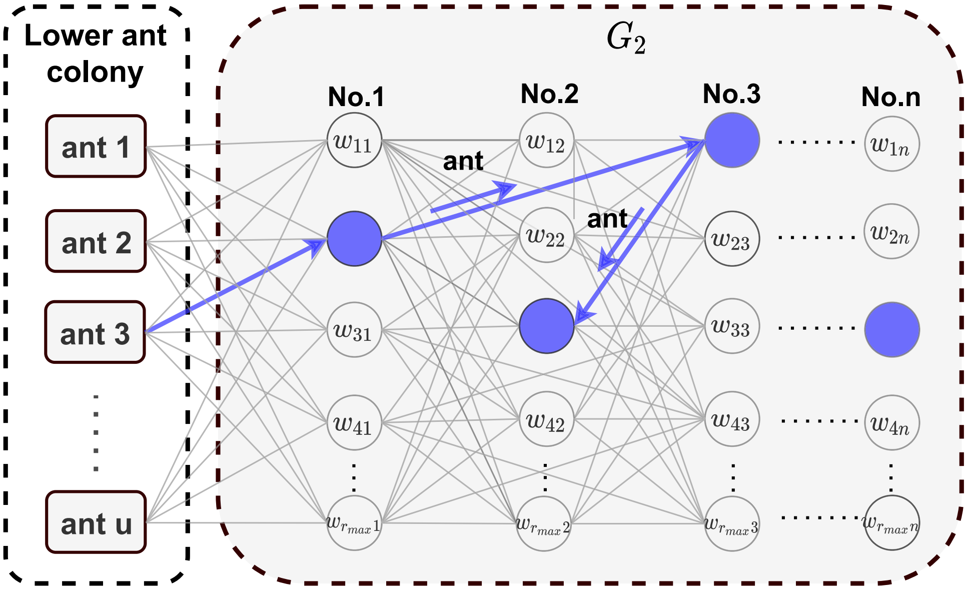

Fig. 4 The search graph G2 of the lower ant colony

parameter that determines the relative importance of exploitation versus exploration. If $q \geq q_0$, the ant will select the vertex with the maximum product value of the pheromone and the heuristic information, namely, exploitation. Otherwise, the ant will randomly select the vertex according to a certain probability, namely, exploration. The probability of selecting any vertex v_{ij} is based on the roulette wheel rule:

$$p_{ij} = \begin{cases} \frac{P_1(i,j)^{\alpha} \cdot H_1(i,j)^{\beta}}{\sum_{v_{kj} \in V_j} P_1(k,j)^{\alpha} \cdot H_1(k,j)^{\beta}}, & v_{ij} \in V_j \\ 0, & \text{otherwise} \end{cases} \tag{4.6}$$

As the lower ant colony needs to construct solutions based on the solution of the upper ant colony, after reaching vertex $w_{y_{j-1}} x_{j-1}$, it selects the next vertices from set $W = \left\{ w_{rx_j} | r \in 1, 2, \cdots, r_{max} \right\}$ and moves according to formula (4.5) based on matrices P_2 and H_2.

In this way, the ants tend to select the vertices with higher pheromone values in the search process. By repeating the transition rule, each ant will obtain a path list from one end of the search graph to the other. Each path list forms the so- lution of the corresponding ant. For example, in Fig. 3, the upper ant 3 constructs the path list $V_X = \left(v_{11}, v_{32}, v_{23}, \cdots, v_{nx_N} \right)$, i.e., the solution of the charging station siting problem is (1, 3, 2, ..., n). Further, in Fig. 4, the lower ant 3 constructs the path list $V_Y = \left(w_{21}, w_{13}, w_{32}, \cdots, w_{3n} \right)$, i.e., the solution of the charging pile allocation problem is (2, 1, 3, ..., 3). The two solutions together constitute an option for CIPL.

4.6 Pheromone update

Pheromones act as a means of information communication between two ant colonies. In each iteration of the solution construction process, two pheromone matrices go through two kinds of updates: local pheromone and global pheromone.

1) Local pheromone update: After an ant visits a vertex, this ant performs a local pheromone update on the pheromone value related to this vertex just visited.

In the upper ant colony, suppose the vertex v_{ij} of G_1 is just visited by an upper ant, and the pheromone value $P_1(i, j)$ related to v_{ij} is updated according to:

$$P_1(i,j) = (1 - \rho) \cdot P_1(i,j) + \rho \cdot \tau_1 \tag{4.7}$$

where ρ is the control of pheromone evaporation and τ_1 is the pheromone value of the upper ant colony.

In the lower ant colony, suppose the vertex w_{ri} of G_2 is just visited by a lower ant. The pheromone value P (r, i) related to w_{ri} is updated according to:

$$P_2(r,i) = (1 - \rho) \cdot P_2(r,i) + \rho \cdot \tau_2 \tag{4.8}$$

where τ_2 is the pheromone value of the lower ant colony.

The local pheromone update can diversify the ant colony search by evaporating pheromones related to the vertices previously selected. This strategy can help the ant colony find new solutions and avoid falling into local optimal solutions too quickly.

2) Global pheromone update: At the end of each iteration, the global pheromone update strategy will be performed based on the historical best solutions. The global pheromone update rules of the two ant colonies are as follows:

$$P_1(i,j) = (1 - \rho) \cdot P_1(i,j) + \rho \cdot F_d\left(X^h_{best}\right), v_{ij} \in V_{X^h_{best}} \tag{4.9}$$

$$P_2(r,i) = (1 - \rho) \cdot P_2(i,j) + \rho \cdot F\left(X^h_{best}, Y^h_{best}\right), v_{ri} \in V_{Y^h_{best}} \tag{4.10}$$

where $Y^h{}_{best}$ and $X^h{}_{best}$ are the historical best solutions obtained by considering the objective function of the lower ant colony. Therefore, this global pheromone update realizes information transmission from the lower ant colony to the upper ant colony. It is helpful to find a solution of the upper ant colony that is beneficial to two optimization objectives.

4.7 Pheromone enhancement of the lower ant colony

The lower search graph G2 consists of n columns of vertices, representing n candidate sites for charging stations. Due to constraints imposed by the upper decision variables, lower ants can only construct solutions on N columns, where $N < < n$. Therefore, only a very small number of nodes in G_2 are selected and updated with each iteration of pheromones, leading to small differences in pheromone concentrations in matrix P^2 and poorer solutions. To improve the solutions generated by the lower ant colony, a pheromone enhancement strategy is proposed to transfer the solution information of the upper ant colony to enhance the differences in pheromone concentrations in P_2.

After the upper ant colony finishes its solution construction process, any upper ant k will transmit a solution $X^k = (x^k{}_1, x^k{}_2, \ldots, x^k{}_j, \ldots, x^k{}_N)$ to the lower ant k. Each entry x^k has a set of corresponding vertices $\left\{ w_{1x_j}, w_{2x_j}, \cdots w_{rx_j}, \cdots, w_{r_{max}x_j} \right\}$ in the search graph G2. The pheromone enhancement strategy will enhance the pheromone of each vertex in the set. Taking w_{rx_j} as an example, the specific steps are as follows.

First, DACO-CIPL finds the similar vertices of w_{rx_j} according to the similarity rule:

$$same\left(w_{rx_j}, w_{ri}\right) = \begin{cases} 1, & \left|1 - \frac{\sum_{t\in T} s(i,t)}{\sum_{t\in T} s\left(x_j,t\right)}\right| \le \gamma \\ 0, & \text{otherwise} \end{cases} \tag{4.11}$$

where $s(x_j, t)$ is a function to indicate whether track t is covered by candidate charging station site $x\,j$ and γ is the threshold of vertex similarity. When the difference degree of the track number is within γ, w_{rx_j} and w_{ri} are judged to be similar.

Then, all the pheromone values of similar vertices of w_{rx_j} are added up to the pheromone value $P_2(r, x_j)$ based on the following formula:

$$P_2\left(r, x_j\right) = P_2\left(r, x_j\right) + \sum_{i=1}^{n} same\left(w_{rx_j}, w_{ri}\right) P_2(r, i) \tag{4.12}$$

The pheromone enhancement strategy can strengthen the pheromone values on some vertices of search space G_2 to increase the difference in pheromone concentration. It allows the lower ant colony to avoid inexperienced search and accelerate its convergence. This strategy is helpful for the lower ant colony to determine a better solution for charging pile allocation while satisfying the charging demand.

4.8 Time complexity analysis

DACO-CIPL must initialize four matrices, namely, H_1, H_2, P_1, and P_2, and has a time complexity of $O(N \times n)$. Furthermore, DACO-CIPL has a total of *itermax* iterations. In each iteration, the upper ant colony searches in G_1, and the lower ant colony searches in $G2$. Each search phase involves μ ants of each colony, and each ant is required to select N nodes. Consequently, the time complexity of the iteration phase is $O(itermax \times \mu \times N)$. In summary, the algorithm's time complexity is $O(itermax \times \mu \times N)$. Here, N denotes the number of charging stations, μ represents the number of ants in each colony, and *itermax* indicates the iteration number.

5 Experimental study

This section will describe experiments performed to investigate the performance of the proposed DACO-CIPL algorithm. We first introduce the experimental setup. Then, a parameter sensitivity analysis is introduced to give proper parameter settings for DACO-CIPL. Finally, the effectiveness of DACO-CIPL is validated by comparing DACO-CIPL with several competitive algorithms on multiple test instances generated from a real taxi dataset. All algorithms are implemented in Python and the PyCharm compiler. The experimental platform is a PC with an Intel Core i7-4770 CPU running at 2.40 GHz and 24 GB of RAM, running on Windows 7.

5.1 Experimental setup

A) Original dataset: This paper employs the taxi track dataset in Washington, DC, USA (link:https://tianchi.aliyun.com/dataset/dataDetail?dataId=94221). This dataset includes one hun- dred thousand complete taxicab tracks for calendar year 2017. Each track includes information about the starting and the ending coordinates.

B) Test datasets: Three different scales (SC) of test datasets are used: a small-scale dataset (SC1), medium-scale dataset (SC2), and large-scale dataset (SC3). Table 1 provides de- tailed information on the three different scales of the test datasets, where m, N, n and r_{max} represent the number of tracks randomly selected from the original dataset, the number of candidate charging station sites, the number of charging stations to be constructed and the maximal number of charging piles of each charging station, respectively. In the experiment, five test instances are generated for each scale of the dataset by randomly selecting m tracks. Thus, there are 15 test instances used in the experiments. These test instances are denoted by SC1-i ($i=1, 2\ldots, 5$), SC2-i ($i=1, 2\ldots, 5$) and SC3-i ($i=1, 2\ldots, 5$).

C) Algorithm comparison: To validate the effectiveness of DACO-CIPL, this paper compares it with three other meth- ods: GA-CIPL [20], PSO-CIPL [25], and AHA algorithm [23]. The first two algorithms only separately use GA and PSO to solve the charging station siting problem. Because the charging pile allocation problem is also a combinatorial optimization problem similar to the charging station siting problem, they can be used to solve the charging pile allocation problem via parameter adjustment. Therefore, this pa- per duplicates their optimization process to determine the charging pile allocation. The AHA has a specific dual-layer solving process to address CIPL. It first determines the charging station sites to maximize charging demand and then adjusts some sites of the charging station and allocates charging piles to minimize economic costs.

5.2 Parameter sensitivity analysis

In this section, we study the effects of different parameters on the performance of DACO-CIPL and select proper settings for them. The parameter settings of the two ant

Table 1 Three scales of test datasets

Scale	m	n	N	*rmax*
SC1	2500	100	5	5
SC2	5000	200	10	10
SC3	10,000	400	20	20

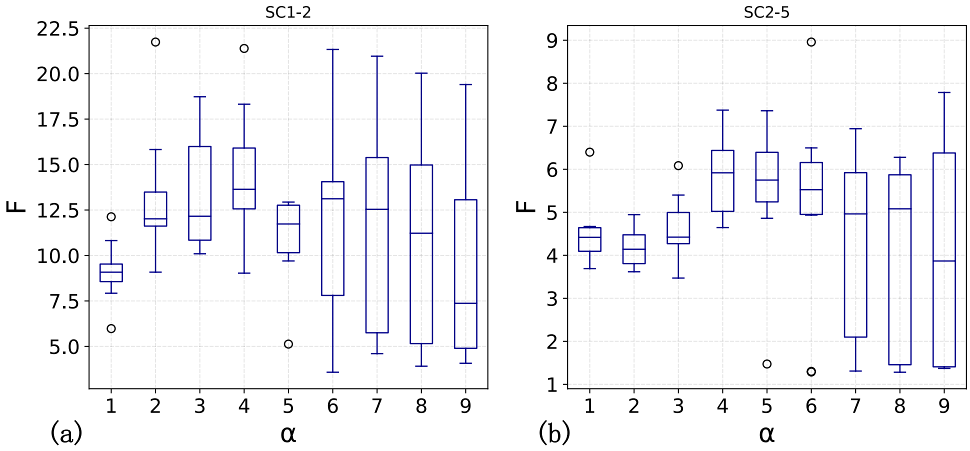

Fig. 5 The performance of DACO-CIPL under different values of α: (**a**) Results on test instance SC1-2, and (**b**) Results on test instance SC2-5

colonies are the same. These parameters include weights for the pheromone trail (α) and for the heuristic information (β), the relative importance of the exploitation versus exploration (q_0), the control of pheromone evaporation (ρ) and the threshold of vertex similarity (γ). Two test instances, SC1-2 and SC2-5, are used in this part of the experiment.

This paper employs the control variate technique in which the value of a single parameter is changed while the values of other parameters remain fixed. We first set the default values of α, β, q_0, ρ and γ to 2, 3, 0.9, 0.3 and 0.7 and test the performance of DACO-CIPL under different candidate values of one parameter according to the objective value F of CIPL. This paper shows the results in box plots, where the top and the bottom of each box indicate the 75th and 25th percentiles, respectively; the line in each box indicates the 50th percentile; the whisker bars below and above each box indicate the minimum value and the maximum value, respectively; and the circle indicates the outlier.

A) Weights for the pheromone trail (α): The parameter α determines the importance of pheromones. To study its effects, this paper tested different values $\{1, 2, 3\ldots, 9\}$ of α. Figure 5 shows the box plots of α on SC1-2 and SC2-5. When $\alpha = 4$, the median value of F is the best on two test in- stances. A larger or smaller value would impair the algorithm performance. This is because when $\alpha < 4$, the ant colony has difficulty learning the search experience. When $\alpha > 4$, the ant colony heavily depends on the search experience, which would reduce the ability to explore new solutions.

B) Weights for heuristic information (β): The parameter β determines the importance of heuristic information. To study its effects, this paper tested different values $\{1, 2, 3\ldots, 9\}$ of β. Figure 6 shows the box plots of β on SC1-2 and SC2-5. When $\beta = 2$, the median value of F is the best on two test instances. A larger value would weaken the effect of pheromone and lead to a worse performance since the attention of the ant colony is largely guided by heuristic information. A smaller value will lead to unstable performance of the algorithm since the small weight of heuristic information may cause the ant colony to search blindly in the early iteration.

C) Relative importance of exploitation versus exploration (q_0): The parameter $q0$ $(0 < q0 < 1)$ controls the probability of exploration. If $q \leq q_0$, the ants randomly search

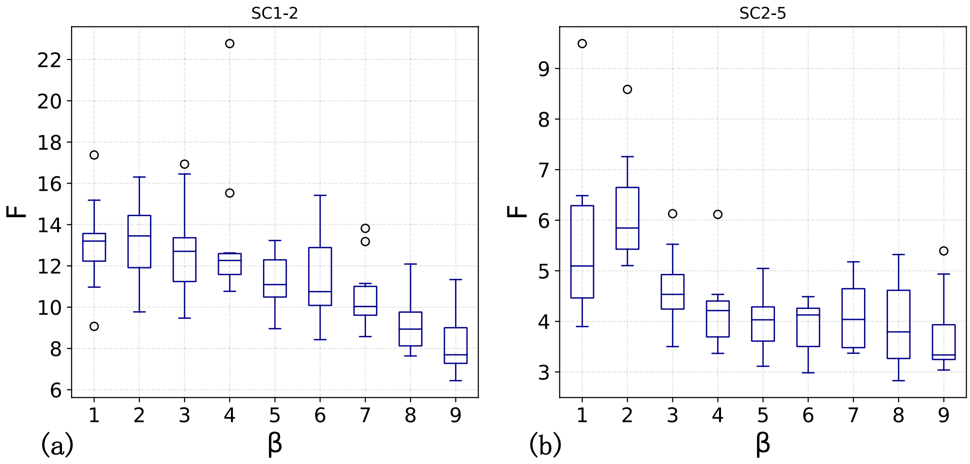

Fig. 6 The performance of DACO-CIPL under different val- ues of β: (**a**) Results on test instance SC1-2, and (**b**) Results on test instance SC2-5

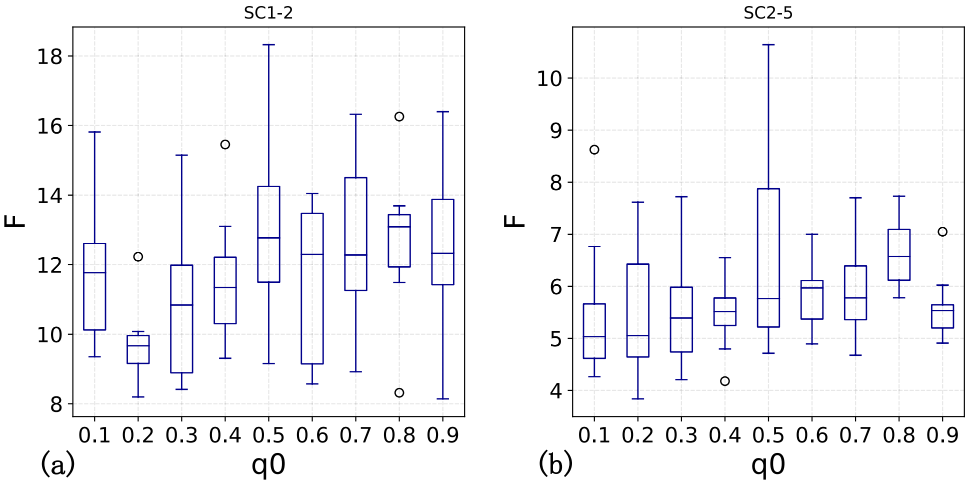

Fig. 7 The performance of DACO-CIPL under different val- ues of $q0$: (**a**) Results on test instance SC1-2, and (**b**) Results on test instance SC2-5

for the next vertex based on the probability. If $q \geq q0$, the ants search for the vertex that has the maximum product of pheromone and heuristic information. To study its effect, we tested several different values *{*0.1*,* 0.2*,* …*,* 0.9*}* of $q0$. Figure 7 shows the box plots of $q0$ on SC1-2 and SC2-5. When $q0=0.8$, the median and maximum values of F are the best on the two test instances. A smaller value would make the ant colony gradually reduce its exploration ability and easily fall into the local optimal solution.

D) Control of pheromone evaporation (ρ): The parameter ρ represents the evaporation rate of pheromones. Figure 8 shows the box plots of ρ on SC1-2 and SC2-5 under different values {0.1, 0.2, …, 0.9}. When $\rho=0.6$, the median value of F is the best on two test instances. A larger value will slow the convergence of the ant colony due to the rapid evaporation of pheromones. A smaller value will cause the ant colony to rapidly converge to the local optimal solution due to excessive accumulation of pheromones.

E) Threshold of vertex similarity (γ): The parameter γ determines the similarity of vertices in the pheromone enhancement strategy. Figure 9 shows the box plots of γ on SC1-2 and SC2-5 under different values *{*0.0*,* 0.1*,* …*,* 0.9*}*. According to the experimental results, when $0.5 \leq \gamma \leq 0.8$, the performance of DACO-CIPL is better than that when $\gamma < 0.5$ on two test instances. Therefore, it can be concluded that a smaller value would reduce the effectiveness of the pheromone enhancement strategy. This is because the enhanced pheromone has no guiding significance when the vertex similarity is too low. When $\gamma=0.8$, F has the largest median value. When $\gamma=0.9$, the high similarity makes it difficult for DACO-CIPL to find similar vertices, resulting in worse performance.

After the above experimental tests, we find that the set of parameters $\alpha = 4$, $\beta = 2$, $\rho = 0.6$, $q_0 = 0.9$, $\gamma = 0.8$, DACO-CIPL performed best. In addition, we conducted parameter sensitivity analysis for PSO-CIPL, GA-CIPL and AHA to determine suitable parameter values. The details are omitted due to space limitations. For PSO-CIPL, the learning factor $c1$ is 2.8 and c_2 is 1.3, and the particle swarm size is 50. For

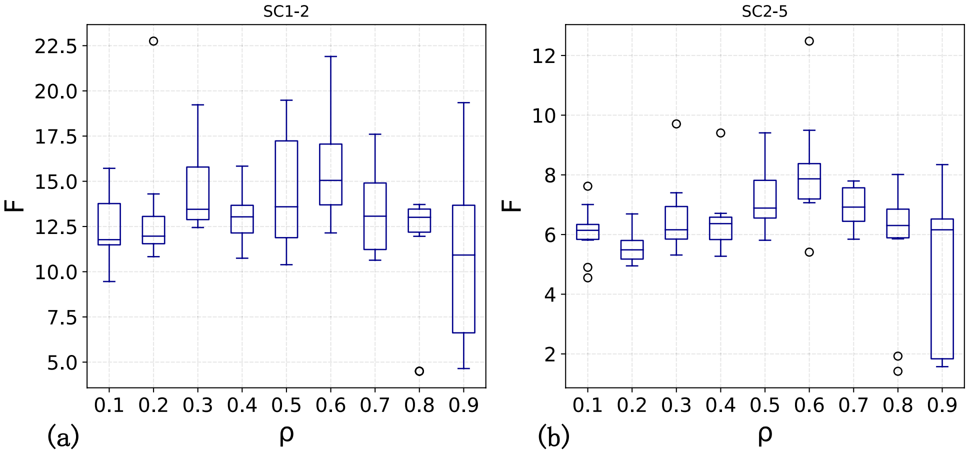

Fig. 8 The performance of DACO-CIPL under different values of ρ: (**a**) Results on test instance SC1-2, and (**b**) Results on test instance SC2-5

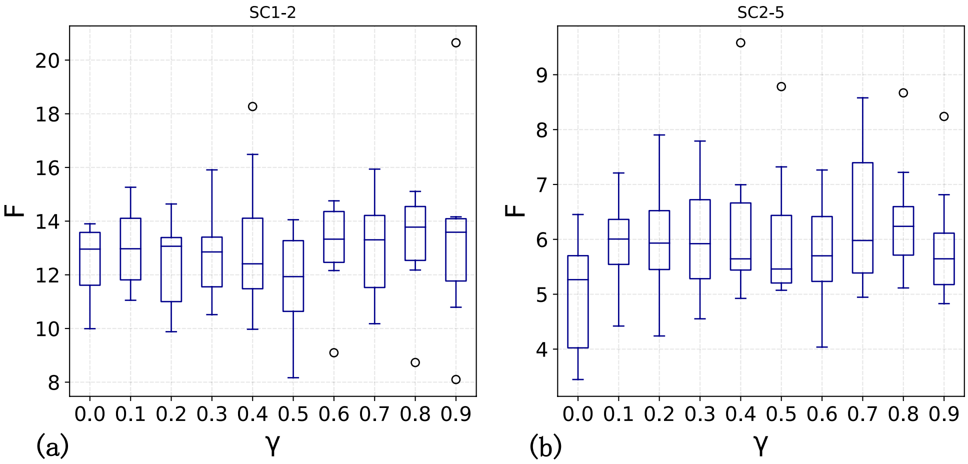

Fig. 9 The performance of DACO-CIPL under different val- ues of γ: (**a**) Results on test instance SC1-2, and (**b**) Results on test instance SC2-5

GA-CIPL, the population size is 50, the crossover probability *pc* is 0.7, and the mutation probability *pm* is 0.2. For AHA, the number of siting iterations *siter* is 20. To ensure the fairness of comparison, the ant number of DACO-CIPL is also set to 50, and the maximum iteration count of GA- CIPL and PSO-CIPL is set to 20, which is the same as that of DACO-CIPL.

5.3 Overall performance

To validate the performance of DACO-CIPL, this section compares DACO-CIPL with AHA, GA-CIPL, and PSO-CIPL on all 15 test instances.

Table 2 provides the objective value of the two best subproblem solutions over 60 runs. In this table, $Best_F$ is the objective value of charging pile allocation. As $Best_F$ merges the objective of the charging station siting problem, it can be used as a comprehensive metric. $Best_{F_d}$ is the corresponding objective value of the best solution for the charging station siting problem. The best result in each test instance is shown in bold. From this table, the following two points can be obtained.

(1) From the best objective value of CIPL ($Best_F$), DACO- CIPL achieved the optimal solution in 6 of 10 test cases in small-scale (SC1) and medium-scale (SC2) test instances. In the large-scale test instance (SC3), DACO-CIPL outperformed all other algorithms examined. These results indicate that the performance of DACO-CIPL is better than that of the other algorithms examined. This may be because the pheromone enhancement strategy provides the lower ant colony with the search experience from previous iterations to obtain better solutions of charging pile allocation, while other algorithms only rely on the experience of a single iteration to solve the charging

Table 2 Comparison of the best objective function values on all the test instances

Test instances	DACO-CIPL		AHA		GA-CIPL		PSO-CIPL	
	$Best_{F_d}$	$Best_F$	$Best_{F_d}$	$Best_F$	$Best_{F_d}$	$Best_F$	$Best_{F_d}$	$Best_F$
SC1-1	**94.74**	**22.95**	93.53	18.53	94.42	21.81	79.41	13.37
SC1-2	84.22	16.55	79.29	13.4	73.7	**17.83**	**95.32**	8.54
SC1-3	93.2	16.16	**111.27**	15.22	90.65	**21.01**	83.04	9.03
SC1-4	94.42	12.53	85.65	12.34	**98.96**	**12.57**	76.75	8.8
SC1-5	99.43	**33.83**	101.57	30.58	**102.82**	32.39	97.56	13.55
SC2-1	**400.28**	**3.71**	287.55	2.53	292.71	3.47	368.92	2.66
SC2-2	**330.17**	**3.14**	280.94	2.92	237.44	1.9	257.77	5.41
SC2-3	**393.19**	**7.18**	358.31	2.65	271.5	1.97	294.1	1.67
SC2-4	257.83	8.72	**265.83**	**9.78**	242.57	6.73	240.74	5.16
SC2-5	**250.99**	**9.34**	236.2	6.92	210.29	6.19	210.39	6.84
SC3-1	**585.09**	**3.74**	535.85	2.27	429.47	2.55	462.73	1.93
SC3-2	**657.23**	**6.29**	503.9	2.12	511.71	2.63	423.23	1.97
SC3-3	**632.33**	**4.43**	451.75	1.95	428.45	1.87	445.24	2.54
SC3-4	**602.27**	**4.24**	492.76	2.57	437.72	2.26	519.89	3.06
SC3-5	**624.12**	**3.88**	559.01	3.17	365.59	2.47	450.57	1.77

Table 3 Comparison of mean performance and time performance on all the test instances

Test instances	DACO-CIPL			AHA			GA-CIPL			PSO-CIPL		
	$MEAN_F$	STD_F	time (s)	$MEAN_F$	STD_F	time (s)	$MEAN_F$	STD_F	time (s)	$MEAN_F$	STD_F	time (s)
SC1-1	13.73	3.36	**3.87**	13.74	3.03	9.49	**15.9**	6.63	5.24	9.43	**2.18**	4.34
SC1-2	**13.77**	1.71	**4.03**	13.03	8.9	9.91	12.11	3.39	5.15	7.02	**1.32**	5.77
SC1-3	10.69	2.13	**4.11**	**16.5**	8.04	17.71	15.85	3.71	5.18	6.04	**1.68**	5.68
SC1-4	**10.49**	**0.77**	**3.88**	9.39	3.26	12.46	10.42	2.44	4.15	7.18	0.88	5.22
SC1-5	19.44	2.31	**3.7**	**21.84**	15.07	11.42	18.01	7.09	8.57	8.94	**1.8**	4.39
SC2-1	**2.63**	0.43	**11.84**	1.86	0.55	187.22	1.53	0.77	17.62	1.91	**0.39**	4.39
SC2-2	**2.65**	**0.25**	**11.85**	1.86	0.64	199.36	1.76	0.75	17.34	1.96	1.22	33.8
SC2-3	**6.69**	0.33	**11.92**	1.33	0.66	231.18	1.44	0.25	17.91	1.27	**0.19**	31.97
SC2-4	**6.28**	1.16	**9.96**	6.21	2.86	117.77	4.87	**0.82**	15.35	3.95	0.86	20.34
SC2-5	**6.64**	1.14	**10.03**	5.57	1.27	169.35	4.66	**1.12**	15.77	4.1	1.26	15.54
SC3-1	**3.11**	0.35	**25.75**	1.65	0.44	1649.2	1.67	0.34	41.37	1.46	**0.18**	122.34
SC3-2	**4.61**	0.72	**24.74**	1.62	0.36	1685.41	1.94	0.42	39.69	1.64	**0.19**	112.06
SC3-3	**3.76**	0.36	**25.82**	1.4	0.34	1930.15	1.66	**0.15**	42.18	1.08	0.56	111.97
SC3-4	**3.42**	0.36	**25.55**	1.82	0.62	1827.785	1.78	**0.28**	40.93	1.44	0.59	124.65
SC3-5	**3.1**	0.29	**26.45**	1.91	0.77	1868.2	1.93	0.25	42.02	1.46	**0.12**	130.11

pile allocation subproblem. Therefore, DACO-CIPL can obtain a better comprehensive objective function F value.

(2) From the best objective value of charging station siting ($Best_{F_d}$): If an algorithm still achieves the best performance metric when it obtains the best objective value of the charging pile allocation, this algorithm has a better ability to comprehensively consider the two subproblems of CIPL. As shown in Table 2, DACO-CIPL obtains the best results on 11 test instances according to the value of $Best_F$. Among all 11 test instances, DACO-CIPL obtains the best results on 10 test instances in terms of $Best_{F_d}$. AHA and GA-CIPL only simultaneously obtain the best results of the two subproblems on one test instance. PSO-CIPL fails to do this. Thus, DACO-CIPL has superior ability to systematically solve the two subproblems of CIPL.

A comparison of mean performance and time performance is shown in Table 3, where"*ME ANF*" denotes the average value of *F* over 60 runs,"*ST DF*" denotes the standard deviation of *F* and"time" denotes the average CPU time. These metrics can reflect the stability and computational cost of each algorithm. The best result in each test instance is shown in bold. From the average value of *F* (*ME ANF*), DACO-CIPL performs best on 12 of 15 test instances, which proves that DACO-CIPL is more stable. Concerning the average CPU time, DACO-CIPL has less CPU time than the other algorithms examined on all test instances. The results indicate that DACO-CIPL is able to achieve better solutions in less time.

To clearly demonstrate the differences between algorithms from a statistical perspective, the Friedman test was con- ducted based on the $Best_F$ values of 15 test instances. Figure 10 presents the overall ranking of the four algorithms according to $Best_F$ values and their corresponding p values from the Friedman test. The results of the Friedman test indicate that the p value is much smaller than 0.05, suggesting a significant difference in the performance of the four algorithms. The x-axis represents the ranking of the algorithms, and the shorter the length of the bar chart is, the higher the performance ranking. From this perspective, DACO-CIPL ranks first, with a significant advantage, followed by AHA, GA- CIPL, and IPSO-CIPL, in descending order. These results indicate that DACO-CIPL has strong comprehensive solving ability for the CIPL problem. Additionally, AHA performs slightly better than GA-CIPL, but its

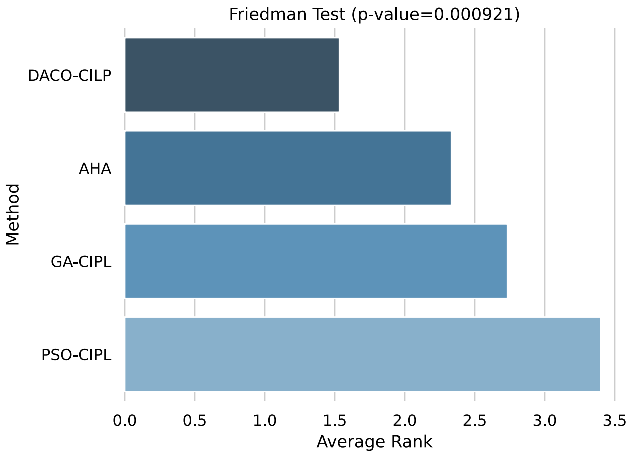

Fig. 10 The results of the Friedman test

average CPU time is too high for large-scale test instances. Moreover, the stability of GA-CIPL is much higher than that of AHA. Therefore, GA-CIPL may have more potential in solving the CIPL problem than AHA.

5.4 Comparison of convergence

To investigate the search efficiencies of different algorithms, we compared the convergence curves of ACO-CIPL, GA-CIPL, PSO-CIPL and DACO-CIPL-noE (DACO-CIPL without pheromone enhancement) on SC2-5, as shown in Fig. 11. First, compared to DACO-CIPL-noE, DACO-CIPL converges faster to a better global solution. This result proves that the pheromone enhancement strategy can effectively accelerate convergence and guide a search. Second, it can be easily found that DACO-CIPL has a better initial solution. This is due to the initial value of pheromones provided by the results of the greedy algorithm, which improves the quality of the initial solution. Finally, compared to GA-CIPL and PSO-CIPL, DACO-CIPL always has the fastest convergence speed. This should give credence to the information transmission mechanism between two ant colonies.

5.5 Visualization of CIPL

To demonstrate the effectiveness of the proposed algorithms in practical applications, this paper presents visualizations of the optimal results of DACO-CIPL and GA-CIPL on test instances SC1-5 and SC2-5. GA-CIPL was selected as the comparison algorithm due to its superior optimization ability and stability compared to other algorithms. In Fig. 12, the charging station sites selected by DACO-CIPL are marked in green, those selected by GA-CIPL are marked in red, and those selected by both algorithms are marked in yellow. Bold font highlights iconic regions near charging stations such as universities, airports, train stations, parks, and neighborhoods on the map. The number of tracks within a 2-km radius of these regions was determined and is summarized in Table 4. To avoid duplicate counting of traffic volume between nearby regions, only the region with the highest track count is considered when there are over 70% of the same tracks between nearby regions. The top 60% of regions ranked by track count in the instances are defined as bustling regions and highlighted in bold in Table 4. The rationality of the result is analyzed based on the sites around these bustling regions.

In Fig. 12(a), it can be observed that GA-CIPL chose 3 bustling regions to build charging stations, namely, Georgetown University, Washington Apartments, and the common selection of Northwest Dicait Street. DACO-CIPL chose to build charging stations at the National Zoological Park and Reagan National Airport, but the other charging stations were slightly off from Washington Apartments, resulting in a lower value of $Best_{F_d}$ than GA-ClPL. As shown in Fig. 12(b), DACOClPL selected 4 bustling regions, namely, DuPont Circle, Washington Union Station, Reagan Washington National Airport, and Smithsonian National Zoological Park, while GAClPL only chose 3 bustling regions, namely, Holiday Inn Express, Washingtoon Union Station, and Georgetown University, resulting in a lower value of $Best_{F_d}$ than DACOClPL.

The number of charging piles in the figures reveals that GA-ClPL tends to allocate charging piles evenly among the selected regions, whereas DACO-ClPL favors allocating more charging piles to regions with higher traffic flow. This is because DACO-ClPL's pheromone enhancement strategy can learn from the charging pile allocation experience of the previous iterations, resulting in a better allocation. On the other hand, GA-CIPL can only search for the best allocation in one iteration, lead into a more conservative average allocation. Additionally, DACO-ClPL allocated more charging piles to Reagan National Airport and Washington Union

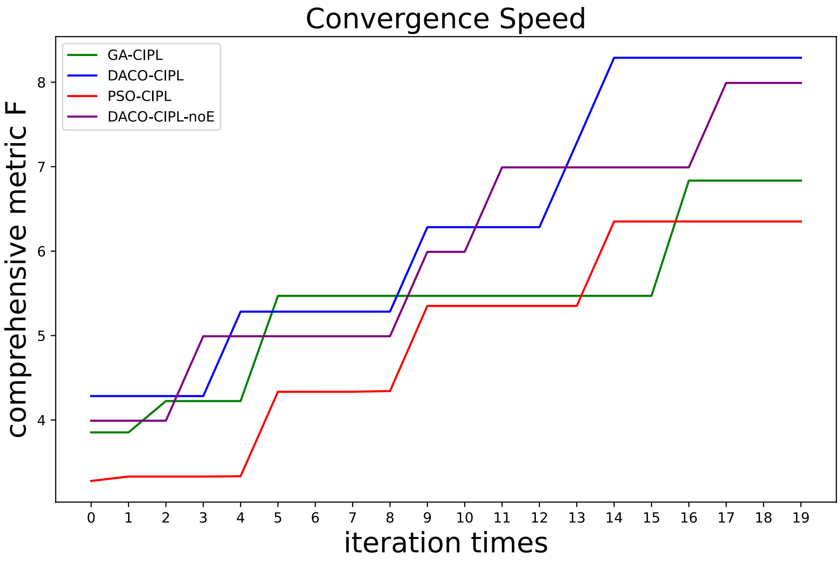

Fig. 11 Comparison of convergence speed on SC2-5

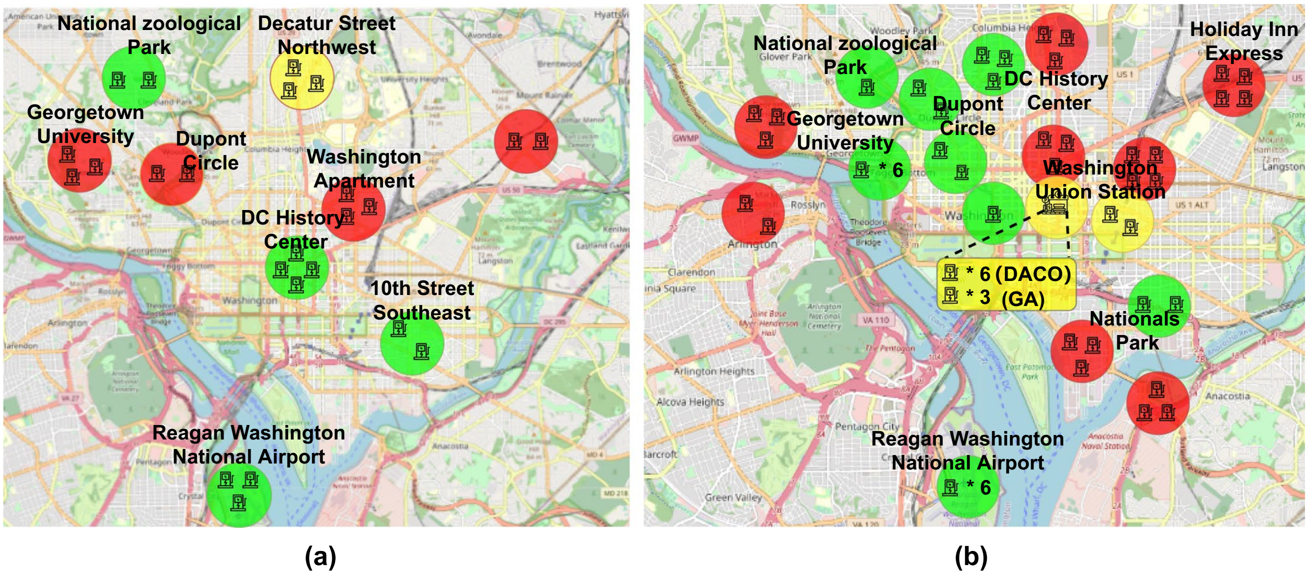

Fig. 12 Visualization ofcharginstation sites on map; (**a**) Results on test instance SC1-5; (**b**) Results on test instance SC2-5

Station, which are usually the end points of vehicle tracks. Concentration charging piles at the end points of tracks can satisfy higher charging demands and save more economic costs than allocating them in more dispersed track starting points. In conclusion, the ClPL obtained by DACO-CIPL can effectively meet the charging demands ofEV users while ensuring in the economic benefits of the operator.

6 Conclusion

It is very important to utilize CIPL for developing EVs and further improving the environment. ClPL consists of two subproblems: charging station siting and charging pile allocation. The existing CIPL methods independently solve the two subproblems and ignore their interconnections and influences, which restricts the rationality of CIPL. To address this issue, this paper proposes an algorithm called DACOCIPL to comprehensively solve the two CIPL subproblems. ln each iteration, the upper ant colony and the lower ant colony construct solutions for the charging station siting problem and the charging pile allocation problem in turn, respectively. The custom-made pheromone adjustment mechanisms of the two ant colonies can achieve bidirectional information transmission to comprehensively solve the two subproblems. The experimenta1 results have proven that DACO-CIPL has a strong ability to balance the two subproblem solutions and provides a better option for CIPL. However, DACO-CIPL only considers the interaction of the two subproblems of CIPL without considering the correlations among other subproblems. As the number of EVs increases and new charging standards are developed, it becomes increasingly important to consider other issues, such as the scale of charging stations, the type of charging piles, and EV users' preferences, as subproblems of CIPL. Therefore, using multicolony ant colony optimization to solve CIPL multisubproblems is a future research direction with great importance.

Table 4 Number of tracks in each region

Region	Number of tracks in SC1-5	Number of tracks in SC2-5
DC History Center	87	131
10th Street Southeast	56	-
Washington Apartment	**705**	-
Dupont Circle(SC2-5)	76	**1403**
Decatur Street Northwest	**896**	-
Georgetown University	**296**	131
Reagan Washington National Airport	**236**	**552**
Nationals Park	-	132
Washington Union Station	-	**1584**
Holiday Inn Express	-	**330**
National zoological Park	**157**	**212**

Acknowledgements The authors sincerely thank peer experts for their guidance and advice. This work is partly supported by NSFC Research Program (61906010, 62276010) and R&D Program of Beijing Municipal Education Commission (KM202010005032, KM202210005009).

Data Availability The authors confirm that the data supporting the findings of this study are available within the article.

Declarations

Conflict of interests The authors declare that there are no conflicts of interest related to this article.

References

1. Hossain Md et al (2022) Extreme level of co2 accumulation into the atmosphere due to the unequal global carbon emission and sequestration. Water Air Soil Pollut 233(4):1–6
2. Liu M, Zhang X, Zhang M, Feng Y, Liu Y, Wen J, Liu L (2021) Influencing factors of carbon emissions in transportation industry based on cd function and lmdi decomposition model: China as an example. Environ Impact Assess Rev 90:106623
3. Li C, Dong Z, Chen G, Zhou Bo, Zhang J, Xinghuo Yu (2021) Data-driven planning of electric vehicle charg- ing infrastructure: a case study of sydney, australia. IEEE Trans Smart Grid 12(4):3289–3304
4. Jie Xu, Huang Y (2022) The short-term optimal resource allocation approach for electric vehicles and v2g service stations. Appl Energy 319:119200
5. Jin Hojun, Lee Sangkeum, Nengroo Sarvar Hussain, Har Dongsoo (2022) Development of charging/discharging scheduling algorithm for economical and energy-efficient operation of multi-ev charging station. Appl Sci 12(9):4786
6. Yi Zhiyan, Liu Xiaoyue Cathy, Wei Ran, Chen Xi, Dai Jiangpeng (2022) Electric vehicle charging demand forecasting using deep learning model. J Intell Transp Syst 26(6):690–703
7. Liang Yi, Wang H, Zhao X (2022) Analysis of factors affecting economic operation of electric vehicle charging station based on dematelism. Comput Ind Eng 163:107818
8. Fausto Fernando, Reyna-Orta Adolfo, Cuevas Erik, Andrade Ángel G, Perez-Cisneros Marco (2020) From ants to whales: meta- heuristics for all tastes. Artif Intell Rev 53:753–810
9. Rodŕıguez A, Camarena O, Cuevas E, Aranguren I, Valdivia-G A, Morales-Castañeda B, Zaldívar D, Pérez-Cisneros M (2021) Group-based synchronous-asynchronous grey wolf optimizer. Appl Math Model 93:226–243
10. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1(4):28–39
11. Perumal Shyam SG, Lusby Richard M, Larsen Jesper (2022) Electric bus planning & scheduling: A review of related problems and methodologies. Eur J Oper Res 301(2):395–413
12. Pan Feng, Bent Russell, Berscheid Alan, Izraelevitz David (2010) Locating PHEV exchange stations in V2G. First IEEE international conference on smart grid communications. IEEE 2010:173–178
13. Schrijver Alexander (1998) Theory of linear and integer programming. John Wiley & Sons
14. Dokeroglu Tansel, Sevinc Ender, Kucukyilmaz Tayfun, Cosar Ahmet (2019) A survey on new generation metaheuristic algorithms. Comput Ind Eng 137:106040
15. He Yi, Song Z, Liu Z (2019) Fast-charging station deployment for battery electric bus systems considering electricity demand charges. Sustain Cities Soc 48:101530
16. Kadri Ahmed Abdelmoumene, Perrouault Romain, Boujelben Mouna Kchaou, Gicquel Céline (2020) A multi-stage stochastic integer programming approach for locating electric vehicle charging stations. Comput Oper Res 117:104888
17. Kınay Ömer Burak, Gzara Fatma, Alumur Sibel A (2021) Full cover charging station location problem with routing. Transp Res B Methodol 144:1–22
18. Hou H, Tang J, Zhao Bo, Zhang L, Wang Y, Xie C (2021) Optimal planning of electric vehicle charging station considering mutual benefit of users and power grid. World Electric Veh J 12(4):244
19. Ma GuoHan, Zheng QiPeng, Wang JingFeng, Zhang TianYi, Chang Da, Li XingLi (2023) Research on optimization of charging pile siting based on immune algorithm. Fifth International Conference on Computer Information Science and Artificial Intelligence (CISAI 2022). SPIE 12566:1148–1156
20. Zhou G, Zhu Z, Luo S (2022) Location optimization of electric vehicle charging stations: Based on cost model and genetic algorithm. Energy 247:123437
21. Zhang Y, Wang Y, Li F, Bin Wu, Chiang Y-Y, Zhang X (2020) Efficient deployment of electric vehicle charging infrastructure: Simultaneous optimization of charging station placement and charging pile assignment. IEEE Trans Intell Transp Syst 22(10):6654–6659
22. Li Wenzao, Yang Lingling, Wen Zhan, Chen Jiali, Wu Xi (2021) On the optimization strategy of EV charging station localization and charging piles density. Wireless Communications and Mobile Computing 2021:1–13
23. Dait Wenkuan, Lit Yuqing, Gan Xiaoying, Xie Gongquan (2018) Fast charging station placement with elastic demand. 2018 IEEE Global Communications Conference (GLOBECOM). IEEE 2018:1–7
24. Aboolian R, Berman O, Drezner Z (2008) Location and allocation of service units on a congested network. IIE Trans 40(4):422–433
25. Lin Wanting, Hua Guowei (2015) The flow capturing location model and algorithm of electric vehicle charging stations. 2015 International Conference on Logistics, Informatics and Service Sciences (LISS). IEEE 2015:1–6

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Junzhong Ji received the PhD degree in computer science and application technology from the Beijing University of Technology. He is a professor and PhD supervisor in the Computer Science college, Beijing University of Technology, Member of Chinese Association for Artificial Intelligence, and Senior member of the China Computer Federation. He was a visiting scholar at Norwegian University and the State University of New York ab Buffalo. His research interests include data mining, machine learning, swarm intelligence and bioinformatics.

Yuefeng Liu received the master degree in computer science and application technology from Beijing University of Technology in 2023. His research interests include swarm intelligence and charging infrastructure planning.

Cuicui Yang Associate professor at Beijing University of Technology. She received her doctor degree in computer science and technology from Beijing University of Technology in 2017. Her research interests over machine learning, computational intelligence, bioinformatics, and brain science.