International Journal of Engineering and Technology Innovation, vol. 15, no. 4, 2025, pp. 488-505

Optimization Method for Cross-Regional Scheduling of Retired Charging Pile Component Reuse

Yuan Wang[1], Junda Li[2], Dongjie Zhou[1], Wei Zhang[1], Tian'an Wang[2], Wei Li[2,*]

[1]Chuxiong Power Supply Bureau of Yunnan Power Grid Co., Ltd., Chuxiong, China

[2]Yunnan Power Grid Energy Investment Co., Ltd., Kunming, China

Received 08 June 2025; received in revised form 22 October 2025; accepted 28 October 2025

DOI: https://doi.org/10.46604/ijeti.2025.15250

Abstract

With the increasing number of decommissioned charging piles, efficient reuse of their components is essential for sustainable resource utilization and intelligent grid management. To address the challenges in recycling and scheduling retired charging pile components, this study proposes a cost-optimization approach for delivery planning in smart grid logistics. An Electric Vehicle Routing Problem with Time Windows (EDVRP-TW) model is formulated that considers vehicle capacity and time constraints. To solve it, an Improved Chicken Swarm Optimization Algorithm (ICOOT) is developed, integrating Circle chaotic mapping, spiral search strategy, and normal cloud mutation to enhance convergence speed and solution quality. Simulation experiments using real-world datasets demonstrate that the proposed method significantly reduces operation and maintenance costs, achieving up to an 11.77% cost reduction. The results validate the effectiveness and applicability of the model and algorithm in intelligent recycling and scheduling of grid materials.

Keywords: retired charging piles, smart grid logistics, green technology, EDVRP-TW, ICOOT

1. Introduction

In recent years, the rapid development and widespread adoption of grid charging infrastructure have driven innovations in the management of decommissioned charging pile components [1]. As a critical component of grid operations, managing decommissioned charging pile parts directly impacts the safety of equipment and the efficient use of resources. Therefore, the management of decommissioned charging pile components has become an increasingly important issue, especially with the continuous advancement of smart technologies, which have raised the demand for more intelligent monitoring and management systems.

Traditionally, the allocation of charging pile materials relies on manual scheduling, a cumbersome and inefficient process. Moreover, due to factors such as material inventory and delivery time constraints, material allocation often cannot be executed in a timely manner, thus affecting the normal operation and service quality of charging piles [2]. This issue becomes especially complex when considering the reuse of decommissioned charging pile materials. Hence, research into vehicle routing problems (VRP) is critical for planning reasonable delivery routes to coordinate various grid materials and decommissioned charging pile components, thus reducing transportation costs and optimizing delivery vehicle routes, which are particularly important for grid operations.

* Corresponding author. E-mail address: liwei6@stu.kust.edu.cn

With the increasing diversification of products and the expansion of logistics networks, the VRP has become a widely applied research direction for optimizing resource allocation and transportation routes in logistics and transportation fields, significantly improving transportation efficiency and effectively reducing operational costs [3]. Meanwhile, continuous advancements in radio frequency identification (RFID) technology have made real-time tracking and management of charging pile materials more efficient and accurate, greatly improving the accuracy and reliability of material scheduling.

By integrating VRP with RFID technology, not only can the scheduling process of charging pile materials be optimized, but overall operational efficiency and service quality can also be significantly improved [4]. Furthermore, the application of 5G technology in the grid sector enables real-time monitoring of electrical equipment, intelligent scheduling, and rapid fault handling, thereby improving grid stability and efficiency while enhancing the flexibility and convenience of grid charging material allocation [5]. In conclusion, VRP, RFID, and 5G technologies provide a solid software and hardware foundation for intelligent charging pile material allocation, and their integration will be key to advancing this field.

With the intensification of competition in the logistics industry and increased demand for faster delivery times, the research on more complex logistics problems has gained significant momentum, and the VRP with Time Windows (VRP-TW) is one of the main directions to solve such problems. Compared to VRP, VRP-TW involves more practical constraints, taking into account not only customer delivery requirements but also their time window preferences, making the model and solution process more complex than VRP [6]. For example, Wang et al. [7] designed a model that delivers requested goods to customers within a specified time window and then returns to the warehouse, with the optimization goal of minimizing total delivery costs. On the other hand, Guan et al. [8] studied the integration of delivery and pickup services with time windows and constructed a hybrid open/closed vehicle routing model to minimize logistics operational costs. S. Umar et al. [9] constructed a vehicle routing model considering inventory holding costs, transportation costs, and time costs, aiming to minimize transportation costs as the objective function.

In recent years, as global climate change issues have become more urgent and the focus on non-renewable resources has increased [10], green logistics has gradually become a key area of research [11], and the Electric Vehicle Routing Problem (EVRP) has naturally emerged as one of the core research topics [12].

Since electric vehicles rely on electricity as their power source, their driving range is related to the battery capacity, requiring charging at charging stations. Kancharla et al. [13] quantified the energy loss of electric vehicles as a coefficient to more accurately calculate transportation costs by considering the limited driving range and battery charging time. Maurizio Bruglieri et al. [14] focused not on reducing distances but on incorporating battery power and aimed at minimizing the weighted combination of vehicle usage, waiting time, charging time, and driving time. Xiao et al. [15] incorporated time window demands into VRP-TW, forming the EVRP-TW problem, and developed a new electric vehicle path model with mobile battery replacement along the route for more flexible and convenient delivery path planning [16]. In summary, while some progress has been made in EVRP-TW problems [17], there is limited research on the recycling planning of decommissioned materials, most of which only focus on delivery aspects [18], lacking a broader approach to the planning of decommissioned materials.

Therefore, this paper constructs a transportation path planning model for grid charging pile materials and their decommissioned components based on the integration of delivery vehicles and RFID, with RFID providing data support for the recycling and transportation of decommissioned grid materials.

The key to transportation path planning for grid charging pile materials and decommissioned components lies in optimizing delivery vehicle routes [19]. Heuristic and metaheuristic algorithms have been used to solve the EVRP-TW problem, with Particle Swarm Optimization (PSO) [20] and Simulated Annealing (SA) [21] algorithms showing good performance. However, for high-dimensional path planning problems involving complex constraints, a single classical algorithm may only provide a local optimal solution [22].

Therefore, various improvement strategies have been introduced to enhance the global optimization ability of algorithms. Bo Lin et al. [23] developed a hybrid variable neighborhood search/tabu search (VNS/TS) heuristic method that can effectively generate high-quality feasible solutions. A Lagrangian Relaxation method, solved by cutting-plane techniques, was also proposed to obtain lower bounds. Wang et al. [24] used the ENSGA-II algorithm, integrating the improved Clarke-Wright savings algorithm and Non-dominated Sorting Genetic Algorithm II (NSGA-II) to obtain the optimal solution, and applied a selection swap mechanism between clustering algorithms and ENSGA-II to enhance both global and local optimization capabilities for quick and effective optimal solution computation. Masoud Kahalimoghadam [25] improved the PSO algorithm by integrating smart droplets with a feedback control system monitoring algorithm to improve performance and convergence to the global minimum solution.

In summary, a feasible approach is to improve heuristic or metaheuristic algorithms to achieve better path-planning results. The Chicken Swarm Optimization (COOT) algorithm is a promising new metaheuristic algorithm [26], which has shown initial applications in various fields and has demonstrated strong capabilities in solving high-dimensional optimization problems [27]. Therefore, the COOT algorithm was improved to reduce its search blindness while maintaining its capability to solve high-dimensional problems, thereby enhancing solution speed without compromising accuracy. These latest studies highlight the increasing trend toward multi-objective and intelligent optimization in green logistics, but few have addressed the joint reuse–routing problem of retired charging-pile components. Therefore, this paper extends the literature by integrating component reuse, RFID-based monitoring, and improved meta-heuristic optimization within a unified framework.

This paper investigates the transportation path planning problem for grid charging pile components and their decommissioned parts by integrating RFID systems and constructing an intelligent delivery transportation system architecture. An electric vehicle delivery model was developed to optimize delivery distance, time, and cost, enabling delivery vehicles to complete grid material delivery tasks at minimal expense. Furthermore, the COOT algorithm was enhanced based on metaheuristic principles to plan vehicle delivery routes and facilitate the intelligent allocation of grid materials. In conclusion, the contributions of this paper are as follows:

(1) Introduce the process of grid charging pile material delivery tasks involving RFID and edge computing technologies, and construct a system architecture for the delivery tasks of grid charging pile materials and their decommissioned components.

(2) Construct a vehicle delivery model involving transportation vehicle energy constraints, cargo constraints, and grid warehouse constraints, with the goal of minimizing transportation costs and ensuring the timely delivery of vehicles.

(3) Improve the COOT algorithm using Circle chaotic mapping and spiral search strategies to enhance search speed and accuracy for effective grid charging pile material delivery path planning.

(4) Conduct simulation experiments using Genetic Algorithms (en), Particle Swarm Optimization (PSO), Simulated Annealing (SA), Grey Wolf Optimization (GWO), and Chicken Swarm Optimization (COOT) with Circle chaotic mapping and spiral search strategies, comparing the results to evaluate the developed model and algorithm.

The remainder of the paper is organized as follows: Section 2 introduces the grid charging pile material and decommissioned component delivery problem based on the integration of RFID systems; Section 3 establishes the vehicle delivery model with the objective of minimizing transportation costs to ensure timely completion of delivery tasks; Section 4 designs the improved path planning algorithm combining Circle chaotic mapping and spiral search strategies; Section 5 conducts simulation experiments and analyzes the results; Section 6 concludes the paper.

2. Task Description

In recent years, research on green logistics and resource recycling has rapidly expanded. For example, Zhang et al. (2023) developed a multi-objective electric vehicle routing model considering carbon emission and service reliability. Liu and Chen

(2022) proposed a hybrid GWO-PSO algorithm for green supply chain scheduling with time-window constraints. Wang et al. (2024) explored the reuse of decommissioned charging facilities in smart grids, emphasizing sustainable material recovery.

With the rapid development and widespread adoption of grid charging infrastructure, the management of decommissioned charging pile components has become a critical part of ensuring the safe operation of grid equipment and improving resource utilization. Efficient management of decommissioned charging pile components requires coordination of warehousing, transportation, and usage processes, as well as the integration of smart technologies for real-time monitoring and management of equipment status.

The management process of decommissioned charging pile components in the grid can be divided into three main modules: the grid equipment management center, the grid equipment warehouse, and the grid equipment usage area, forming a complete equipment lifecycle management system. As shown in Fig. 1, the process for managing decommissioned charging pile components in the grid is described in detail as follows: By installing a Radio Frequency Identification (RFID) system on charging piles in the grid equipment warehouse, the system can automatically identify the charging piles and track the location and status of items in the supply chain in real-time. In inventory management, it can update stock data in real-time to improve logistics efficiency and transmit equipment information from the warehouse to the equipment management center via the internet.

In the grid equipment usage area, the equipment management center can remotely start, stop, or adjust the operational parameters of the equipment via the internet or a dedicated communication network. It can also monitor the usage of the equipment in real-time and quickly receive alerts when equipment faults occur. This enables technicians to use remote connection technology to diagnose and troubleshoot the equipment remotely. As the central hub of the entire process, the equipment management center plays a decisive role. By utilizing information transmission technology, it can obtain real-time data on the usage of grid equipment and warehouse information, allowing for coordinated scheduling of warehouse deliveries to usage locations and the return of decommissioned charging pile components to the warehouse.

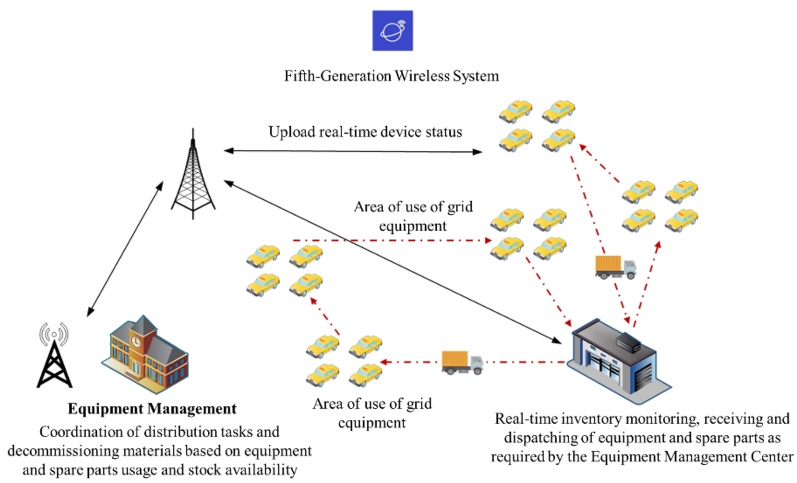


Fig. 1 Grid Charging Pile Component Management Process

The core of the equipment management center is to coordinate the entry and exit of equipment in the grid equipment warehouse, minimizing operational costs while maximizing the residual value of decommissioned charging pile components. It must also ensure optimal scheduling of charging pile materials to achieve global optimization, prevent delays in equipment delivery, and balance workloads across regions.

3. Related Definitions and Model Formulation

This section presents the related definitions, assumptions, and mathematical formulation of the proposed scheduling model for retired charging-pile components. Key parameters and constraints are defined to clearly describe the electric vehicle routing problem with time windows (EDVRP-TW). The model aims to minimize total transportation cost while ensuring efficient reuse and timely delivery of decommissioned components in smart-grid logistics.

3.1. Assumptions and definitions

The EDVRP-TW is formulated as a single-objective nonlinear programming model aimed at minimizing the cost of electric vehicles under constraints of time windows, cargo capacity, and power capacity. This section introduces some relevant symbols and variable definitions used in the EDVRP-TW model, as shown in Table 1. Additionally, several reasonable assumptions based on the basic assumptions in the traditional EVRP and DVRP optimization models are described as follows:

(a) Assumption 1: The demand, location, and time windows of customers are known, and the service time of all customers is ignored, with all customers being identical.
(b) Assumption 2: Electric vehicles travel at a constant speed, and the energy consumption rate for each vehicle is known and stable under different road sections and load conditions.
(c) Assumption 3: The number of vehicles is known and fixed, and the capacity of each vehicle is limited.
(d) Assumption 4: The task demands within the time window are fixed and do not change, and each vehicle completes its tasks within its working hours.

Table 1 Summary of the notation

Set	Description
B	Set of electric vehicles,$B = \{0,1,2,3,\ldots,b\}$,$b \in B$
C	Set of depots,$C = \{0,1,2,3,\ldots,c\}$,c$\in C$
I	Set of delivery destinations,$I = \{0,1,2,3,\ldots,i\}$,$i \in I$
J	Set of delivery tasks,$J = \{0,1,2,3,\ldots,j\}$,$j \in J$
L	Set of charging stations,$L = \{0,1,2,3,\ldots,l\}$,$l \in L$
N	Set of time periods,$N = \{0,1,2,3,\ldots,n\}$,$n \in N$
Parameter	Description
K_b	Electricity consumption per kilometer of EV b, $b \in B$
q	The energy consumption rate of EV
E	Total battery capacity of EV
E_{min}	Set the minimum battery charge percentage of EV
$\|Q_b\|$	The total number of depots served by EV b, $b \in B$
B	Unit time penalty cost for delayed arrival
M	Unit time penalty cost for violating time windows
Variable	Description
d_{it}	Distance from node i to t, $i, t \in I$
G_{ib}	The time EV b arrives at delivery destinations i, $b \in B$, $i \in I$
H_i	The end time of the transportation time window for delivery destination i， $i \in I$
O_c	The start time of the service time window for depot c, $c \in C$
AT_{cb}	The time when EV b arrives at depot c, $c \in C$, $b \in B$
H_c	The end time of the service time window for depot c, $c \in C$
GT_{cb}	The time when EV b arrives at depot or customer point c, $c \in C$, $b \in B$
e_{tb}	The remaining battery charge of EV b upon arrival at delivery destination t, $t \in I$, $b \in B$
e_{ib}	The remaining battery charge of EV b upon arrival at delivery destination i, $i \in I$, $b \in B$
g_b	The cargo weight of EV b, $b \in B$
G_b	The maximum loading weight of EV b, $b \in B$
v_b	The cargo volume of EV b, $b \in B$
V_b	The maximum volume of EV b, $b \in B$
p_i	The demand of delivery destination i, $i \in I$
p_c	The initial inventory of depot c, $c \in C$
L_c	The minimum inventory requirement of depot c, $c \in C$
Q_j	Priority coefficient of task j, $j \in J$
Q	Preset priority threshold
Decision variable	
O_{itb}	If EV b travels from depot i to t, $O_{itb} = 1$; otherwise, $O_{itb} = 0$
O_{icb}	If EV b travels from depot i to c, $O_{icb} = 1$; otherwise, $O_{icb} = 0$
x_{cib}	If EV b travels from depot c to i, $x_{cib} = 1$; otherwise, $x_{cib} = 0$
p_{lbn}	If EV b charges at node i in time period n, $P_{lbn} = 1$; otherwise, $P_{lbn} = 0$
y_{ci}	If warehouse i ships to i, $y_{ci} = 1$; otherwise, $y_{ci} = 0$
z_{ic}	If i ship to warehouse c, $z_{ic} = 1$; otherwise, $z_{ic} = 0$

3.2. Model formulation

The energy consumption of electric vehicles can be integrated into the EVRP model, as their energy consumption characteristics are closely related to the resistances encountered during driving. These include frictional resistance, air resistance, and rolling resistance [28] (Xinkai Wu & David Freese, 2015). Experimental studies have shown that the impact of these resistances on energy consumption can be summarized and quantified through a comprehensive coefficient, K_b.

This paper proposes a single-objective nonlinear optimization model aimed at minimizing the total transportation cost to solve the EDVRP-TW problem. The total transportation cost objective function is established in Eq. (1), where $d_{it}{\cdot}O_{itb}{\cdot}K_b$ epresents the transportation cost incurred by the electric vehicle traveling between the warehouse and the delivery destination, and $\beta(\max\{\sum_{b\in B} G_{ib} - H_i\,, o\})$ is the penalty incurred due to delays of the electric vehicle.

$$f = \min \sum_{i,t\in I} \sum_{b\in B} d_{it} \cdot O_{itb} \cdot K_b + \min \sum_{i\in I} \beta\left(\max\left\{ \sum_{b\in B} G_{ib} - H_i, 0 \right\} \right) \tag{1}$$

Constraint (2) ensures that each task is assigned to only one vehicle; this prevents multiple vehicles from being assigned to the same task. This can be represented as

$$\sum_{c,i\in C\cup I} \sum_{b\in B} x_{cib} = 1 \tag{2}$$

Constraint (3) ensures that after a vehicle reaches a node, it must leave that node, forming a complete path. This forms a closed loop in the transportation network and ensures no vehicle becomes stuck at a node, which can be illustrated by

$$\sum_{c,i\in C\cup I} \sum_{b\in B} x_{cib} = \sum_{c,i\in C\cup I} x_{icb} \tag{3}$$

Constraints (4) and (5) specify the time constraints for when a vehicle leaves and arrives at the warehouse. Constraint (4) deals with departure times, while constraint (5) addresses arrival times at the warehouse. This can be computed by

$$O_c \le AT_{cb} \le H_C, \forall c \in C, b \in B \tag{4}$$

$$O_c \le GT_{cb} \le H_C, \forall c \in C, b \in B \tag{5}$$

Constraint (6) specifies the time window constraint for when a vehicle reaches the delivery destination, which can be computed as

$$AT_{cb} + G_{icb} - M(1 - O_{icb}) \le GT_{cb}, \forall i \in I, c \in C, b \in B \tag{6}$$

Constraint (7) ensures that no closed loops are formed between partial warehouses by limiting the number of connections in the total path. This can be represented as

$$\sum_{i\in I, c\in C} O_{icb} \le |Q_b| - 1 \tag{7}$$

Constraint (8) ensures that the vehicle has enough battery power to travel from the delivery point i to point t. This can be expressed as

$$e_{tb} \le e_{ib} - d_{it} \cdot q \cdot O_{itb}, \forall i, t \in I, b \in B \tag{8}$$

Constraint (9) ensures that the vehicle's battery power cannot exceed the maximum battery capacity E, nor can it drop below zero. This can be represented as

$$0 < e_{ib} \leq E, \forall i \in I, b \in B \tag{9}$$

Constraint (10) specifies that when a vehicle charges at the charging station, its battery level cannot exceed the maximum battery capacity. This can be formulated as

$$\sum_{n \in N} qnP_{lbn} \leq E - e_{ib}, \forall i \in I, b \in B, l \in L \tag{10}$$

Constraint (11) ensures that the vehicle's remaining battery power upon arrival at the delivery point i is not below the minimum battery threshold. This can be computed by

$$\frac{e_{ib}}{E} \geq E_{\min}, \forall i \in I, b \in B \tag{11}$$

Constraint (12) ensures that the vehicle's cargo weight does not exceed its carrying capacity. This can be represented as

$$\sum_{b \in B} g_b \leq G_b \tag{12}$$

Constraint (13) ensures that the vehicle's cargo volume does not exceed its total loading capacity. This can be formulated as

$$\sum_{b \in B} v_b \leq V_b \tag{13}$$

Constraint (14) ensures that the vehicle is neither empty nor overloaded when departing from the warehouse. This can be expressed as

$$0 < g_{cb} \leq G_b, \forall c \in C, b \in B \tag{14}$$

Constraint (15) ensures that the total quantity shipped from the warehouse to all destinations does not exceed the warehouse's available inventory. This can be represented as

$$\sum_{i \in I} y_{ci} p_i \leq p_c, \forall c \in C \tag{15}$$

Constraint (16) ensures that the demand at each destination is fully supplied by one or more warehouses. This can be expressed as

$$\sum_{c \in C} y_{ci} p_i \leq p_c, \forall i \in I \tag{16}$$

Constraint (17) ensures that the harvested quantity from each warehouse does not exceed its maximum capacity. This can be formulated as

$$\sum_{i \in I} z_{ic} p_i \leq p_c, \forall c \in C \tag{17}$$

Constraint (18) ensures that each warehouse can serve both as a shipping point and a receiving point. This can be represented as

$$y_{ci} + z_{ic} \leq 1, \quad \forall c \in C, i \in I \tag{18}$$

Constraint (19) ensures that the inventory level after shipping from the warehouse does not fall below the specified minimum stock requirement. This can be expressed as

$$p_c - \sum_{i \in I} y_{ci} \cdot p_i \geq \mathsf{L}_c, \forall c \in C \tag{19}$$

Constraint (20) ensures that tasks with higher priority are scheduled for shipping first, which helps optimize the flow of goods based on priority and guarantees that time-sensitive deliveries are made before others. This can be represented as:

$$\sum_{c \in C} y_{ci} \cdot Q_j \geq Q, \forall i \in I, j \in J \tag{20}$$

Constraint (21) limits the frequency of receiving and shipping operations at a warehouse within a given time period. This can be formulated as

$$\sum_{i \in I} y_{ci} \cdot p_i \cdot z_{ic} = 0, \forall c \in C \tag{21}$$

4. The Chicken Swarm Optimization-Based Approach

Heuristic algorithms are widely used to solve large-scale logistics network optimization problems due to their efficiency. Genetic algorithms, with their high computational speed and ability to find global optimal solutions, are often applied to combinatorial optimization problems. At the same time, Particle Swarm Optimization (PSO) algorithms, with their simple structure, strong global search capability, and ease of implementation, are commonly used in solving various optimization problems. To overcome the limitations of the original Chicken Swarm Optimization (COOT) algorithm, such as slow convergence and local optima, this study proposes an Improved COOT (ICOOT) algorithm. ICOOT integrates three mechanisms—circle chaotic mapping to enhance population diversity, spiral search to balance exploration and exploitation, and normal cloud mutation to maintain solution robustness. These enhancements improve convergence stability and make ICOOT well-suited for the complex, multi-constraint scheduling of retired charging-pile components in smart-grid logistics.

4.1. Chicken Swarm Optimization

The Chicken Swarm Optimization (Coot) algorithm is a bio-inspired optimization method that simulates the foraging behavior of the white-crowned chicken. This algorithm is commonly used to solve the Vehicle Routing Problem with Time Windows (VRPTW), with the main goal of minimizing the total travel distance. The basic principles and optimization process of the algorithm are as follows:

Firstly, the white-crowned chickens are divided into different populations, and their upper and lower bounds are specified. This can be formulated as

$$CootPos(i) = rand(1, d) * (ub - lb) + lb \tag{22}$$

$$lb = \left[lb_1, lb_2, ..., lb_d\right], ub = \left[ub_1, ub_2, ..., ub_d\right] \tag{23}$$

The white-crowned chicken exhibits various group behaviors on the water's surface, ultimately leading to food or a specific location. These group behaviors are mainly divided into random movement, chain movement, guided movement, and leader movement. Random movement models the stochastic behavior of individuals, allowing them to explore the search space in a non-deterministic manner and maintain population diversity during optimization. This mechanism consists of a basic random position update and an additional perturbation component that enhances the algorithm's ability to escape local optima. The combined random movement process can be represented as

$$Q = rand(1, D) \cdot (ub - lb) + lb \tag{24}$$

$$Cootpos(i) = Cootpos(i) + A \cdot R_2 \cdot (Q - Cootpos(i)) \tag{25}$$

where $Cootpos(i)$ represents the current position of *Coot*, *lb* denotes the lower bound of the search space, *ub* represents the upper bound of the search space, R_2 is a random number within the interval [-1,1], and *A* can be expressed as follows

$$A = 1 - \frac{t}{T} \tag{26}$$

where *T* represents the maximum number of iterations, and *t* denotes the current iteration number.

Chain Movement:

The chain movement calculates the average position of two coots which can be expressed as

$$Cootpos(i) = 0.5(Cootpos(i-1) + Cootpos(i)) \tag{27}$$

where $Cootpos(i\text{-}1)$ represents the last *Coot*.

Guided Movement:

Individual coots update their positions based on the group leader, gradually moving closer to the leader. This can be represented in an equation as

$$K = 1 + (iMODN_L) \tag{28}$$

where *K* represents the selected *K*-th leader, and N_L denotes the number of leaders.

Following the leader movement describes the behavior in which individuals adjust their positions by following a superior solution, enabling the algorithm to accelerate convergence while maintaining structured search guidance. This mechanism models how individuals move toward the leader based on both deterministic and adaptive components, and the resulting position update can be represented as

$$Cootpos(i) = LeaderPos(K) + 2 \cdot R_1 \cdot \cos(2R\pi) \cdot (LeaderPos(K) - Cootpos(i)) \tag{29}$$

Leader movement:

The leader moves toward the current optimal neighborhood which can be computed by

$$LeaderPos(i) = \begin{cases} B \cdot R_3 \cdot cos(2R\pi) \cdot (gBest - LeaderPos(i)) + g_{Rer}, R_4 < 0.5 \\ B \cdot R_3 \cdot cos(2R\pi) \cdot (gBest - LeaderPos(i)) - g_{Rer}, R_4 \geq 0.5 \end{cases} \tag{30}$$

where R_3 and R_4 represent random numbers within the interval [-1,1], *gBest* denotes the current best position of each individual, and B can be expressed using Eq. (29). T is the maximum number of iterations, and t represents the current iteration number which can be expressed by

$$B = 1 - \frac{t}{T} \tag{31}$$

4.2. Improved Chicken Swarm Optimization

In order to enhance the optimization capability of the basic COOT algorithm and adapt it to the multi-constraint scheduling problem of retired charging-pile components, several improvement mechanisms are introduced in this section. The following subsection presents the detailed structure and working principles of the Improved COOT (ICOOT) algorithm.

4.2.1. Circle chaotic mapping

Circle chaotic mapping introduces randomness and diversity through its chaotic properties, enhancing the global search ability and convergence performance of various intelligent optimization algorithms. This chaotic mapping is widely applied in several stages, including population initialization, position and velocity updates, mutation and crossover, pheromone updating, and state transitions, significantly improving the efficiency and effectiveness of optimization algorithms in solving complex problems. The incorporation of Circle chaotic mapping can enhance the performance of the Chicken Swarm Optimization algorithm. By using Circle chaotic mapping to generate the initial population, the distribution of initial individuals becomes more uniform, thus improving the algorithm's global search ability. The formula for Circle chaotic mapping is as follows

$$x_{i+1} = \mathrm{mod}(x_i + b - (\frac{a}{2\pi})\sin(2\pi x_i), 1) \tag{32}$$

where a and b are control parameters, commonly taking values of 0.5 and 0.2, the chaotic trajectory state value range is (0, 1). By introducing Circle chaotic mapping during the position update process, chaotic disturbances are incorporated, enhancing the diversity of individuals and preventing the algorithm from getting trapped in local optima. During the population initialization phase, Circle chaotic mapping can cover a wider search space, thus avoiding overly concentrated initial solutions. Circle chaotic mapping enhances the global search capability and population diversity of the Chicken Swarm Optimization algorithm, while also improving the convergence speed and adaptability of the algorithm. These advantages make it an effective tool for improving the Chicken Swarm Optimization algorithm, especially in solving complex optimization problems.

4.2.2. Spiral Search Strategy

Spiral search is a search strategy first introduced in the Whale Optimization Algorithm (WOA). The spiral search strategy helps optimization algorithms explore the entire search space more effectively by simulating the behavior of organisms (such as whales) moving along a spiral path in nature. The spiral path covers both global and local areas, aiding the algorithm in finding better global solutions. The algorithm improves the convergence speed of optimization by combining a broad search in the early stages and a refined search in the later stages. It quickly covers the search space in the early stages and gradually approaches the optimal solution in the later stages, accelerating the overall convergence process.

In the Chicken Swarm Optimization (COOT) algorithm, population diversity is crucial for preventing premature convergence. The spiral search strategy introduces complex movement patterns, increasing the differences between individuals, thereby improving population diversity and adaptability. Overall, the spiral search strategy, with its unique search path and nonlinear characteristics, shows significant advantages in the Chicken Swarm Optimization model. It not only enhances the algorithm's global search ability and avoids local optima but also improves convergence speed, increases population diversity, and enhances robustness and adaptability, thereby significantly improving the algorithm's performance.

The core process of the spiral search strategy is the design of the spiral search path, which can be represented as

$$\mathbf{X}(t+1) = \mathbf{X}^{*} + D \cdot e^{bt} \cdot \cos(2\pi t) \tag{33}$$

where $\boldsymbol{X}(t+1)$ is the position of the individual at the t+1-th iteration, $\boldsymbol{X}^{*}$ is the position of the current best individual, D is the distance between the individual and the best individual, b is the constant that defines the spiral shape, and t is the current iteration number. The spiral search strategy effectively balances exploration (global search) and exploitation (local search). Through the nonlinear variation of the spiral path, the algorithm can adaptively adjust its search pattern at different stages, achieving comprehensive exploration of the search space and in-depth exploitation of potential optimal solutions. By introducing the spiral search strategy, the Chicken Swarm Optimization (COOT) algorithm has significantly improved its optimal search capability and convergence speed.

4.2.3. Normal Cloud Mutation Strategy

The Normal Cloud Mutation (NCM) strategy is a mutation approach based on the cloud model and normal distribution. It performs mutation operations by adding random variables that follow a normal distribution to the individual positions, thereby increasing the diversity of the population and enhancing the global search ability and convergence performance of the optimization algorithm. The specific mathematical description is as follows

$$\mathbf{X}' = \mathbf{X} + \mathbf{N}(0, \sigma^{2}) \tag{34}$$

where $\mathbf{X}'$ is the mutated individual, $\mathbf{X}$ is the individual before mutation, and $\mathbf{N}(0,\sigma^2)$ represents a normal distribution random variable with a mean of 0 and variance σ^2. In metaheuristic algorithms, introducing mutation operations can significantly enhance the algorithm's ability to escape local optima. The mutation operator introduces local random search, providing the algorithm with exploration diversity and flexibility. On the one hand, in the later stages of the solving process, the mutation operation can accelerate the algorithm's convergence to the optimal solution. On the other hand, it can also maintain the diversity of solutions, preventing the algorithm from prematurely converging to suboptimal solutions. By balancing local search and global exploration, the mutation operation improves the overall performance and adaptability of the algorithm. In this paper, common mutation mechanisms are analyzed, and after a comprehensive consideration, the Normal Cloud Mutation strategy is introduced. This strategy, based on random mutation operations with normal distribution, increases the diversity of the population and enhances the global search capability of the optimization algorithm.

4.3. Algorithm flow of ICOOT

This section presents the detailed workflow of the Improved Chicken Swarm Optimization (ICOOT) algorithm. Based on the standard COOT framework, the ICOOT algorithm integrates three enhanced mechanisms, including circle chaotic mapping for population initialization, a spiral search strategy to balance global and local exploration, and a normal cloud mutation operator to prevent premature convergence. These mechanisms collectively aim to improve solution diversity and optimization stability.

Algorithm 1 provides the pseudocode representation of the ICOOT algorithm. It describes how the population roles are assigned and updated in each iteration, how individuals exchange information through position adjustments, and how the improved operators are embedded within the update rules. The pseudocode allows for a clear understanding of the algorithm logic and supports reproducibility in implementation. Furthermore, the algorithm structure ensures that the search process gradually transitions from broad exploration in early iterations to refined exploitation in later stages, enabling the algorithm to converge efficiently toward high-quality solutions. This workflow also ensures that the ICOOT algorithm can be adapted to various complex optimization scenarios with dynamic constraints.

Algorithm 1: The improved chicken swarm optimization algorithm.

```
Initialize the first population of coots randomly by Eq. 22 and Eq. 23;
Add Eq. 32 to generate the initial population using the Circle chaos mapping;
Initialize the parameters of P = 0.5, NL (number of leaders), N_coot (number of coots);
N_coot = N_pop − NL;
Random selection of leaders from the coots;
Calculate the fitness of coots and leaders;
Find the best coot or leader as the Global optimum (gBest);
while the end criterion is not satisfied do
    Calculate A, B parameters by Eq. 26 and Eq. 31;
    if rand < P then
        R, R1, R3 are random vectors along the dimensions of the problem;
    else
        R, R1, R3 are random numbers;
    end
    for i = 1 to the number of coots do
        Calculate the parameter of K by Eq. 28;
        if rand > 0.5 then
            Update the position of the coot by Eq. 29;
        else
            if rand < 0.5 then
                i = 1;
                Update the position of the coot by Eq. 27;
            else
                Update the position of the coot by Eq. 25;
            end
        end
    end
    Calculate the fitness of coots;
    if fitness of coot < fitness of leader(k) then
        Temp = leader(k);
        leader(k) = coot;
        coot = Temp;
    end
    Adding Eq. 33 Updating iterative individual locations:
    for number of leaders do
        if rand < 0.5 then
            Update the position of the leader by Eq. 30.1;
        else
            Update the position of the leader by Eq. 30.2;
        end
        if fitness of leader < gBest then
            Temp = gBest;
            gBest = leader;
            leader = Temp // update Global optimum
        end
    end
    Iter = Iter + 1;
end
```

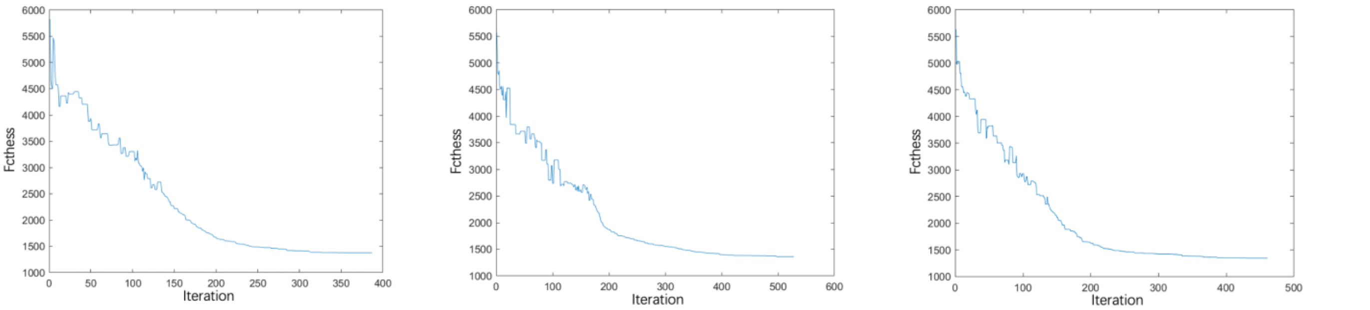
5. Simulation Experiments

This section compares the performance of the proposed method with five other heuristic algorithms: Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Simulated Annealing (SA), Grey Wolf Optimizer (GWO), and Chicken Swarm Optimization (COOT).

To evaluate the performance of the proposed algorithm, the publicly available R-C test suites [29] were utilized. In the test set, the locations of warehouses, customers, and charging stations are distributed on a 100×100 grid network. The number of customers ranges from 30 to 200, and the number of charging stations is set to 2, 4, and 6. The electric vehicle (EV) capacity is set to 5000 kg, the battery capacity is 962 kWh, the energy consumption per kilometer is 1 kWh/km, and the transportation cost per kilometer is 10 RMB/km, based on operational data provided by a logistics company in Southwest China. All simulation experiments were conducted using MATLAB R2024a as the algorithm programming tool, with the operating system as Windows 11, a computer with 16 GB of memory, and an AMD Ryzen 9 7940HX CPU @ 2.40 GHz. To ensure the accuracy and stability of the experimental results, each test instance was independently run 100 times, with the maximum number of iterations set to 1000 generations.

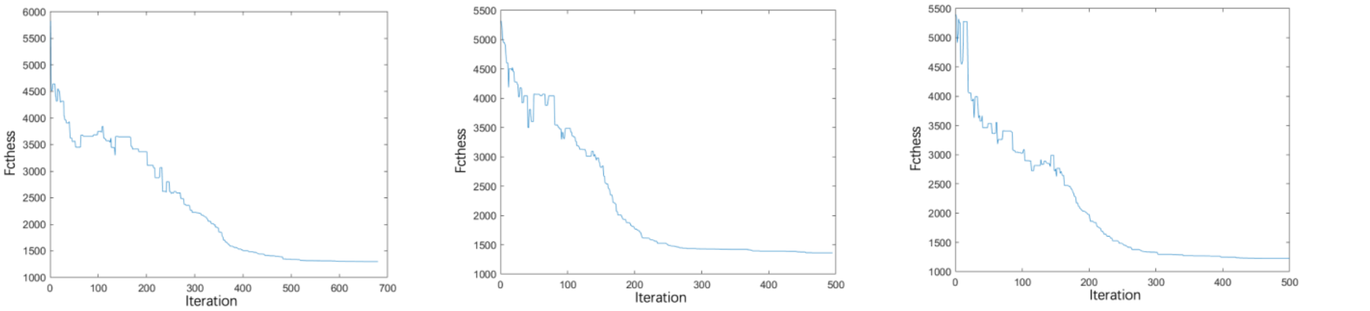
5.1. Results of distribution simulation

The simulation results after 100 runs are shown below, where Figs. 2 to 3 display the average best iteration curves of GA, PSO, SA, GWO, COOT, and ICOOT over 100 runs. To ensure fairness of comparison, all algorithms were executed under the same parameter settings and termination conditions. The performance curves illustrate the variation of the objective function during the iterative optimization process, reflecting both global search capabilities and convergence behavior.



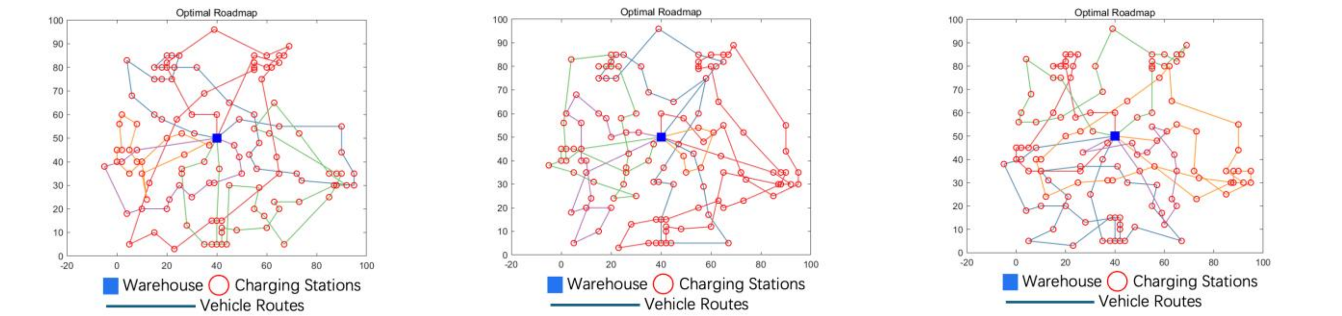
(a) Average iteration profiles of GA over 100 runs
(b) Average iteration profiles of PSO over 100 runs
(c) Average iteration profiles of SA over 100 runs

Fig. 2 Convergence Curves of GA, PSO, and SA Algorithms



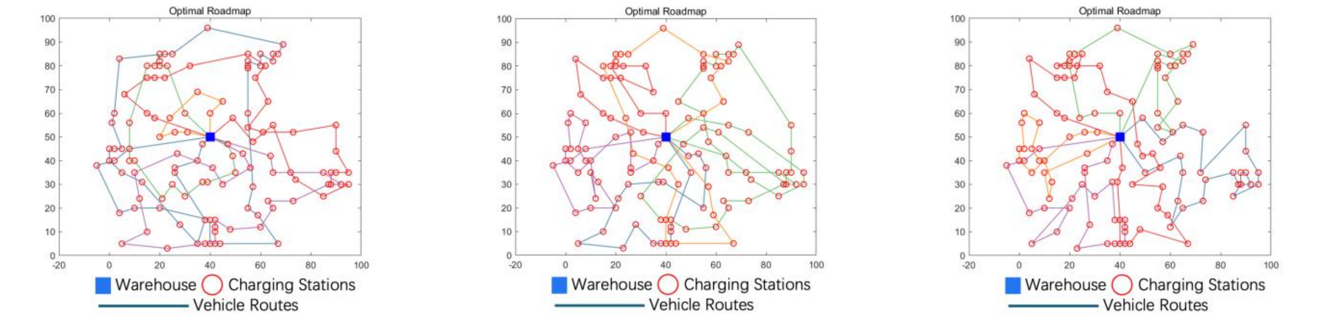
(a) Average iteration profiles of GWO over 100 runs
(b) Average iteration profiles of COOT over 100 runs
(c) Average iteration profiles of ICOOT over 100 runs

Fig. 3 Convergence Curves of GWO, COOT, and ICOOT Algorithms



(a) EV scheduling scheme obtained by GA solution
(b) EV scheduling scheme obtained by PSO solution
(c) EV scheduling scheme obtained by SA solution

Fig. 4 Vehicle Routing Plans Generated by GA, PSO, and SA



(a) EV scheduling scheme obtained by GWO solution

(b) EV scheduling scheme obtained by COOT solution

(c) EV scheduling scheme obtained by ICOOT solution

Fig. 5 Vehicle Routing Plans Generated by GWO, COOT, and ICOOT

Table 2 Algorithm-generated EV delivery plan

Algorithm	EVs	Optimal distribution routes	Cost
GA	EV-1	0->1->90->10->81->100->9->28->27->23->22->24->26->29->67->8->6->82->75->115->113->116->71->20->47->48->21->3->92->102->0	13748.149
	EV-2	0->64->58->59->62->88->83->41->45->98->79->96->12->109->91->4->63->65->60->57->56->5->61->7->66->25->76->89->0	
	EV-3	0->86->14->120->15->46->44->69->40->70->17->97->84->106->19->53->49->55->2->93->117->114->52->50->51->54->110->18->72->105->85->39->42->43->87->0	
	EV-4	0->111->119->74->34->37->38->68->11->94->99->35->32->31->108->30->103->0	
	EV-5	0->101->112->77->16->104->78->73->107->95->80->118->13->36->33->0	
PSO	EV-1	0->59->58->64->7->23->27->28->24->22->6->91->82->62->75->101->87->104->16->85->43->42->39->110->18->105->113->0	13623.139
	EV-2	0->88->55->49->53->83->97->17->70->40->69->44->46->98->15->45->41->72->106->84->19->51->50->52->2->93->60->4->63->65->61->57->56->5->3->21->48->47->54->20->71->92->115->102->89->0	
	EV-3	0->111->86->120->35->68->38->37->13->34->12->78->107->73->14->103->109->76->8->29->67->25->66->26->9->11->36->33->0	
	EV-4	0->117->114->116->77->112->0	
	EV-5	0->1->90->119->74->10->81->100->94->99->32->31->108->96->79->95->80->118->30->0	
SA	EV-1	0->1->86->78->46->44->69->40->70->41->45->98->96->79->107->12->30->14->87->85->105->72->110->17->39->42->43->15->95->80->118->13->36->33->0	13440.541
	EV-2	0->111->120->34->37->38->68->35->81->23->8->22->24->26->66->29->67->25->109->76->89->0	
	EV-3	0->90->27->28->9->100->94->11->99->10->91->6->7->64->61->58->56->5->57->60->65->63->4->75->102->0	
	EV-4	0->19->53->49->55->21->3->93->59->62->82->119->74->31->32->108->73->104->16->116->20->51->52->50->48->47->54->71->2->92->115->0	
	EV-5	0->77->18->97->84->106->83->88->117->114->113->112->101->103->0	
GWO	EV-1	0->1->111->86->14->120->45->46->15->12->34->37->38->68->35->11->94->9->24->22->7->5->56->57->58->61->60->65->63->4->75->105->72->18->110->39->42->43->95->80->118->13->36->33->0	12972.982
	EV-2	0->10->81->100->28->27->23->6->64->59->62->93->117->114->2->3->21->48->50->52->51->47->54->20->71->92->115->102->0	
	EV-3	0->101->112->77->16->104->78->73->107->31->32->99->29->67->26->66->25->8->76->0	
	EV-4	0->90->119->74->109->91->82->89->0	
	EV-5	0->88->83->55->49->53->19->106->84->97->17->41->70->40->69->44->98->96->79->108->30->103->87->85->116->113->0	

Table 2 Algorithm-generated EV delivery plan (continued)

Algorithm	EVs	Optimal distribution routes	Cost
COOT	EV-1	0->1->77->70->40->69->44->46->15->98->96->79->107->73->104->16->72->116->113->112->0	13628.561
	EV-2	0->90->10->81->100->9->28->27->23->91->6->8->25->67->76->89->0	
	EV-3	0->88->83->55->49->53->19->106->84->97->17->41->45->78->101->117->114->51->52->50->21->3->5->56->57->61->60->65->63->4->82->75->115->71->20->54->47->48->2->92->102->0	
	EV-4	0->93->62->59->58->64->7->22->24->26->66->29->109->103->87->85->43->42->39->110->18->105->0	
	EV-5	0->111->86->120->14->119->74->34->37->38->68->11->94->99->35->32->31->108->30->12->95->80->118->13->36->33->0	
ICOOT	EV-1	0->1->77->88->83->106->97->84->19->20->55->53->49->50->52->51->54->47->48->21->3->71->2->92->115->102->0	12300.71
	EV-2	0->10->81->100->9->28->27->23->22->66->29->8->6->91->82->101->112->113->116->85->105->72->18->110->17->39->42->43->16->87->0	
	EV-3	0->64->58->59->62->93->117->114->75->4->63->65->60->57->56->5->61->7->24->26->67->25->109->76->89->0	
	EV-4	0->90->119->74->34->37->38->68->11->94->99->35->32->31->12->108->30->103->0	
	EV-5	0->111->86->14->120->15->98->46->44->69->40->70->41->45->104->78->73->107->96->79->95->80->118->13->36->33->0	

Table 3 Comparison table of performance indicators for different algorithms in related problems

Indicator \ Algorithm	GA	PSO	SA	GWO	COOT	ICOOT
Cost(RMB)	13748.149	13623.139	13440.541	12972.982	13628.561	12300.71
Computation time(sec)	124	181	158	218	164	137
Relative gap between the best solutions(%)	11.77	10.75	9.27	5.47	10.80	-
Violation of the time window	0	0	0	0	0	0

The results are based on randomly selected runs from 100 trials. Figs. 2 and 3 show the convergence curves of the six algorithms, while Figs. 4 and 5 display the charging station component and decommissioned material distribution paths generated by GA, PSO, SA, GWO, COOT, and ICOOT. Table 2 presents the charging station component and decommissioned material distribution schemes generated by GA, PSO, SA, GWO, COOT, and ICOOT, including the delivery targets and transportation costs for each vehicle. Table 3 summarizes the comparative performance of GA, PSO, SA, GWO, COOT, and ICOOT. As shown, ICOOT achieves the lowest total cost with no time-window violations, demonstrating superior stability and efficiency over other algorithms.

5.2. Analysis of results

From Figs. 2(a) and 2(c), it can be observed that GA and SA have slower convergence speeds and relatively lower solution quality, indicating that both need improvement in balancing global exploration and local exploitation. From Fig. 2(b), although PSO shows faster convergence in the early stages, its stability is slightly inferior. Figs. 3(a) and 3(b) show that GWO and COOT have better stability and solution quality, but their local exploitation capabilities are somewhat lacking. From Fig. 3(c), it can be seen that the proposed ICOOT algorithm has stronger search capabilities during both the initial search and global search phases compared to GA, PSO, SA, GWO, and COOT, with superior convergence speed and accuracy.

From Fig. 4(a), it can be seen that GA exhibits higher flexibility in path selection, with a relatively even path distribution and a dense connection network in the central region. Fig. 4(b) shows that, compared to the GA scheme, PSO appears more concentrated in path selection, especially in the central region, with less consideration for the edge regions. Fig. 4(c) demonstrates that the SA algorithm shows good global search capability, with a wide path coverage and effective connections even in the edge regions. Fig. 5(a) illustrates that the GWO algorithm, while maintaining a certain search width, tends to form denser connections in the central area. Fig. 5(c) shows that ICOOT demonstrates higher efficiency in path selection, with a distribution that is both wide and somewhat concentrated, especially in the connections in the central area, highlighting the algorithm's advantage in balancing global search and local exploitation.

Additionally, from the detailed results in Table 3, the total delivery costs for GA, PSO, SA, GWO, COOT, and ICOOT are 13748.149 RMB, 13623.139 RMB, 13440.541 RMB, 12972.982 RMB, 13628.561 RMB, and 12300.71 RMB, respectively. ICOOT reduces the cost by 11.77%, 10.75%, 9.27%, 5.47%, and 10.80% compared to GA, PSO, SA, GWO, and COOT. In summary, the designed ICOOT demonstrates strong global and local search capabilities and high robustness. When the number of customers or depots increases, the computational complexity of ICOOT grows approximately linearly with the population size and iteration count.

Preliminary tests on extended cases (up to 400 customers) show that ICOOT maintains stable convergence behavior and acceptable computation time, with less than a 1.5× increase in runtime compared with the 200-customer scenario. These results indicate that ICOOT possesses good scalability and can be effectively applied to large-scale cross-regional scheduling problems. To verify the statistical reliability of the observed improvements, a paired t-test was conducted between the ICOOT and GWO algorithms using the results from 100 independent runs. The p-value was found to be less than 0.05, indicating that the cost reduction achieved by ICOOT is statistically significant at the 95% confidence level. In addition, a 95% confidence interval of [–1450.6, –1100.3] RMB for the cost difference further confirms the robustness of the improvement.

From the above results, it is clear that the proposed routing model comprehensively addresses the distribution objectives and transportation costs of delivery vehicles and can effectively generate delivery routes. ICOOT benefits further from the integration of enhanced strategies, such as Circle chaotic mapping and the spiral search strategy. Compared to general path planning algorithms, it reduces costs, thereby contributing to the intelligent and efficient distribution of charging station components and retired materials.

6. Conclusions

This study investigates the optimization of vehicle routing for the recycling and delivery of decommissioned charging pile components under time window constraints. The main contributions and findings of the research are summarized as follows:

(1) A single-objective nonlinear optimization model (EDVRP-TW) was constructed to minimize total transportation costs while ensuring timely deliveries within defined time windows.

(2) An Improved Chicken Swarm Optimization Algorithm (ICOOT) was developed, incorporating Circle chaotic mapping, spiral search strategy, and normal cloud mutation to enhance search performance.

(3) Extensive simulation experiments demonstrated that the proposed ICOOT algorithm consistently outperformed GA, PSO, SA, GWO, and standard COOT in terms of cost reduction and convergence speed.

(4) The proposed ICOOT algorithm achieves an 11.77% reduction in total delivery cost, implying a substantial saving in transportation and handling expenses for grid operators. Such improvement can significantly reduce annual operating costs and resource waste in the recycling and reuse of retired charging-pile components.

(5) The proposed approach offers practical value for intelligent scheduling in smart grid logistics and contributes to the efficient reuse of retired charging infrastructure components.

Future research will focus on extending the proposed approach toward a comprehensive multi-objective optimization framework that simultaneously minimizes cost, energy consumption, and environmental impact, thereby aligning with the green logistics vision emphasized in the Introduction. Moreover, future studies will explore real-time adaptive scheduling strategies under uncertain demand and integrate the developed model into RFID-based recycling and logistics pilot projects within smart grid operations to enhance its practical applicability and sustainability.

Acknowledgment

This work was supported by the Science and Technology Project of China Southern Power Grid Co., Ltd. (Grant No. YNKJXM20230458), Research on Key Technologies for Intelligent Logistics and Digital Management of Charging Infrastructure.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] Z. Wei, Y. Geng, H. Tang, Y. Zhao, and B. Lin, "Cost-Effective Sizing Method of Vehicle-to-Building Chargers and Energy Storage Systems During the Planning Stage of Smart Micro-Grid," eTransportation, vol. 21, article no. 100343, 2024.

[2] J. Luo and M. R. Chen, "Multi-Phase Modified Shuffled Frog Leaping Algorithm with Extremal Optimization for the MDVRP and the MDVRPTW," Computers & Industrial Engineering, vol. 72, pp. 84-97, 2014.

[3] L. Cheng, L. Zhong, X. Zhang, and J. Xing, "A Staged Adaptive Firefly Algorithm for UAV Charging Planning in Wireless Sensor Networks," Computer Communications, vol. 161, pp. 132-141, 2020.

[4] J. Luo and M. R. Chen, "Improved Shuffled Frog Leaping Algorithm and Its Multi-Phase Model for Multi-Depot Vehicle Routing Problem," Expert Systems with Applications, vol. 41, no. 5, pp. 2535-2545, 2014.

[5] Y. Li, H. Soleimani, and M. Zohal, "An Improved Ant Colony Optimization Algorithm for the Multi-Depot Green Vehicle Routing Problem with Multiple Objectives," Journal of Cleaner Production, vol. 227, pp. 1161-1172, 2019.

[6] J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco, "GRASP/VND and Multi-Start Evolutionary Local Search for the Single Truck and Trailer Routing Problem with Satellite Depots," Engineering Applications of Artificial Intelligence, vol. 23, no. 5, pp. 780-794, 2010.

[7] Y. Wang, X. Hong, Y. Wang, J. Zhao, G. Sun, and B. Qin, "Token-Based Deep Reinforcement Learning for Heterogeneous VRP with Service Time Constraints," Knowledge-Based Systems, vol. 300, article no. 112173, 2024.

[8] Y. Wang, L. Ran, X. Guan, J. Fan, Y. Sun, and H. Wang, "Collaborative Multicenter Vehicle Routing Problem with Time Windows and Mixed Deliveries and Pickups," Expert Systems with Applications, vol. 197, article no. 116690, 2022.

[9] S. U. Sherif, P. Asokan, P. Sasikumar, K. Mathiyazhagan, and J. Jerald, "Integrated Optimization of Transportation, Inventory and Vehicle Routing with Simultaneous Pickup and Delivery in Two-Echelon Green Supply Chain Network," Journal of Cleaner Production, vol. 287, article no. 125434, 2021.

[10] S. Erdoğan and E. Miller-Hooks, "A Green Vehicle Routing Problem," Transportation Research Part E: Logistics and Transportation Review, vol. 48, no. 1, pp. 100-114, 2012.

[11] C. Doppstadt, A. Koberstein, and D. Vigo, "The Hybrid Electric Vehicle – Traveling Salesman Problem," European Journal of Operational Research, vol. 253, no. 3, pp. 825-842, 2016.

[12] M. Schiffer and G. Walther, "The Electric Location Routing Problem with Time Windows and Partial Recharging," European Journal of Operational Research, vol. 260, no. 3, pp. 995-1013, 2017.

[13] B. Lin, B. Ghaddar, and J. Nathwani, "Electric Vehicle Routing with Charging/Discharging Under Time-Variant Electricity Prices," Transportation Research Part C: Emerging Technologies, vol. 130, article no. 103285, 2021.

[14] M. Bruglieri, F. Pezzella, O. Pisacane, and S. Suraci, "A Variable Neighborhood Search Branching for the Electric Vehicle Routing Problem with Time Windows," Electronic Notes in Discrete Mathematics, vol. 47, pp. 221-228, 2015.

[15] J. Xiao, S. Wang, J. Xiao, and G. Q. Huang, "Delivery Routing for Electric Vehicles with En-Route Mobile Battery Swapping," Transportation Research Part E: Logistics and Transportation Review, vol. 192, article no. 103838, 2024.

[16] W. Tu, Z. Fang, Q. Li, S. L. Shaw, and B. Chen, "A Bi-Level Voronoi Diagram-Based Metaheuristic for a Large-Scale Multi-Depot Vehicle Routing Problem," Transportation Research Part E: Logistics and Transportation Review, vol. 61, pp. 84-97, 2014.

[17] D. A. de Jesus Pacheco, D. M. Clausen, and J. Bumann, "A Multi-Method Approach for Reducing Operational Wastes in Distribution Warehouses," International Journal of Production Economics, vol. 256, article no. 108705, 2023.

[18] A. Gutiérrez-Sánchez and L. B. Rocha-Medina, "VRP Variants Applicable to Collecting Donations and Similar Problems: A Taxonomic Review," Computers & Industrial Engineering, vol. 164, article no. 107887, 2022.

[19] A. Subramanian, E. Uchoa, and L. S. Ochi, "A Hybrid Algorithm for a Class of Vehicle Routing Problems," Computers & Operations Research, vol. 40, no. 10, pp. 2519-2531, 2013.

[20] M. C. Chen, Y. H. Hsiao, R. H. Reddy, and M. K. Tiwari, "The Self-Learning Particle Swarm Optimization Approach for Routing Pickup and Delivery of Multiple Products with Material Handling in Multiple Cross-Docks," Transportation Research Part E: Logistics and Transportation Review, vol. 91, pp. 208-226, 2016.

[21] N. Boysen, "Truck Scheduling at Zero-Inventory Cross Docking Terminals," Computers & Operations Research, vol. 37, no. 1, pp. 32-41, 2010.

[22] H. Liu, W. Yang, and R. Kang, "A Correlation for Sand Erosion Prediction in Annular Flow Considering the Effect of Liquid Dynamic Viscosity," Wear, vol. 404-405, pp. 1–11, 2018.

[23] S. R. Kancharla and G. Ramadurai, "Electric Vehicle Routing Problem with Non-Linear Charging and Load-Dependent Discharging," Expert Systems with Applications, vol. 160, article no. 113714, 2020.

[24] Y. Wang, Y. Wei, X. Wang, Z. Wang, and H. Wang, "A Clustering-Based Extended Genetic Algorithm for the Multidepot Vehicle Routing Problem with Time Windows and Three-Dimensional Loading Constraints," Applied Soft Computing, vol. 133, article no. 109922, 2023.

[25] M. Kahalimoghadam, R. G. Thompson, and A. Rajabifard, "Self-Adaptive Metaheuristic-Based Emissions Reduction in a Collaborative Vehicle Routing Problem," Sustainable Cities and Society, vol. 110, article no. 105577, 2024.

[26] I. Naruei and F. Keynia, "A New Optimization Method Based on COOT Bird Natural Life Model," Expert Systems with Applications, vol. 183, article no. 115352, 2021.

[27] A. Albaker, M. Alturki, and A. Farah, "A Novel Modified COOT Algorithm for Parameter Extraction of PV Modules Including Measurement Noise Consideration," International Journal of Hydrogen Energy, vol. 80, pp. 357-369, 2024.

[28] X. Wu, D. Freese, A. Cabrera, and W. A. Kitch, "Electric Vehicles' Energy Consumption Measurement and Estimation," Transportation Research Part D: Transport and Environment, vol. 34, pp. 52-67, 2015.

[29] S. Zhang, M. Chen, W. Zhang, and X. Zhuang, "Fuzzy Optimization Model for Electric Vehicle Routing Problem with Time Windows and Recharging Stations," Expert Systems with Applications, vol. 145, article no. 113123, 2020.