S & M 3043

Green Logistics of Vehicle Dispatch under Smart IoT

Jianing Cao,[1] Jingcheng Zhang,[2] Mingxian Liu,[3] Shi Yin,[1*] and Yuqiang An[4]

[1]Faculty of Civil Aviation and Aeronautics, Kunming University of Science and Technology,
Kunming 650500, China
[2]Faculty of Science, Kunming University of Science and Technology, Kunming 650500, China
[3]Lijiang Power Supply Bureau, Yunnan Power Grid Co., Ltd., Kunming 674100, China
[4]Logistics Center, Hongyun Honghe Tobacco (Group) Co., Ltd., Lincang 650231, China

(Received April 7, 2022; accepted July 26, 2022)

Keywords: logistics dispatch, Industrial Internet of Things, RFID, green logistics vehicle routing, bald eagle search

As environmental problems increase, green development has become an indispensable part of the modern logistics system. Therefore, to help the logistics industry achieve the green development goals of energy saving and emission reduction, we studied the green logistics vehicle routing problem. Considering that the Internet of Things (IoT) plays an essential role in logistics scheduling, we constructed a Smart-IoT-based multi-objective green logistics vehicle scheduling model. This model guarantees the accuracy of vehicle-related information in all aspects of dispatch through IoT sensor technologies such as Global Positioning System (GPS) sensors, load sensors, and radio frequency identification (RFID) sensors and considers practical dispatch constraints such as dynamic vehicle carbon emission and physical constraints on heterogeneous vehicles. Furthermore, to address problems such as the poor convergence accuracy of the traditional path planning algorithm, we designed an improved path optimization algorithm by introducing the Levy flight strategy and simulated annealing mechanism in the bald eagle search algorithm to improve the search space and convergence speed of the algorithm. Then we conducted simulation experiments based on the scheduling task of a real logistics company. We also compared the designed algorithm horizontally with cutting-edge algorithms such as the sparrow search algorithm and crow search algorithm to verify the feasibility of the designed model and algorithm. Our results provide a practical reference for helping the logistics industry achieve green development.

1. Introduction

As a new generation of information technology, the extensive integration between intelligent Internet of Things (IoT) technology and the logistics industry has promoted the development of informatization, intelligence, and automation in the logistics industry. Owing to the efficiency of decision-making based on smart IoT technology, the logistics industry has reduced the redundancy and waste of many resources, providing a feasible path to achieve the green development goals of energy saving and emission reduction.(1)

*Corresponding author: e-mail: shi.yin@buaa.edu.cn
https://doi.org/10.18494/SAM3934

Intelligent IoT constitutes an information sensing network utilizing a variety of sensors combined with information transmission equipment and uses radio frequency identification (RFID) technology to realize intelligent identification and monitoring of dispatch links and provide a guarantee for the dispatch of green logistics vehicles.(2) Green logistics vehicle dispatch based on intelligent IoT is essentially a variant of the green vehicle routing problem (GVRP), which extends the optimization objectives of the general vehicle routing problem (VRP) by using IoT technology, thus organically integrating energy saving and emission reduction with vehicle path planning.(3) The GVRP, a research hotspot in recent years, has extensive applications.(4,5) Among them, Sadati and Çatay(6) studied the green vehicle scheduling problem involving multiple centers, constructed a mixed-integer linear programming model with the shortest travel distance as the optimization objective, and solved the problem in a short time using a hybrid general variable neighborhood search and Tabu search approach. Foroutan *et al.*,(7) on the other hand, studied the GVRP involving heterogeneous vehicles and developed a mixed-integer nonlinear programming model using operational and environmental costs as measures, and finally found a solution using a metaheuristic algorithm. Qiu *et al.*(8) designed a two-layer pollution path optimization model based on carbon pricing and freight schedules to provide effective decision-making for long-haul road freight reduction using an interactive algorithm integrating particle swarm optimization (PSO) and a domain search. Although the above studies show that the GVRP has been extended to multiple variants and is widely used in various industries, research on the smart IoT-based GVRP for the logistics industry is currently lacking.

The main methods commonly used to solve vehicle scheduling problems, which are essentially NP-hard problems, are precision-based algorithms(9–11) and biologically inspired swarm intelligence (SI) algorithms.(12,13) The green logistics vehicle scheduling problem studied in this paper has higher requirements on the solution algorithm's generalizability and global search capability because it involves complex factors such as multiple dispatch centers and heterogeneous vehicle groups.(14) The SI algorithm has a more comprehensive and thorough search capability than the exact algorithm, which facilitates a more available solution,(15) so many researchers have used SI to plan vehicle paths. Among them, the artificial bee colony algorithm,(16) bat algorithm,(17) and ant colony algorithm(18) have been effectively applied. In addition, to further improve the optimization performance of algorithms, researchers have attempted to introduce improvement strategies based on the classical algorithm and investigate the search mechanism of the algorithm.(19) Among them, Mehlawat *et al.*(20) improved the genetic algorithm (GA) using fuzzy simulation and solved the multi-warehouse GVRP with fuzzy travel time, and the experimental simulation results showed the improved feasibility and robustness of their algorithm. On the other hand, Yu *et al.*(21) developed a simulated annealing (SA) algorithm introducing a restart policy to solve a hybrid VRP and verified its superiority utilizing a numerical example. The above studies show that introducing improved strategies adapted to the search mechanism of the classical SI algorithm can effectively reduce the blindness of the search. Therefore, in this paper, we design an improved path planning algorithm, the improved bald eagle search algorithm (IBES), which is based on the bald eagle search algorithm (BES) combined with the Levy flight strategy and SA mechanism, to enhance its global search and jump out of local optima, thus providing a feasible solution for vehicle path planning.

We study the green logistics vehicle dispatch problem under intelligent IoT. The Global Positioning System (GPS), load, RFID, and vehicle dispatch processes are combined. On the basis of the vehicle dynamics model, a green logistics vehicle dispatch model with the lowest carbon cost, lowest dispatch cost, and highest dispatch fairness as the optimization objectives is constructed to meet the dispatch task needs of the logistics industry under green development. Furthermore, the Levy flight strategy and SA mechanism are introduced into the BES to complete the path planning of logistics vehicles and provide assistance to promote the realization of intelligence in the logistics industry. To summarize, the contributions of this paper are as follows.

1. We introduced the concept of basic logistics vehicle dispatch tasks involving multiple sensors under intelligent IoT.
2. We constructed a vehicle carbon emission cost adaptation function, vehicle in-transit transportation cost adaptation function, and vehicle distribution fairness adaptation function based on a vehicle dynamics model.
3. We designed the IBES combining the Levy flight strategy and SA mechanism to improve the convergence speed of the algorithm while ensuring convergence accuracy to achieve efficient and intelligent scheduling decisions in the logistics industry.
4. On the basis of the actual scheduling data of a logistics company, we employed path planning algorithms such as the sparrow search algorithm (SSA), crow search algorithm (CSA), SA, GA, and the proposed IBES for simulation experiments, and verified the effectiveness of the proposed algorithm and model by cross-sectional comparison.

The rest of the paper is organized as follows. In Sect. 2, the green logistics vehicle scheduling task flow under smart IoT is introduced; in Sect. 3, the developed green logistics vehicle scheduling model is presented; in Sect. 4, IBES incorporating the Levy flight strategy and SA mechanism is designed; in Sect. 5, simulation experiments are carried out, and the results are analyzed; finally, Sect. 6 summarizes the paper.

2. Task Description

Smart IoT combines IoT technology with intelligent algorithms to carry out data analysis and application. Real-world data is acquired through IoT technology and transmitted to the computing center using wireless communication technology, and the computing center realizes the integration, analysis, prediction, and application of the data using relevant intelligent algorithms. The green logistics vehicle scheduling task based on intelligent IoT is carried out using GPS sensors, load sensors, RFID antennas, and tags preset in the logistics park and logistics vehicles to achieve timely access to physical information of logistics vehicles and real-time verification of order data to ensure an orderly operation process from logistics vehicles entering the park to the completion of order loading. The main architecture of the logistics-industry-oriented traffic dispatch task based on intelligent IoT includes information sensing, network transmission, and intelligent application layers.

Real-time vehicle load information is obtained at the information sensing layer by a load sensor on the logistics vehicle. In contrast, the vehicle's physical information, the physical load limit, and other related information are obtained by an identity recognition sensor when entering the logistics park. Furthermore, when the logistics vehicle is loaded in the warehouse, the RFID electronic tag attached to the finished goods entering the vehicle is identified by an RFID sensor to obtain the actual loading information of the logistics vehicle. In addition, the logistics park obtains the location information of vehicles in real time through vehicle GPS sensors. At the layer of network transmission, the logistics center completes the data aggregation at the layer of information sensing in the logistics park and transits outside the park through a wireless local area network, 4G/5G, and other short-range and long-range communication technologies. It transmits the data to the intelligent application layer. At this layer, using the cloud computing platform of the logistics center and the collected vehicle and order information, vehicle distribution path planning is carried out using intelligent algorithms. After allocating an order, the logistics park sends the order of the vehicle to the vehicle mobile terminal, and the vehicle goes to the logistics warehouse for loading. In the loading process, the RFID identification results are summarized and synchronized to the logistics center for the comparison of order allocation results and for accounting, thus improving the scheduling accuracy at the warehouse level. The vehicle also sends real-time location information to the logistics center via a GPS sensor during the dispatch to carry out a real-time comparison of the predetermined dispatch results, thus ensuring the survivability and accuracy of dispatch. The described task architecture of the logistics-industry-oriented traffic scheduling based on intelligent IoT is shown in Fig. 1. In this paper, we mainly establish and solve the vehicle scheduling model at the intelligent application layer based on the scheduling task architecture for efficient utilization at the resource scheduling layer, thus promoting the green development of the logistics industry and reducing carbon emission.

3. Vehicle Dispatch Model

3.1 Model assumptions

The distribution task of logistics vehicles has the requirements of long-distance transport and high timeliness; therefore, its dispatch process is different from general urban dispatch, and it is difficult to fully consider relevant factors such as road conditions, vehicle dynamic speed, and other related factors. We make the following assumptions by combining the vehicle scheduling system of logistics companies with the actual scheduling task requirements.

1. A logistics vehicle always undergoes approximately uniform motion in the long-distance distribution process, i.e., at the maximum speed while ensuring the vehicle's safety.
2. The logistics vehicle should return to the dispatch center.
3. The goods distributed by this logistics company are homogeneous, such as finished cigarettes, and the logistics vehicle can carry all goods up to the maximum load, i.e., the weight of the goods is the deciding factor in establishing the model.
4. The statutory transportation permit used by the logistics company for scheduling is as follows.

(1) A single customer point order is handled by a single dispatch vehicle.
(2) The dispatch vehicle must arrive at the customer point of the order from the dispatch center within the valid time of the order permit issued by the management.

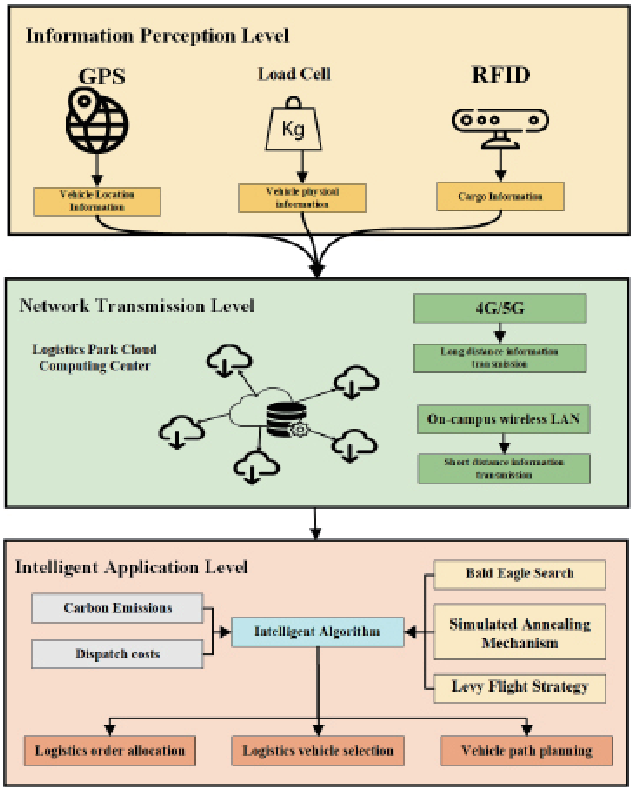

Information Perception Level
GPS
Load Cell
Kg
RFID
Vehicle Location Information
Vehicle physical information
Cargo Information
Network Transmission Level
Logistics Park Cloud Computing Center
4G/5G
Long distance information transmission
On-campus wireless LAN
Short distance information transmission
Intelligent Application Level
Carbon Emissions
Dispatch costs
Intelligent Algorithm
Bald Eagle Search
Simulated Annealing Mechanism
Levy Flight Strategy
Logistics order allocation
Logistics vehicle selection
Vehicle path planning

Fig. 1. (Color online) Scheduling task architecture.

3.2 Carbon emission model based on vehicle dynamics

The core of the research on green logistics vehicle scheduling based on intelligent IoT lies in combining carbon emission and vehicle scheduling. Current models incorporating vehicle carbon emission include the average speed model,(22) empirical regression model,(23) and kinetic model.(24) Among them, the average speed model is the simplest, but it requires a large amount of experimental data, while for the logistics scheduling studied in this paper, it is difficult to collect relevant data due to the operational cost and the complexity of the situation. The empirical regression model is also not applicable because it is only valid in the range of the available data. We thus employ the kinetic carbon emission model used by Barth and Boriboonsomsin(25) and integrate the vehicle engine speed, displacement, and other power factors to establish an effective carbon emission model for logistics vehicles, as follows.

$$CE = CEF \cdot FC \tag{1}$$

$$FC = \frac{1}{\alpha} \cdot FR \cdot \frac{D}{v} \tag{2}$$

$$FR = \frac{1}{\gamma} \cdot \omega \left(\mu NE + \frac{P}{\delta} \right) \tag{3}$$

$$P = \frac{1}{\varepsilon} \cdot P_t + P_a \tag{4}$$

$$P_t = \frac{1}{1000} v \cdot [M_k (a + g \times \sin\phi) + M_k \times g \times C_r \times \cos\phi + \frac{1}{2} \rho v^2 \times A \times C_a] \tag{5}$$

Equation (1) represents the carbon emission of the vehicle, Eq. (2) represents the amount of fuel consumed by the vehicle traveling between customer points *i* and *j*, Eq. (3) represents the fuel consumption rate of the vehicle, Eq. (4) represents the total power of the vehicle, and Eq. (5) represents the tractive power of the vehicle.

In these equations, CEF is the vehicle carbon emission factor, α is the unit conversion factor, γ is the fuel calorific value, ω is the mass ratio of fuel to air, μ is the engine friction coefficient, N is the engine speed, E is the engine displacement, δ is the efficiency parameter of the engine, ε is the engine efficiency, P_a represents other power requirements of the vehicle, such as steering and brakes, and it is generally set to zero, v is the vehicle travel speed, M_k is the total mass of the vehicle, g is gravitational acceleration, a is the vehicle acceleration, ϕ is the road gradient, C_r is the coefficient of resistance to vehicle rotation, ρ is the air density, A is the windward area of the vehicle, C_a is the air resistance coefficient, and D is the travel distance.

According to Assumption 1, owing to the long distribution distance of logistics vehicles and the road gradient, the change in vehicle acceleration is complex and unmeasurable; thus, the carbon emission model needs to be simplified, i.e., we set the acceleration as $A = 0$ and the road slope as $\phi = 0$. Combining Eqs. (1)–(5) yields the simplified carbon emission equation

$$CE = \left[\frac{CEF\omega}{\alpha\gamma} \left(\frac{\mu NE}{v} + \frac{\frac{1}{2} AC_a \rho v^2}{1000\varepsilon\delta} + \frac{M_k \cdot gC_r}{1000\varepsilon\delta} \right) \right] D. \tag{6}$$

All variables in Eq. (6) are constant except for M_k and D. To observe the relationship between carbon emission and M_k and D and to facilitate the subsequent modeling, we rewrite Eq. (6) as

$$CE = (W_1 + W_2 M_k) D, \tag{7}$$

where

$$W_1 = \frac{CEF\omega}{\alpha\gamma} \left(\frac{\mu NE}{\mathrm{v}} + \frac{\frac{1}{2} AC_a \rho v^2}{1000\varepsilon\delta} \right), \tag{8}$$

$$W_2 = \frac{CEF\omega}{\alpha\gamma} \cdot \frac{gC_r}{1000\varepsilon\delta}. \tag{9}$$

3.3 Vehicle scheduling optimization model for intelligent IoT-based green logistics

3.3.1 Real-time access to vehicle information

An advantage of logistics vehicle dispatch based on intelligent IoT is the use of sensor technology to accurately obtain vehicle-related information in each stage of the process of logistics vehicle dispatch, through which logistics resources are integrated and utilized to save energy in the use of logistics warehouse cargo resources. The information on logistics vehicles in the logistics warehouse is obtained as follows.

$$M_{cik}^t = M_k^t + L_k^t + \sum_P (O_P \times R_{ck}^P) \tag{10}$$

$$\sum_P (O_P \times R_{ck}^P) = \sum_i^A q_i^{tk} \tag{11}$$

$$M_{ijk}^t = M_{cik}^t - \sum_j^H q_j^{tk} \tag{12}$$

$$R_{ck}^P = \begin{cases} 1, & \text{vehicle monitors the entry of goods through the RFID sensor} \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

Equation (10) represents the total mass of a t-type vehicle when it departs from the dispatch center, where M_k^t is the net weight of vehicle *k*, L_k^t is the load weight of vehicle *k* when it enters the dispatch center, which is obtained by a load sensor in the vehicle, and O_p is the mass of goods *P*. Equation (11) represents the vehicle loading constraints, where *A* is the collection of order points for vehicle *k* and $\sum_i^A q_i^{tk}$ denotes the total weight of orders to be dispatched. Equation (12) represents the total mass of t-type vehicle *k* between customer points *i* and *j*, *H* is the set of dispatched order points for vehicle *k*, and $\sum_j^H q_j^{tk}$ represents the total weight of dispatched orders. In Eq. (13), when $R_{ck}^P = 1$ (0), the vehicle monitors (does not monitor) the entry of goods *P* through the RFID sensor.

3.3.2 Real-time access to vehicle information

Carbon emission is different from other scheduling optimization indicators, which leads to a lack of intuitive optimization results. Thus, on the basis of the established carbon emission model, we convert the carbon emission into the carbon cost by introducing carbon cost coefficients, which are combined with Eq. (7) to obtain the carbon emission costs of vehicles in the scheduling process as follows.

$$CC = \sum_{i,j\in I} w_c(W_1 + W_2 M_{cik}^t) d_{ci} \times X_{cik}^t + \sum_{i,j\in I} w_c(W_1 + W_2 M_{cik}^t) d_{ij} \times X_{ijk}^t + C_{cf}\left(\sum_{i}^{A} q_i^{tk} / M_c\right), \quad (14)$$

where

$$x_{cik}^t = \begin{cases} 1, & \text{denotes a } t\text{-type vehicle } k \text{ departing from} \\ & \quad \text{dispatch center } c \text{ and arriving at customer point } i \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

$$x_{ijk}^t = \begin{cases} 1, & \text{denotes a } t\text{-type vehicle } k \text{ departing from} \\ & \quad \text{customer point } i \text{ and arriving at customer point } j \\ 0. & \text{otherwise} \end{cases} \quad (16)$$

The first two terms on the right of Eq. (14) represent the cost of carbon emissions generated by vehicle *k* during travel, and the third term represents the fixed carbon emission cost of the transport center served by vehicle *k*. Equations (15) and (16) define the decision-making variables.

3.3.3 Scheduling cost adaptation function

The in-transit transportation cost of vehicle *k* during the dispatch process is

$$TC = \sum_{i,j\in I} [w_d (d_{ci} \times X_{cik}^t + d_{ij} \times X_{ijk}^t)] + C_{kp}, \quad (17)$$

where

$$C_{kp} = C_{maintain} + C_{tolls} + C_{other}. \quad (18)$$

Equation (18) represents the vehicle transport subsidy cost, which includes the vehicle maintenance cost $C_{maintain}$, vehicle passage cost C_{tolls}, and other costs C_{other}. w_d is the cost per unit distance traveled.

3.3.4 Scheduling fairness fitness function

Since the logistics scheduling task involves many logistics vehicles, the centralization of the distribution task will lead to the idleness of some vehicles. Therefore, the scheduling fairness is constructed as an optimization index to realize the greening and rationalization of resource utilization at the comprehensive utilization level of vehicles. The scheduling fairness fitness function of vehicles in the scheduling process is shown in Eq. (19), where the first term on the right indicates the deviation degree of the total weight of vehicle *k* and the average scheduling weight, and the second term indicates the deviation degree of the total distance of vehicle *k* and the average scheduling distance.

$$FF = (M_{cik}^{t} - \overline{M})^2 \times \left[\sum_{i,j \in A} (d_{ci} \times X_{cik}^{t} + d_{ij} \times X_{ijk}^{t}) - \overline{D} \right]^2 \tag{19}$$

In summary, we combine the vehicle carbon cost adaptation function *CC*, the scheduling cost adaptation function *TC*, and the scheduling fairness adaptation function *FF* to construct the general objective function

$$\min C = \sum_{c \in \mathrm{C}} \sum_{t \in T} \sum_{k \in V} CC \times TC \times FF\,. \tag{20}$$

Combining the established scheduling model with Assumptions 1–4 and the scheduling requirements of a logistics company, we summarize the constraints as follows.

$$G_{ck}^{t} = \begin{cases} 1, & \text{dispatch center } c \text{ recognizes the entry} \\ & \quad \text{of the vehicle } k \text{ through the vehicle GPS} \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

$$Q_{min}^{t} \le M_{cik}^{t} \le Q_{max}^{t} \tag{22}$$

$$\sum_{i \in I} X_{cik}^{t} = \sum_{j \in I} X_{jck}^{t} = 1 \tag{23}$$

$$\sum_{k \in V} X_{cik}^{t} \le \sum_{t \in T} \sum_{k \in V} G_{ck}^{t} \tag{24}$$

$$\sum_{c \in C} \sum_{t \in T} \sum_{k \in V} X_{cik}^{t} = 1 \tag{25}$$

$$\sum_{i \in A} \frac{d_{ci} \times X_{cik}^{t}}{v} \le T_i \tag{26}$$

$$\sum_{i,j \in A} \frac{(d_{ci} \times X_{cik}^{t} + d_{ij} \times X_{ijk}^{t})}{v} \le T_j \tag{27}$$

$$\sum_{i \in I} X_{ijk}^{t} = Y_{ik}^{t} \tag{28}$$

$$\sum_{j \in I} X_{ijk}^{t} = Y_{jk}^{t} \tag{29}$$

$$Y_{ik}^{t} \in \{0,1\} \tag{30}$$

$$\sum_{i,j\in H} x_{ijk}^{t} \le |A| - 1, 2 \le |A| \le n-1, A \in \{1,2,\cdots,n\} \tag{31}$$

$$\sum_{c} G_{ck}^{t} = 1 \tag{32}$$

Equation (21) indicates that G_{ck}^{t} can only be 0 or 1. Dispatch center c recognizes the entry of vehicle *k* through the vehicle GPS when $G_{ck}^{t} = 1$ but not when $G_{ck}^{t} = 0$. Equation (22) represents the vehicle load constraint, where $Q_{k\,\mathrm{max}}$ and $Q_{k\,\mathrm{min}}$ are the upper and lower load limits of vehicle *k*, respectively. Equation (23) guarantees the return of vehicles to the same dispatch center. Equation (24) indicates that the number of dispatched vehicles cannot exceed the total number of vehicles in the dispatch center. Equations (25)–(27) represent the dispatch permit constraint of the finished goods, where Eq. (25) indicates that each customer point can only be visited by one dispatch vehicle and Eqs. (26) and (27) indicate that the vehicle must complete the dispatch task within the maximum time limit. Equations (28) and (29) ensure that the vehicle leaves the customer point. Equation (30) indicates that y_{ik}^{t} can only be 0 or 1. *t*-type vehicle *k* serves customer point *i* when y_{ik}^{t} = 1 but not when $y_{ik}^{t} = 0$. Equation (31) ensures the elimination of subloops. Equation (32) indicates that the vehicle can only travel to one dispatch center.

4. Improved Bald Eagle Search Algorithm Incorporating Levy Flight and SA Mechanism

The smart IoT-based green logistics vehicle scheduling problem studied here is essentially a variant of the VRP, and such complex optimization problems often involve complex constraints, large scales, and multiple optimization objectives, making them difficult to solve. The SI-based metaheuristic algorithm is efficient and convenient in solving such complex optimization problems.(26) Therefore, we use the SI algorithm to optimize the cross-territory multi-constraint logistics scheduling path. As an emerging SI algorithm, the BES has been effectively applied in the transportation field due to its novel search mechanism, making it a good search space and giving it a high search speed.(27) Therefore, we have designed an improved BES (IBES) incorporating the Levy flight strategy and SA mechanism to solve the studied green logistics vehicle scheduling problem.

4.1 Traditional bald eagle search algorithm

The BES was proposed by Alsattar *et al.*(28) in 2020. Similarly to other group intelligence algorithms, its main idea is to model the behavior of bald eagles hunting for prey by dividing it into three stages: selecting the search space, searching for the prey, and diving to capture the prey, as outlined below.

4.1.1 Select search space

A bald eagle randomly selects the search space and determines the best search position according to the number of preys to facilitate the search for prey. The position $P_{i,new}$ of bald eagle *i* is updated as follows by multiplying the a priori information of the random search by *α*:

$$P_{i,new} = P_{best} + \alpha \cdot r(P_{mean} - P_i). \tag{34}$$

Here, *α* is the parameter for controlling position changes, *r* is a random number between 0 and 1, P_{best} is the best search position determined by the current BES, P_{mean} is the average distribution of the positions of the bald eagles at the end of the previous search, and P_i is the position of the ith bald eagle.

4.1.2 Search for prey

Each bald eagle flies in a spiral shape in the search space to search for prey and find the best position to dive to catch prey. The position in the spiral flight is updated using the following polar equations:

$$\theta(i) = a \cdot \pi \cdot rand\ , \tag{35}$$

$$r(i) = \theta(i) + R \cdot rand\ , \tag{36}$$

$$\begin{cases} xr(i) = r(i) \cdot \sin(\theta(i)), \\ yr(i) = r(i) \cdot \cos(\theta(i)), \end{cases} \tag{37}$$

$$\begin{cases} x(i) = xr(i) / \max(|\,xr\,|), \\ y(i) = yr(i) / \max(|\,yr\,|), \end{cases} \tag{38}$$

where *θ*(*i*) and *r*(*i*) are the polar angle and polar diameter of the spiral equation, respectively, *a* and *R* are parameters controlling the spiral trajectory, *rand* is a random number between 0 and 1, and *x*(*i*) and *y*(*i*) are the positions of bald eagle *i* in polar coordinates. After the coordinate transformation, the positions of the bald eagles are updated as

$$P_{i,new} = P_i + x(i) \cdot (P_i - P_{mean}) + y(i) \cdot (P_i - P_{i+1})\ . \tag{39}$$

4.1.3 Dive to catch prey

The bald eagles dive quickly from the best position in the search space to the target prey and attack the prey at the same time. The motion is described by the following polar equations:

$$\begin{cases} \theta(i) = a \cdot \pi \cdot rand, \\ r(i) = \theta(i), \end{cases} \tag{40}$$

$$\begin{cases} xr(i) = r(i) \cdot \sin(h(\theta(i))), \\ yr(i) = r(i) \cdot \cos(h(\theta(i)), \end{cases} \tag{41}$$

$$\begin{cases} x_1(i) = xr(i) / \max(|\, xr\,|), \\ y_1(i) = yr(i) / \max(|\, yr\,|). \end{cases} \tag{42}$$

The equation for updating the position of the bald eagle during the dive phase is

$$P_{i,new} = rand \cdot P_{best} + x_1(i) \cdot (P_i - c_1 P_{mean}) + y_1(i) \cdot (P_i - c_2 P_{mean}), \tag{43}$$

where c_1 and c_2 are the intensities of movement of the bald eagle toward the optimal and central positions, respectively.

4.2 Improvement strategies

As mentioned above, a bald eagle's hunting process can be divided into selecting a search space, searching the space for individual prey to be captured, and diving to capture the prey. When selecting a search space, the bald eagle uses the information available from the previous stage to determine the next search area. When randomly selecting another search area, it determines the corresponding domain on the basis of the previous search domain, which makes it easy for the bald eagle to fall into a local optimum when selecting the search space, thus resulting in the inability to accurately capture prey in the global search for the optimum process and greatly reducing the effectiveness of the algorithm. Therefore, we introduce the Levy flight strategy and SA mechanism to enrich the algorithm's search space and enhance the global search optimization ability of the bald eagle.

4.2.1 Levy flight strategy

Levy flight refers to a random walk with a heavy-tailed probability distribution of step lengths. The Levy flight strategy is widely used in the measurement and simulation of random and pseudo-random natural phenomena. The movement of predators, birds, and marine animals in search of food can be described by Levy flight, which obeys the following distribution formula with parameter (step length) s:

$$Levy(s) \sim u = t^{-1-\beta}, \quad \beta \in (0,2] \tag{44}$$

$$s = \mu / |\, \nu\,|^{1/\beta}, \tag{45}$$

$$\delta_\mu = \left[\frac{\Gamma(1+\beta)\cdot(\sin(\pi\beta/2))}{\Gamma((1+\beta)/2)\cdot\beta\cdot(2^{(\beta-1)/2})} \right]^{1/\beta}, \quad \delta_\nu = 1 \tag{46}$$

where $\beta = 1.5$ and $\Gamma(1+\beta) = (1+\beta)!$.

The classical bald eagle algorithm, controlled by the position change parameter α and random parameter r, tends to make the population converge prematurely and fall into a local optimum when selecting the search space. In this paper, the Levy flight strategy is introduced into the BES and acts on parameters α and r. The bald eagle searches for the prey to be captured by a unique spiral flight search, which increases the diversity of the search traversal, allowing it to jump out of a local optimum solution and improve the convergence of the original algorithm to find the best prey location. The position update, shown in Eq. (34), is improved to

$$P_{i,new} = P_{best} + \alpha \cdot r(P_{mean} - P_i) \times Levy. \tag{47}$$

4.2.2 Simulated annealing mechanism

When searching for prey using the polar spiral, eagles can explore new search space solutions. In this situation, SA has powerful local search capabilities. The use of SA can improve the local search ability and thus improve the exploration process of the local domain of the solution space. To improve the ability of a bald eagle to search for prey in a given space, the population location solution obtained by the bald eagle at that stage is used as the initial solution of SA, and cumulative iterations are performed to find the optimal solution so that the bald eagle can capture prey through more accurate diving and improve the quality of the global solution. The SA mechanism accepts solutions that are inferior to the current solution with a certain probability, allowing it to jump out of a local optimal solution and obtain the global optimal solution. When the domain solution is better than the current solution, it is considered as a completely new solution; otherwise, the probability of the latest solution is determined by the Boltzmann probability

$$p = \exp(-\theta / T). \tag{48}$$

Here, θ is the difference between the best solution and the currently generated domain solution and T is a parameter that decreases periodically in accordance with a certain law during the search.

4.3 Basic flow of the improved algorithm

The pseudo-code of IBES based on Levy flight and the SA mechanism is shown in Table 1.

The basic flow of IBES is as follows.

Step 1: Initialize the number and location of the bald eagle population.

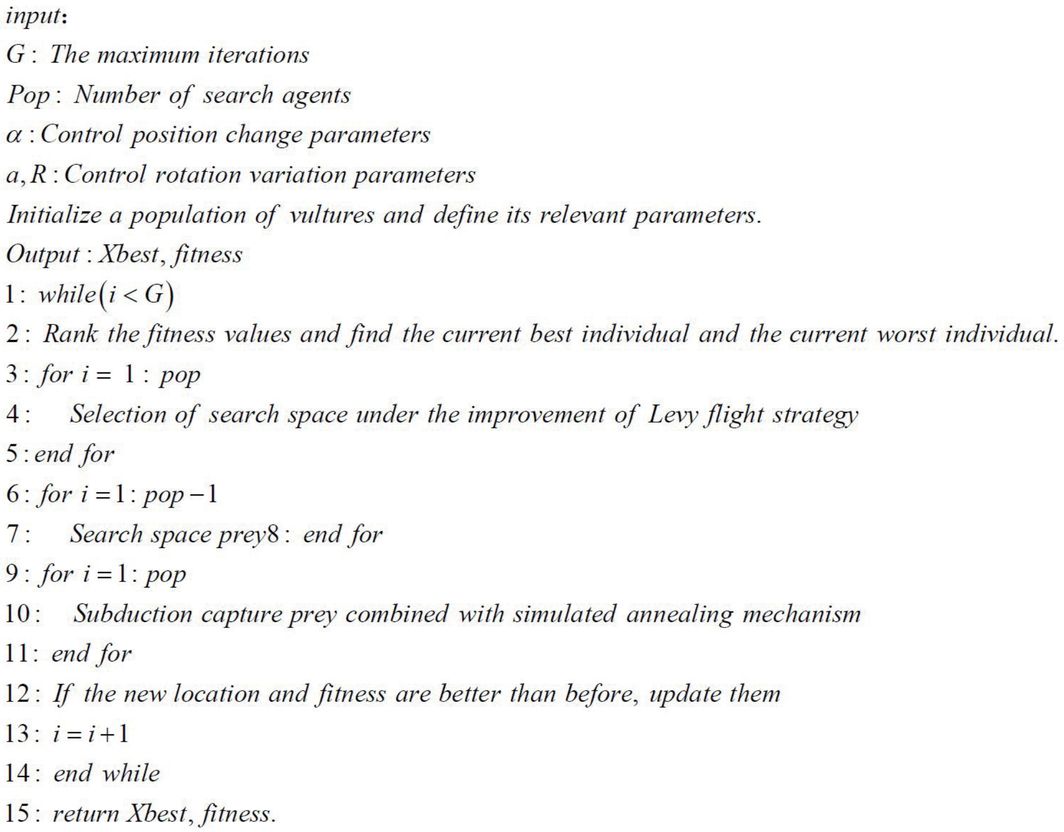
Step 2: Calculate the fitness function and initialize the population location.

Step 3: Select the search space using Eq. (47).

Step 4: Update each bald eagle position in polar coordinates using Eq. (38) and search for the prey using Eq. (39) combined with the SA mechanism.

Table 1
Pseudo-code of the IBES.

Improved bald eagle search algorithm (IBES)



```
input:
G : The maximum iterations
Pop : Number of search agents
α : Control position change parameters
a, R : Control rotation variation parameters
Initialize a population of vultures and define its relevant parameters.
Output : Xbest, fitness
while(i < G)
Rank the fitness values and find the current best individual and the current worst individual.
for i = 1 : pop
    Selection of search space under the improvement of Levy flight strategy
end for
for i = 1 : pop − 1
    Search space prey8 :  end for
for i = 1 : pop
    Subduction capture prey combined with simulated annealing mechanism
end for
If the new location and fitness are better than before, update them
i = i + 1
end while
return Xbest, fitness.
```

Step 5: Update each bald eagle dive position using Eq. (43).
Step 6: Update the optimal solution and optimal position.
Step 7: If the maximum number of iterations is reached, the optimal bald eagle position and the global optimal solution are output; otherwise, return to Step 3.

5. Model Validation and Evaluation

To verify the model and algorithm established in this paper, simulation experiments based on the scheduling task of a real logistics company were conducted. Three dispatch centers and 80 order customer points in a dispatch task were selected as the simulation data. IBES, BES, SSA, CSA, GA, and SA were used as simulation algorithms, and the results are compared.

5.1 Example validation

To verify the effectiveness of the proposed algorithm, four benchmark test functions were selected for numerical validation, and PSO, BES, and IBES were compared. To increase the validity of the results, each algorithm was run 30 times independently on each benchmark function. Table

2 shows the types, ranges of values, and optimal solutions of the four benchmark test functions. Table 3 shows the specific results of the benchmark test functions. Figures 2(a)–2(d) show the results of the algorithm validation for the (a) Sphere function, (b) Rastrigin function, (c) Ackley function, and (d) Kowalik function, respectively. The table shows that the mean value of IBES for each benchmark function is closer to the theoretical global optimum than the other two algorithms, and the standard deviation is smaller, indicating less fluctuation of the results. Therefore, IBES outperformed PSO and BES in terms of finding the optimal solution.

5.2 Simulation experiment

The parameters and the values of the model in the simulation experiment are shown in Table 4. The types of logistics vehicles and their related information are shown in Table 5. Selected information related to the 80 orders is shown in Table 6. The partial distance matrix between the dispatch centers and the customer points is shown in Table 7. Information about the logistics centers is shown in Table 8.

5.3 Simulation results

Using the simulation data, the proposed IBES and SA, GA, CSA, SSA, and BES were run 100 times, where some of the parameters of IBES are shown in Table 9. The convergence curves for the best and worst optimization results for each algorithm are shown in Figs. 3 and 4, respectively. The average convergence curve of each algorithm is shown in Fig. 5. The best, worst, and average optimization objective function values of the six algorithms in 100 runs are shown in Table 10.

The results of one of the 100 randomly selected runs are presented. The convergence curves of the six randomly selected algorithms are shown in Fig. 6. The optimized objective function

Table 2
Parameters related to the benchmark test function.

Functions	Types	Dimensionality	Range of values	Optimal solution
f_1	Sphere	30	[−100, 100]	0
f_2	Rastrigin	30	[−5.12, 5.12]	0
f_3	Ackley	30	[−32, 32]	0
f_4	Kowalik	30	[−5, 5]	0

Table 3
Algorithm validation results.

Statistical quantities	Algorithms	f_1	f_2	f_3	f_4
Average value	PSO	1617.6983	236.9663	9.4045	0.0092688
	BES	3.2966×10^{-19}	73.4004	0.21289	0.0071868
	IBES	1.9004×10^{-13}	66.7804	0.057934	0.0044889
Standard deviation	PSO	717.9285	30.8476	2.0599	0.0094827
	BES	1.2138×10^{-11}	68.8264	0.6496	0.0091435
	IBES	5.4203×10^{-13}	61.4417	0.31732	0.0080813

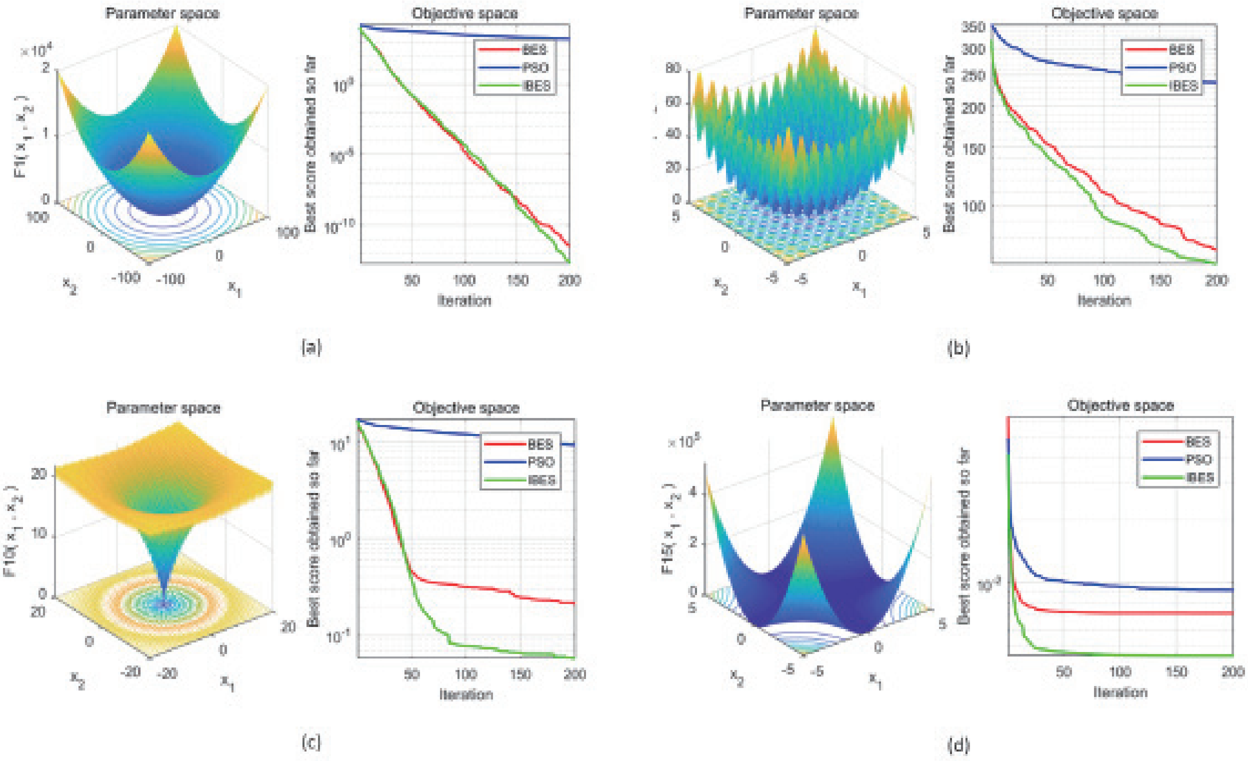


Fig. 2. (Color online) Graphs showing results of numerical verification.(a) Test result of Sphere function; (b) test result of Rastrigin function; (c) test result of Ackley function; and (d) test result of Kowalik function.

Table 4
Relevant parameters of the model.

Parameters	Meaning	General values
w_d	Travel cost factor	1.6
w_c	Cost of carbon factor	4.54
CEF	Carbon emission factor (kg/l)	2.32
α	Unit conversion factor (g/s $\rightarrow$ l/s)	737
δ	Engine efficiency parameters	0.9
N	Engine speed (rev/s)	38
E	Engine displacement (l)	5
μ	Engine friction coefficient	0.2
ω	Mass ratio of fuel to air	1
γ	Fuel calorific value (kJ/g)	44
ε	Vehicle transmission efficiency	0.4
g	Gravity acceleration (m/s^2)	9.8
C_r	Coefficient of resistance to vehicle rotation	0.01
ρ	Air density (kg/m^3)	1.201
A	Windward area of the vehicle (m^2)	3.912
C_a	Air resistance coefficient	0.7

Table 5
Logistics vehicle model parameters.

Vehicle type *t*	*t*1	*t*2	*t*3
Unladen mass /t	5	8	8
Lower load limit /t	11	14	13
Upper load limit /t	28	28	25
Quantity /t	16	11	18
Subsidy cost/CNY	500	675	850

Table 6
Information table of 80 orders for a scheduling task.

Order number	Customer point	Order weight (t)	Permit time limit (day)	Order number	Customer point	Order weight (t)	Permit time limit (day)
1	Xiangcheng	2.5	5	41	Shaoshan	2.5	5
2	Zhumadian	5	3	42	Hengyang	1.5	4
3	Wuhan	5	3	43	Leiyang	1.5	4
4	Huangshi	2	3	44	Shaoyang	2	4
5	Daye	2.5	3	45	Wugang	2.5	4
6	Shiyan	1.5	4	46	Yueyang	1.5	4
7	Danjiangkou	2.5	5	47	Miluo	2.5	5
8	Yichang	4	5	48	Linxiang	4	5
9	Danyang	2	4	49	Changde	2	4
10	Xiangfan	3	5	50	Jinshui	3	5
11	Laohekou	5	4	51	Zhangjiajie	1.5	4
12	Zaoyang	2.5	5	52	Yiyang	2.5	5
13	Yicheng	1.5	3	53	Yuanjiang	2.5	4
14	Ezhou	3	3	54	Chenzhou	3	3
15	Jingmen	3.5	4	55	Zixing	3. 5	4
16	Zhongxiang	2.5	4	56	Yongzhou	2.5	4
17	Xiaogan	2.5	5	57	Lingshutan	2.5	5
18	Yingcheng	1.5	3	58	Huaihua	2.5	4
19	Anlu	5	2	59	Hongjiang	1.5	2
20	Jingzhou	5	3	60	Loudi	1.5	4
21	Shishou	3	3	61	Lingshuijiang	3	4
22	Honghu	3.5	4	62	Lianyuan	3.5	4
23	Huangzhou	2	5	63	Lianyungang	2	5
24	Macheng	5	3	64	Jishou	1.5	3
25	Wuxiang	5	2	65	Guangzhou	1.5	2
26	Xianning	2	5	66	Panyu	2	5
27	Chibi	1.5	3	67	Huadu	2.5	4
28	Suizhou	5	5	68	Zengcheng	1.5	5
29	Guangshui	2	4	69	Conghua	2	4
30	Enshui	2.5	4	70	Shaoguan	2.5	4
31	Lichuan	2.5	5	71	Lechang	2.5	5
32	Xiantao	5	3	72	Shenzhen	1.5	4
33	Qianjiang	5	3	73	Zhuhai	1.5	4
34	Tianmen	2	3	74	Shantou	2	4
35	Changsha	2.5	3	75	Chenghai	2.5	4
36	Liuyang	3	4	76	Foshan	3	4
37	Zhuzhou	2.5	5	77	Nanhai	2.5	5
38	Liling	4	5	78	Shunde	4	5
39	Xiangtan	2	4	79	Sanshui	2	4
40	Xiangxiang	3	5	80	Gaoming	3	5

Table 7
Distance matrix (km).

	Xiangcheng, Henan	…	Gaoming, Guangdong	*C*1	*C*2	*C*3
Xiangcheng, Henan	0	…	1211.09	1843	1694	1780
Zhumadian, Henan	248.7	…	1128.58	2199	2050	2143
Wuhan, Hubei	291.46	…	886.22	1893	1744	1842
Huangshi, Hubei	285.71	…	882.56	2072	1923	2009
Daye, Hubei	332.98	…	863.47	1987	1838	1901
Shiyan, Hubei	537.01	…	1116.77	1994	1845	2079
Danjiangkou, Hubei	361.18	…	1166.83	1925	1790	1970
Yichang, Hubei	334.16	…	888.21	1820	1686	1865
Danyang, Hubei	192	…	893.09	1793	1659	1838
Xiangfan, Hubei	248.37	…	1057.22	1987	1852	2032
…	…	…	…	…	…	…
Le Chang, Guangdong	1141	…	294	1855	1769	1676
Shenzhen, Guangdong	1160.7	…	163.07	1305	1219	1132
Zhuhai, Guangdong	1068.2	…	95.31	1765	1671	1676
Shantou, Guangdong	1122.1	…	408.29	1843	1722	1707
Chenghai, Guangdong	1068.3	…	2216.38	1358	1272	1230
Foshan, Guangdong	1069.1	…	414.53	1758	1638	1622
Nanhai, Guangdong	1125.3	…	79.23	1743	1636	1645
Shunde, Guangdong	1065.4	…	80.77	1675	1568	1577
Sanshui, Guangdong	1128.6	…	66.93	1325	1218	1227
Gaoming, Guangdong	1220.67	…	63.49	1358	1230	1354

Table 8
Logistics Center Information.

Number	*C*1	*C*2	*C*3
Name	Kunming	Honghe	Huize
Daily storage capacity (t)	795	536	90

Table 9
Algorithm parameters of IBEA.

Parameter	Meaning	Value
$iter_{max}$	Number of iterations	200
N	Population size	60
lm	Control position change parameter	2
a	Control rotational trajectory change parameter	4
R	Control the rotational trajectory change parameter	1.5
T	Simulated annealing initial temperature	500
$delta$	Temperature decay coefficient	0.95
Lk	Number of iterations at each temperature	10

values, running time, carbon cost CC, and dispatch cost TC of the six algorithms in this run are shown in Table 11, and the vehicle selection and dispatch results generated by IBES in this run are shown in Table 12.

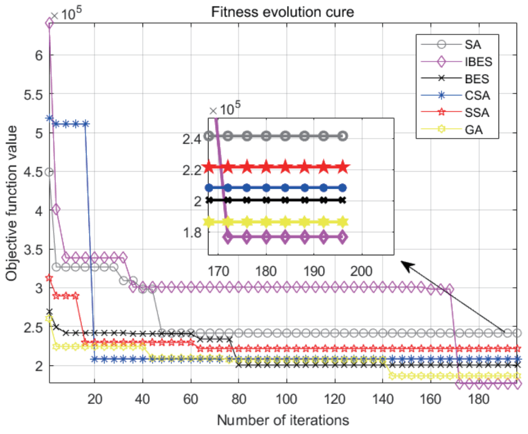


Fig. 3. (Color online) Best convergence curves of SA, GA, CSA, SSA, BES, and IBES in 100 runs.

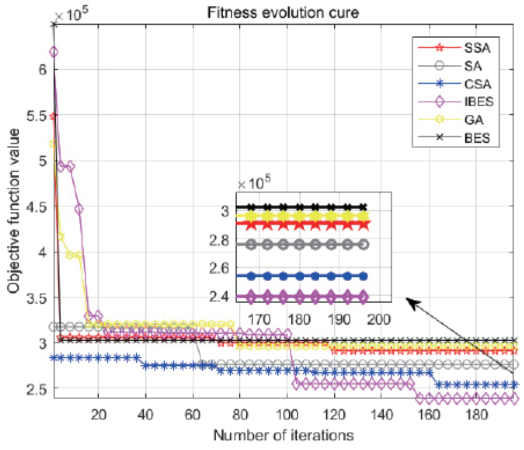


Fig. 4. (Color online) Worst convergence curves for SA, GA, CSA, SSA, BES, and IBES in 100 runs.

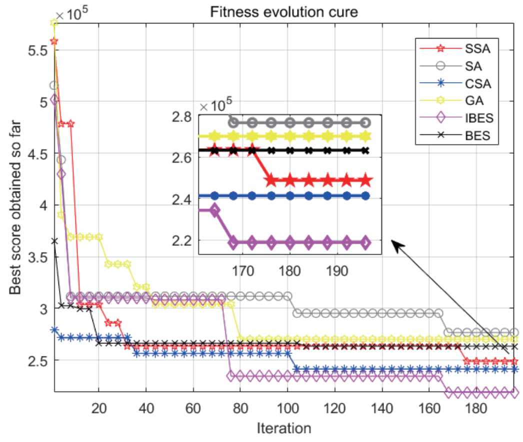


Fig. 5. (Color online) Average convergence curves of SA, GA, CSA, SSA, BES, and IBES after 100 runs.

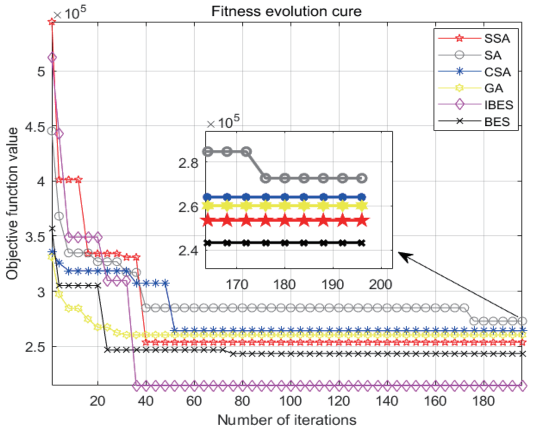


Fig. 6. (Color online) Convergence curves of SA, GA, CSA, SSA, BES, and IBES for one of the 100 runs.

Table 10
Run results of the six algorithms SA, GA, CSA, SSA, BEA, and IBEA.

Algorithm	Index		
	Best	Worst	Average
SA	2.42×10^5	2.86×10^5	2.77×10^5
GA	2.31×10^5	2.96×10^5	2.70×10^5
CSA	2.35×10^5	2.54×10^5	2.41×10^5
SSA	2.42×10^5	2.91×10^5	2.50×10^5
BES	2.27×10^5	3.03×10^5	2.48×10^5
IBES	1.77×10^5	2.39×10^5	2.19×10^5

Table 11
XResults of a particular run of the six algorithms SA, GA, CSA, SSA, BEA, and IBEA.

Algorithm	SA	GA	CSA	SSA	BES	IBES
Objective function value	2.70×10^5	2.60×10^5	2.67×10^5	2.56×10^5	2.42×10^5	2.11×10^5
Running time (s)	18.10	19.24	16.92	96.14	5.77	23.40
CC	2.30×10^4	2.27×10^4	2.31×10^4	2.42×10^4	2.22×10^4	2.12×10^4
TC	1.20×10^5	1.11×10^5	1.29×10^5	1.23×10^5	1.14×10^5	1.01×10^5

Table 12
IBEA results for one run of path generation

Dispatch Center	Scheduling path results	Vehicle type
*C*1	*C*1→34→9→43→*C*1 *C*1→17→4→*C*1 *C*1→16→30→45→5→*C*1 *C*1→52→57→40→63→*C*1 *C*1→18→54→66→6→10→*C*1 *C*1→1→24→68→*C*1 *C*1→61→20→55→25→56→*C*1 *C*1→50→67→7→*C*1 *C*1→70→47→53→78→*C*1 *C*1→15→46→74→72→42→*C*1 *C*1→3→80→33→*C*1 *C*1→75→71→*C*1 *C*1→11→13→*C*1	T1 T2 T2 T3 T2 T3 T3 T1 T2 T2 T3 T3 T2 T1
*C*2	*C*2→49→35→79→*C*2	T3
	*C*2→26→8→*C*2	T1
	*C*2→29→48→31→*C*2	T1
	*C*2→21→39→41→*C*2	T2
	*C*2→37→51→*C*2	T2
*C*3	*C*3→77→65→*C*3	T2
	*C*3→12→32→60→14→*C*3	T3
	*C*3→69→64→44→*C*3	T3
	*C*3→36→38→62→19→*C*3	T3
	*C*3→76→28→73→*C*3	T2
	*C*3→58→59→23→2→*C*3	T3
	*C*3→22→27→*C*3	T1

5.4 Analysis of results

From Figs. 3–5, it can be seen that, despite the poor convergence results in the early stage compared with the other algorithms, IBES can quickly jump out of a local optimal solution and search over a wider space to obtain better optimization results than the other algorithms. This is due to the introduced Levy flight strategy and SA mechanism. From the results of the six algorithms in Table 10, it can be seen that the best result of IBES is only 1.77×10^5 after 100 runs, the smallest value among the six algorithms and 26.9, 23.4, 24.7, 26.9, and 22.0% less than the values for SA, GA, CSA, SSA, and BES, respectively. The worst result of IBES after 100 runs is 2.39×10^5, which is also the smallest value among the six algorithms and 16.4, 19.3, 5.91, 17.9, and 21.1% less than the values for SA, GA, CSA, SSA, and BES, respectively. After averaging over 100 runs, the result of IBES is 2.19×10^5, which is 20.9, 18.9, 9.13, 12.4, and 11.7% less than the values for SA, GA, CSA, SSA, and BES, respectively. In summary, the introduced improvement strategies improved the global search capability and robustness of the algorithm.

Furthermore, the run results in Table 11 show that the value of the objective function for IBES is only 2.11 × 10^5, which is 21.8, 18.8, 21.0, 17.6, and 12.8% less than the values for SA, GA, CSA, SSA, and BES, respectively. The running time of IBES is 23.40 s, which is 7.14, 21.6, 38.3, and 306% higher than those of SA, GA, CSA, and BES, respectively, and 75.7% lower than that of SSA. The convergence curve in Fig. 6 shows that IBES converges after 40 iterations. The convergence time to the optimal solution of IBES is 7.83, 6.61, 8.23, 12.4, and 4.50% shorter than those of SA, GA, CSA, BES, and SSA, respectively. The scheduling cost TC of IBES is 15.8, 9.00, 21.7, 17.9, and 11.4% lower than those of SA, GA, CSA, BES, and SSA, respectively. Overall, the designed algorithm can reduce the costs for logistics companies compared with other metaheuristics.

In summary, compared with the traditional metaheuristic algorithms, our proposed IBES has good convergence speed and the ability to jump out of a local optimum while guaranteeing convergence accuracy, enabling it to effectively solve the green logistics vehicle scheduling problem. It is suitable for effective decision-making at the intelligent application level for logistics enterprises based on intelligent IoT and for realizing the efficient utilization of logistics resources.

6. Conclusions

We investigated the green logistics vehicle scheduling problem based on smart IoT. By constructing optimization indexes such as the vehicle dispatch carbon cost and transportation cost, we established a green logistics vehicle dispatch model applicable to the intelligent application level that involved multi-dimensional dispatch constraints such as multiple dispatch centers, heterogeneous vehicles, and distribution permit restrictions. Furthermore, IBES was constructed by introducing the Levy flight strategy and SA mechanism into BES to improve the established model. An experimental simulation with real logistics scheduling tasks showed that IBES has greater robustness and a higher convergence speed and accuracy than other metaheuristic algorithms and can effectively solve the green logistics vehicle scheduling problem and realize green and efficient logistics enterprises at the level of intelligent applications. The proposed model and algorithm can be extended to similar areas of logistics dispatch, such as urban super-distributions. In this study, we assumed an ideal vehicle and road situation, which requires further adaptation for actual application. In the future, we will focus on more realistic scheduling situations such as complex road conditions and a dynamic speed of vehicles in the logistics scheduling process to further extend the effectiveness and generalizability of the research.

Acknowledgments

This work was supported by a Hongyun Honghe Tobacco (Group) Co., Ltd., Technology Project under Grant HYHH2021XX04, the Science and Technology Plan Youth Project of Yunnan Science and Technology Department under Grant 202101AU070041, and the Talent Training Project of Natural Science Research Foundation of Kunming University of Science and Technology under Grant KKZ3202065057.

References

1 Y. Su and Q.-M. Fan: IEEE Access **8** (2020) 839. https://doi.org/10.1109/ACCESS.2019.2961701
2 Y. Song, F. R. Yu, L. Zhou, X. Yang, and Z. He: IEEE Internet Things J. **8** (2021) 4250. https://doi.org/10.1109/JIOT.2020.3034385
3 J. Kong and C. Chan: J. Beijing University of Posts and Telecommun. **43** (2020) 77. https://doi.org/10.13190/j.jbupt.2019-196
4 R. Moghdani, K. Salimifard, E. Demir, and A. Benyettou: J. Cleaner Prod. **279** (2021). https://doi.org/10.1016/j.jclepro.2020.123691
5 J. Mańdziuk: IEEE Trans. Emerging Top. Comput. Intell. **3** (2019) 230. https://doi.org/10.1109/TETCI.2018.2886585
6 M. E. H. Sadati and B. Çatay: Transp. Res. Part E Logist. Transp. Rev. **149** (2021). https://doi.org/10.1016/j.tre.2021.102293
7 R. A. Foroutan, J. Rezaeian, and I. Mahdavi: Appl. Soft Comput. **94** (2020). https://doi.org/10.1016/j.asoc.2020.106462
8 R. Qiu, J. Xu, R. Ke, Z. Zeng, and Y. Wang: Eur. J. Oper. Res. **286** (2020) 203. https://doi.org/10.1016/j.ejor.2020.03.012
9 S. Dabia, E. Demir, and T. V. Woenselc: Transp. Sci. **51** (2017) 607. https://doi.org/10.1287/trsc.2015.0651
10 J. Andelmin and E. Bartolini: Transp. Sci. **51** (2017) 1288. https://doi.org/10.1287/trsc.2016.0734
11 S. Yin, M. Zhu, and H. Liang: Chin. J. Aeronaut. **32** (2019) 1244. https://doi.org/10.1016/j.cja.2019.03.003
12 M. A. Islam, Y. Gajpal, and T. Y. ElMekkawy: Appl. Soft Comput. **110** (2021). https://doi.org/10.1016/j.asoc.2021.107655
13 X. Wang, T. Choi, H. Liu, and X. Yue: IEEE Trans. Intell. Transp. Syst. **17** (2016) 3132. https://doi.org/10.1109/TITS.2016.2542264
14 W. Wang and Y. Wu: Control and Decis. **28** (2013) 1799. https://doi.org/10.13195/j.kzyjc.2013.12.009
15 Y. Pan, H. Luo, L. Xing, and T. Ren: Control Theory Appl. **36** (2019) 1573. https://doi.org/10.7641/CTA.2019.90120
16 S. S. Choong, L. Wong, and C. P. Lim: Swarm Evol. Comput. **44** (2019). https://doi.org/10.1109/SMC.2017.8122629
17 E. Osaba, X. Yang, I. Fister, J. D. Ser, P. Lopez-Garcia, and A. J. Vazquez-Pardavila: Swarm Evol. Comput. **44** (2019) 273. https://doi.org/10.1016/j.swevo.2018.04.001
18 Q. Zhang and S. Xiong: Appl. Soft Comput. **71** (2018) 917. https://doi.org/10.1016/j.asoc.2018.07.050
19 S. Yin and M. Zhu: Eng. Comput. **36** (2019) 1608. https://doi.org/10.1108/EC-09-2018-0402
20 M. K. Mehlawat, P. Gupta, A. Khaitan, and W. Pedrycz: IEEE Trans. Fuzzy Syst. **28** (2020) 1155. https://doi.org/10.1109/TFUZZ.2019.2946110
21 V. F. Yu, A. A. N. P. Redi, Y. A. Hidayat, and O. J. Wibowo: Appl. Soft Comput. **53** (2017) 119. https://doi.org/10.1016/j.asoc.2016.12.027
22 G. Zhang, L. Wu, and J. Chen: Int. J. Environ. Res. Public Health **18** (2021) 1660. https://doi.org/10.3390/ijerph18041594
23 Y. Liu, Y. Yuan, H. Guan, X. Sun, and C. Huang: Res. Transp. Bus. Manage. **38** (2021). https://doi.org/10.1016/j.rtbm.2020.100487
24 R. Qiu, J. Xu, R. Ke, Z. Zeng, and Y. Wang: Eur. J. Oper. Res. **286** (2020) 203. https://doi.org/10.1016/j.ejor.2020.03.012
25 M. Barth and K. Boriboonsomsin: Transp. Res. Part D Transp. Environ. **14** (2009) 400. https://doi.org/10.1016/j.ejor.2020.03.012
26 J. Chen, B. Dan, and J. Shi: J. Cleaner Prod. **277** (2020). https://doi.org/10.1016/j.jclepro.2020.123932
27 S. A. Angayarkanni, R. Sivakumar, and Y. V. R. Rao: J. Ambient Intell. Hum. Comput. **12** (2021) 1293. https://doi.org/10.1007/s12652-020-02182-w
28 H. A. Alsattar, A. A. Zaidan, and B. B. Zaidan: Artif. Intell. Rev. **53** (2020) 2237. https://doi.org/10.1007/s10462-019-09732-5