Vehicle routing problem of contactless joint distribution service during COVID-19 pandemic

Dawei Chen, Shuangli Pan, Qun Chen *, Jiahui Liu

School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China

A R T I C L E   I N F O

Article history:
Received 6 April 2020
Received in revised form 8 August 2020
Accepted 17 September 2020
Available online 1 October 2020

Keywords:
Multi-vehicle multi-trip routing problem
Contactless joint distribution service
Tabu search operator
Artificial bee colony algorithm

A B S T R A C T

In order to prevent the further spread of the COVID-19 virus, enclosed management of closed gated communities is necessary. The implementation of contactless food distribution for closed gated communities is an urgent issue. This paper proposes a contactless joint distribution service to avoid contact between couriers. Then a multi-vehicle multi-trip routing problem for contactless joint distribution service is proposed, and a mathematical programming model for this problem is established. The goal of the model is to increase residents’ satisfaction with food distribution services. To solve this model, a PEABCTS algorithm is developed, which is the enhanced artificial bee colony algorithm embedded with a tabu search operator, using a progressive method to form a solution of multi-vehicle distribution routings. Finally, a variety of numerical simulations were carried out for statistical research. Compared with the two distribution services of supportive supply and on-demand supply, the proposed contactless joint distribution service can not only improve residents' satisfaction with the distribution service but also reduce the contact frequency between couriers. In addition, compared with various algorithms, it is found that the PEABCTS algorithm has better performance.

1. Introduction

Since the beginning of 2020, COVID-19 has spread globally. Given COVID-19 has a strong human-to-human transmission, the Chinese government had decided to block the traffic system in Wuhan, Hubei Province, one of the areas badly affected by COVID-19, in order to prevent further spread of the epidemic. The Wuhan government implemented unified enclosed management of 7148 gated communities in the city, and 9 million citizens in the city were required to be isolated at home. During the period of enclosed management, it has become a severe problem for the local government to design reasonable and efficient vehicles’ routing schemes for the food distribution to ensure the supply of food demand for residents in closed gated communities. In the following, for simplicity, we write "communities" to refer to the enclosed gated communities.

In order to solve this problem, the Wuhan government has introduced two distribution services: supportive supply and on-demand supply. Supportive supply refers to food directly provided by agricultural enterprises or foreign assistance to communities to ensure the basic food demands of residents. On-demand supply refers to residents purchasing personalized foods from the markets through e-commerce platforms and delivering it through couriers the next day. In order to restrict mutual contact between couriers and residents, couriers are not allowed to enter communities, so couriers can only drop off the foods at the unified designated delivery places of the communities. Fig. 1 shows the locations of 52 communities (red boxes) and 8 markets (green dots) in a region of approximately 4.5 km² in Wuhan.

However, the two existing distribution services have the following two problems. First, supportive supply and on-demand supply are independent of each other. The distribution tasks of supportive supply are completed by the vehicles organized by government, while e-commerce platforms or large local markets complete the distribution tasks of on-demand supply by third-party couriers. Separating food distribution tasks can ensure the effectiveness and punctuality of food distribution, but it will increase the cost of vehicle routing. In addition, the non-disclosure of the distribution information of the supportive supply and on-demand supply and the independent assignment of distribution tasks increase the possibility of contact between the couriers at delivery places. During the epidemic, not only the residents should be isolated, but also the mutual contact between couriers in pickup places (markets) and drop-off places (communities) should be avoided. Therefore, in order to reduce the cost of vehicle routing and avoid the contact between couriers, the contactless joint distribution service is proposed in this paper, which combines the characteristics of the two distribution services. Contactless joint distribution service collects the distribution tasks of supportive supply and on-demand supply together, and coordinates the distribution routes in a unified manner, focusing on the optimization of the routes of food distribution from suppliers or markets to communities. The contactless joint distribution service comprehensively considers the distribution tasks of supportive supply and on-demand supply, so it has greater flexibility in practical operations and can obtain a more efficient distribution scheme under the same condition.

Our contributions are as follows. First, we introduce two existing supply services of food distribution: supportive supply and on-demand supply.

http://dx.doi.org/10.1016/j.trip.2020.100233
2590-1982/© 2020 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).
Secondly, we propose a more efficient contactless joint distribution service for the food distribution of closed gated communities. Thirdly, we use special problem structures and design a new algorithm. Fourthly, we use numerical simulations for extensive case studies and management analysis.

The following context is organized as follows. Section 2 reviews related work on meal pickup and delivery problem and artificial bee colony algorithm. The contactless joint distribution service is proposed in Section 3. The issue and model are presented in Section 4. A new kind of enhanced artificial bee colony algorithm (PEABCTS) embedded with a tabu search operator and the mechanism of progressive construction solution is introduced in Section 5. In Section 6, extensive numerical studies are conducted to study the effectiveness of the proposed solution methodology. Finally, we present the conclusions from our study as well as directions for future research in Section 7.

2. Related work

The COVID-19 outbreak is one of the public health emergencies, which also includes H1N1 influenza, bora virus disease, and other illnesses such as severe acute respiratory syndrome (SARS). There are some studies (Sheu and Pan, 2014; Wei and Arun, 2007) attempted to combine medical rescue with emergency logistics, delivering goods to the affected areas, and evacuating the injured to the medical center. However, in response to contagious public health emergencies, medical logistics and food logistics should be isolated. Therefore, an innovative type of food distribution service should be launched in order to ensure the supply of essential food for the residents in communities.

The meal pickup and delivery problem (MPDP) is generally a multi-trip routing problem with a soft time window and multiple supplementary locations. Most of the researches on food delivery problems focus on the dynamic routing problem for couriers. To this end, Ulmer et al. (2017) proposed a dynamic stochastic model and an insertion heuristic algorithm to solve the order delivery problem of a single restaurant. Wang (2018) developed two mainstream heuristics: iterative local search and adaptive large neighborhood search. Yildiz and Savelsbergh (2019) proposed an accurate method to solve the meal delivery routing problem (MDRP). Reyes et al. (2018) introduced dynamic deterministic models for MDRP, developed heuristic methods based on auctions, and sought effective solutions by anticipating future system status.

The shared logistics service in MPDP is also related to the collaborative transportation model, which is seen as an attractive option to reduce the overall route selection cost by combining the vehicles of multiple shippers or carriers and reducing air transport (Liu et al., 2010). Steever et al. (2019) formally defined the virtual food court delivery problem (VFCDP), where a single customer’s order can include multiple restaurants. Wang (2018) newly offers a premium service called sharing + service, which allows vehicles to make new pickups before all meals on the vehicle are delivered. The distribution model in this paper has the characteristics of VFCDP and sharing + service at the same time. In our distribution model, residents have orders in each market, so when couriers visit a market, they can receive predetermined food for multiple communities.

The artificial bee colony algorithm (ABC) has been widely used to solve various VRP models and has successfully achieved better development and exploration. For example, Karaboga and Gorkemli (2011) proposed a new ABC algorithm called combinatorial ABC to solve the traveling salesman problem.
problem. Based on the ABC algorithm, Zhang et al. (2014) introduced the evolutionary concepts of genetic algorithm and EVRP local search algorithm and used them to solve environmental vehicle routing problems. Sedighizadeh and Mazariepour (2017) proposed a combined PSO-ABC algorithm to solve the VRP problem.

Some scholars have studied how to improve ABC's early development optimization ability. Some of them enhance ABC development ability by improving neighborhood search. For example, Su et al. (2019) studied the long-term carpooling problem with a time window and proposed an artificial bee colony algorithm combining five variable neighborhood searches and tabu list. Based on the artificial bee colony algorithm, Khan and Maiti (2019) modified multiple update rules and K-opt operations to solve the traveling salesman problem. The update rules are to exchange the solution of the problem by using the characteristics of the exchange sequence and the exchange operation. Karaboga and Gorkemli (2014) proposed a new definition of the best solution among the neighbors of current food sources to precisely search the onlooker stage. Xiang et al. (2018) introduced a gravity model into the ABC algorithm to select better neighbors for the current individual and improve development ability.

Some studies enhance the development ability of ABC by improving the search direction. For example, Xu et al. (2020) introduces a coevolution framework into ABC and designs a global best leading artificial bee colony algorithm with an improved strategy to accelerate its convergence and conquer the dependency of dimension separately. Kiran and Findik (2015) documented individual updates. If the individual is successfully updated, the update direction of the record will be used for the next generation. Zhu and Kwong (2010) introduced the global best information for ABC to improve its development ability and obtain higher solution accuracy.

Some scholars have studied methods to improve the accurate optimization ability of ABC in the later period. For example, adding multiple colonies strategies to the artificial bee colony algorithm (Gao et al., 2015; Ng et al., 2017; Zhou et al., 2019) can maintain the divergence of the population, avoid premature convergence, and improve Algorithm's later optimization ability. In addition, Szeto et al. (2011) have developed an enhanced ABC heuristic algorithm to improve the development ability of the employed bee stage. Iqbal et al. (2015) proposed a hybrid methodology combining the ABC algorithm with a multi-step local search. Gao et al. (2015) defined a new search direction mechanism to overcome the oscillation phenomenon of employed bees and proposed an intelligent learning mechanism to accelerate the convergence rate of the worst bees. Besides, it is also possible to better balance ABC's early global optimization ability and late accurate optimization ability by introducing adaptive programs (Kiran et al., 2015; Pandiri and Singh, 2019; Yin and Chuang, 2016).

The contactless joint distribution service includes not only contactless delivery between the couriers and the residents but also contactless transportation between the couriers during the distribution process. Neither the existing food delivery problem models nor the emergency rescue logistics models can be directly used to solve the contactless joint distribution problem. The food distribution model in this paper focuses on avoiding contact between couriers at pickup places and drop-in places.

### 3. Three types of supply services

Fig. 2 shows a flowchart of the three supply services, where the dash-line represents the information flow, and the solid line represents the food flow. The specific differences between the three types of supply services are as follows.

#### 3.1. Supportive supply

There are two food sources for supportive supply. One source is from a group of agricultural production enterprises with delivery capacities. These agricultural product enterprises put out a limited number of “special vegetable packages” every day. Another source is food assistance from other places. The supportive food is packaged and delivered by the vehicles to the scheduled communities. After the supportive food reached the communities, the community committees provide the information of the supportive food to the residents through the network. Then the residents choose and buy the supportive food provided, and the volunteers organized by community committees will deliver supportive food to residents' homes. Supportive food has a standard price slightly lower than the market price in order to guarantee the primary livelihood of residents in communities.

#### 3.2. On-demand supply

There are two food sources for on-demand supply. One source is from large chain supermarkets with delivery capacities. These supermarkets use their vans to deliver vegetables and meats directly to communities following the residents' purchase orders. Another source is from small-sized or medium-sized markets that sell food through e-commerce platforms. After residents buy food, the e-commerce platforms will deliver the food every next day. On-demand food is rich in variety, which meets the residents' personalized demands in addition to supportive food. However, because of the

---

**Fig. 2.** Three types of supply services.
wide variety of on-demand food and non-uniform distribution, couriers can only deliver food to the communities’ delivery places and pick them up by residents, respectively. The solid-line from “Residents” to “Communities” in Fig. 2b means that residents go to the delivery places to pick up their food. Residents’ pickup behaviors are random, which are not conducive to unified management, and easily leads to contact between multiple residents at delivery places.

3.3 Contactless joint distribution service

Although the distribution services of supportive supply and on-demand supply can significantly reduce the risk of infection brought by queued purchases in markets. However, these two models still have two shortcomings. First, food is distributed by multiple platforms and multiple companies. Although the punctuality of food distribution is guaranteed, returning the vehicles to the pickup places (markets and agricultural production enterprises) after dropping off the food will result in empty vehicles, which increases the costs of routing. Secondly, although the services of supportive supply and on-demand supply ensure that no contact occurs between the couriers and the residents during the distribution process, there may still be contact between the couriers at the pickup places and the drop-off places (communities). Residents also have contact behaviors at delivery places due to their random pickup behaviors. The contactless joint distribution service proposed can not only reduce routing costs by uniformly arranging distribution tasks, but also avoid contacting between multi-platform couriers. After the supportive and the on-demand foods were delivered to the communities, the community committees can uniformly manage the orders of on-demand food and supportive food, and then notify the residents to pick up the foods in batches or send them to the residents’ homes through volunteers.

4. The contactless joint distribution model

The vehicle routing problem of contactless joint distribution (shown in Fig. 3) can be described as: residents in multiple communities buy on-demand food in multiple markets, and one supplier provides supportive food to each community. Multiple vehicles are used to complete food distribution tasks. There is only one courier in each vehicle. It is required to arrange a reasonable route and schedule for each vehicle to reach communities, markets, and the supplier. In the case of keeping contactless between couriers, it is required to improve residents’ satisfaction with food delivery service as much as possible. The contactless joint distribution has many influencing factors in actual operation. In order to simplify and effectively establish a multi-vehicle multi-trip routing model, the following hypothesis are made for its distribution process:

1) The routes of all vehicles start and end in the same garage, and each vehicle has the same capacity.
2) Supportive foods come from a single supplier.
3) There is only one delivery place to drop off the food in each community, and only one delivery place to pick up food in each market and the supplier.
4) The food at delivery places of communities shall be picked up by the residents themselves or distributed by volunteers without contact.
5) The amount of food required in the markets from communities is determined and can be distributed by a vehicle at one time if its capacity allows.
6) The delivery is incomplete delivery, which means foods that are not delivered can be refunded, but it will reduce residents’ satisfaction with the delivery service.
7) The freshness of food will not decrease in the markets or the supplier, but it will decrease over time during the distribution process.

The symbols needed to describe the model of contactless joint distribution are shown in Table 1, and the established mathematical model is as follows:

\[
\max P = \sum_{i \in K} \sum_{j \in V} \left[ a \sum_{i \in B} u_{iik} \theta_{ijk} + \beta \sum_{i \in S} u_{iik} \theta_{ijk} \right] \]

s.t.

\[
t_{ik} + d_{ij}/v + \frac{u_{iik} + \sum_{k \in S} u_{iik}}{c_{1}} + M \cdot (1-x_{ijk}) \leq t_{jk}, \forall i \in V, \forall j \in K \]

\[
t_{ik} \leq T_{max}, \forall i \in V, \forall k \in K \]

\[
t_{ik} = 0, \forall k \in K \]

\[
\sum_{j \in V} \sum_{i \in K} x_{ijk} = \sum_{i \in K} \sum_{k \in B} x_{iik} = \sum_{j \in V} \sum_{i \in K} x_{ijk} = K, \forall j \in V, \forall k \in K \]

\[
\sum_{i \in K} x_{ijk} = \sum_{i \in K} x_{iik}, \forall j \in V, \forall k \in K \]

\[
t_{ik} + \frac{u_{iik}}{c_{1}} + \frac{\sum_{k \in S} u_{iik}}{c_{1}} + \phi \leq t_{ij}', \forall i \in V, \forall k, k' \in K, k \neq k' \]

\[
0 \leq u_{iik} \leq \sum_{j \in V} \sum_{i \in K} x_{ijk} \cdot q_{ij} \cdot \sum_{k \in B} x_{iik}, \forall j \in C, \forall k \in B \cup S, \forall k \in K \]

\[
f_{ik} = \sum_{k \in S} \left[ \left( f_{ik} - \sum_{l \in B} u_{ikl} \right) \cdot x_{ik} \right] + \sum_{k \in S} \left[ f_{ik} + \sum_{l \in B} u_{ikl} \right] \cdot x_{ik}, \forall j \in V, \forall k \in K \]

\[
0 \leq f_{ik} \leq Q, \forall i \in V, \forall k \in K \]

\[
\theta_{ijk} = 1 - \left( f_{ik} - t_{jk} \right) / T_{life}, \forall i, j \in V, \forall k \in K \]

\[
q_{ij} \leq S_{i}, \forall i, j \in V \]

\[
x_{ijk} \in \{0, 1\}, \forall i, j \in V, \forall k \in K \]

Fig. 3. Schematic diagram of vehicle routing for three types of supply services.
Table 1  
Symbol description. 

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set</td>
<td></td>
</tr>
<tr>
<td>$K$</td>
<td>A set of vehicles, $K = {1, 2, \ldots, k, \ldots, K}$.</td>
</tr>
<tr>
<td>$C$</td>
<td>A set of communities, $C = {1, 2, \ldots, c, \ldots, C}$.</td>
</tr>
<tr>
<td>$B$</td>
<td>A set of markets that provide on-demand food, $B = {1, 2, \ldots, b, \ldots, B}$.</td>
</tr>
<tr>
<td>$S$</td>
<td>A set of a single supplier that provides supportive food, $S = {S}$.</td>
</tr>
<tr>
<td>$V$</td>
<td>A set of places, which include communities, markets, the supplier, and garage, $V = {C, A, B, S}$.</td>
</tr>
</tbody>
</table>

Parameter  

| $Q$  | The capacity of the vehicle. |
| $v$  | The speed of the vehicle (in km/h). |
| $d_{ij}$ | Travel distance from place $i$ to place $j$. |
| $T_{\text{max}}$ | Maximum working hours (in h). |
| $\rho_{\text{rand}}$ | The time required to pick up or drop off a unit food onto/from the vehicle (in h). |
| $\sigma_{ijk}$ | The amount of food demand from market $j$ to community $i$, or the amount of food that supplier $j$ should put out to community $i$. |
| $y_{ijk}$ | The amount of food the market or supplier $j$ can provide to community $i$. |
| $T_{\text{min}}$ | The minimum length of the road travel. |
| $\beta$ | The time that $k$-th vehicle arrives at place $i$. |
| $\alpha$ | A weight coefficient for on-demand food delivery services. |
| $\theta$ | A weight coefficient for supportive food delivery services. |
| $M$ | A huge positive number. |

Variable  

| $x_{ik}$ | A binary variable that defines the route of the $k$-th vehicle. If $x_{ik} = 1$, the $k$-th vehicle departs from place $i$ to place $j$; 0 otherwise. |
| $u_{ik}$ | The amount of food delivered by the $k$-th vehicle from market or supplier $i$ to community $j$. |
| $f_{ik}$ | The amount of food in the $k$-th vehicle when it arrives at place $i$, $f_{ik} = 0$. |
| $\gamma_{ijk}$ | The freshness of food delivered from market or supplier $j$ to community $i$ in the $k$-th vehicle. |
| $t_{ik}$ | The time that $k$-th vehicle arrives at place $i$. |

Objectives function (1) represents maximizing residents' satisfaction with food delivery services. Constraint (2) describes the time relationship between the $k$-th vehicle access places, where $d_{ij}/v$ is the normal road travel time, $\rho_{\text{rand}}(\sum_{j=1}^{B} u_{ik} + \sum_{j=1}^{S} u_{ik})$ is the time used to pick up or drop off the food at place $i$. In order to make the total delivery time of each vehicle similar, and to avoid concentrating the delivery work too much on one vehicle, constraint (3) proposes the maximum working time $T_{\text{max}}$ limit. Constraint (4) indicates that the vehicle’s departure time and departure place. Constraint (5) ensures that each vehicle starts and ends at the garage. Constraint (6) guarantees the conservation of food flow. Constraint (7) is to ensure that there is a time interval between two couriers when they arrive at the same place, to achieve the purpose of contactless between couriers. Constraint (8) indicates that the vehicle only picks up the food needed by the communities on the future route. Eq. (9) guarantees the conservation of food flow on vehicles. Constraint (10) ensures that the amount of food on the $k$-th vehicle should not exceed the capacity of the vehicle. Eq. (11) is a calculation function of food freshness. It is assumed that the freshness of the food will decrease linearly with time during the distribution process. Constraint (12) guarantees that markets and the supplier will not be out of stock. Constraints (13)–(14) are the domains of decision variables.

The maximum working hours $T_{\text{max}}$ and the number of vehicles $K$ are artificially set hyperparameters to provide an interface for human-computer interaction so that the model can obtain schemes that satisfy the administrators for different situations.

5. PEABCTS algorithm

ABC has been applied to solve VRP problems with great success (Karaboga and Gorkemli (2011), Zhang et al. (2014), Sedighizadeh and Mazaheripour (2017), Su et al. (2019), Khan and Maiti (2019)). It is worthwhile to evaluate the performance of the ABC algorithm for solving the vehicle routing problem of the contactless joint distribution. The model of contactless joint distribution is complicated because it allows vehicles to complete the food distribution tasks of both supportive and on-demand supply during the journey, and adds the restriction of contactless between couriers. To this end, based on the enhanced artificial bee colony algorithm, a new artificial bee colony algorithm with a tabu search operator and mechanism of progressive construction solution (PEABCTS) was proposed. The enhanced artificial bee colony algorithm (EABC) is an improved heuristic algorithm proposed by Szeto et al. (2011) based on the ABC algorithm (Karaboga, 2009). The tabu search operator in the onlooker bee stage can enhance the development ability of ABC to solve the distribution scheme efficiently. The mechanism of progressive construction solution allows the algorithm to obtain a high-quality initial solution to reduce the computational time of algorithm.

There are three types of bees in the ABC algorithm, including employed bees, onlooker bees, and scout bees. Different types of bees play different roles in the exploration and development of food sources and search for the optimal solution through multiple iterations. Each iteration consists of three steps. The first step is to match each employed bee with a food source (solution of the problem). It collects information about the food source (the fitness of the solution) and performs a neighborhood search to find a better food source nearby. If a better food source is found, the employed bees will discard the matched food source and remember the better food source. In the second step, onlooker bees track employee bees based on probability selection and use the areas near the corresponding food sources (the neighborhood operator) to find other better food sources to replace the old food sources. In the third step, if the quality of a food source does not improve after several consecutive (minimum) iterations, the employed bee will abandon the matched food source and become a scout bee, which randomly finds a new food source to replace the old food source.

The improvement of the PEABCTS algorithm over the EABC algorithm has three points: First, in the employed bee stage, the crossover operators are used between the old food source and the current optimal food source to increase the possibility of obtaining a better new food source. Secondly, in the onlooker bee stage, using a tabu search operator to search for better food sources has a better optimization ability than the EABC algorithm that only depends on the neighborhood operators. Finally, in order to solve the multi-vehicle routing problem more efficiently, the PEABCTS algorithm uses a method of gradually increasing the number of vehicles to progressively construct a feasible solution of the multi-vehicle routes as a food source.

Based on the above discussion, the PEABCTS algorithm process for solving the vehicle routing problem of contactless joint distribution is as follows:

Step 1: Construct a random route of one vehicle as to the initial food source: $X_{r} \leftarrow L_{K}^{r}, K' = 1$, Where $y = 1, 2, \ldots, Y$, $Y$ is the number of food sources.

Step 2: Each employee bee is matched to a food source, $f(X_{r})$ represents the fitness value of each food source (see Section 5.2).

Step 3: Initialize $A = 0$ and $W_{3} = W_{2} = \ldots = W_{1} = 0$, where $A$ represents the number of repetitions of the entire foraging process; $W_{1}$ represents the number of times the tabu search operator is applied to the food sources $y, y = 1, 2, \ldots, Y$.

Step 4: Repeat the foraging process.

Step 4.1: If the condition of the progressive construction solution is satisfied, a new food source is formed: Construct $(X_{r}, A, A_{\text{max}})$, $Y \rightarrow (X_{r}, y = 1, 2, \ldots, Y)$ (see Section 5.3).

Step 4.2: Employed bee stage

i. Use the crossover operators for each food source $(X_{r}, X_{y}) \rightarrow X_{r}^{1}, X_{y}^{1}$, where $X_{r}$ represents the optimal food source among all existing food sources (see Section 5.4).

ii. If $f(X_{r}^{1}) > f(X_{y})$, then $X_{r} \leftarrow X_{r}^{1}$, where $X_{y}^{1}$ represents $X_{y}^{1}$ or $X_{y}^{2}$.

Step 4.3: Onlooker bee stage

i. Based on the fitness of all food sources, each onlooker bee uses a roulette selection method to select a food source to match.
ii. Use the tabu search operator to perform in-depth optimization on the food sources matched by the onlooker bees: 

Tabu search \( (X_i) \rightarrow X_i \) (see Section 5.5).

iii. If \( f(X_i) > f(X_j) \), then \( X_i \rightarrow X_j, W_j \leftarrow 0 \); otherwise set \( W_j \leftarrow W_j + 1 \).

Step 4.4: Scout bee stage

For each food source, if \( W_j = W_{\text{limit}} \), the food source is modified by one of the neighborhood transformation operators used in the tabu search operator: \( X_i \rightarrow X_j \), then \( X_i \leftarrow X_j, W_j \leftarrow 0, A \leftarrow A + 1 \).

Step 5: Record the current optimal food source \( X_{\text{best}} \) so far.

Step 6: If \( A = A_{\text{max}} \) stop the foraging process and output the optimal global solution.

5.1. The expression of the distribution scheme

The distribution scheme \( Z \) is composed of the sub-distribution schemes of \( K \) vehicles: \( Z = (Z_k), k \in K \), the sub-distribution scheme of the \( k \)-th vehicle consists of a route \( L_k = (l_{ik}), k \in K \), a schedule \( T_k = (t_{ik}) \), and a delivery scheme \( U_k = (u_{ijk}), k \in K \). In order to express that a vehicle can visit the same place repeatedly, the route index range is \( 0 \) times of \( V + 1 \), i.e., \( 0 < i \leq (V + 1) \). The method to calculate the real place index of the \( i \)-th place in \( L_k \) is \( l_{ik} = \left( (l_{ik} - \varsigma) \mod (V + 1) \right) \), where \( \varsigma \) is a minimal positive number, \( \lceil \cdot \rceil \) means an integer up, and \( \mod \) expresses the remainder. For example, \( a \mod b = c \) expresses the remainder of a divided by \( b \) as \( c \). If \( l_{ik} = 0 \), it expresses the index of the garage; if \( 0 < l_{ik} < C \), it expresses the index of one community; if \( C < l_{ik} < C + B \), it expresses the index of one market; if \( l_{ik} = C + B + S \), it expresses the index of the supplier.

For example, For a feasible distribution route \( L_k = (0,6,10,2,7,3,8,0) \) with \( k = 1, B = 2, C = 2, \omega = 2 \). The real place index of \( L_k, l_{ik} = 10 \) is \( l_{ik} = \left( (10 - \varsigma) \mod (2 + 2 + 2) \right) = 4 \), since \( C < 4 \leq C + B, 4 - C = 2 \), so \( L_k, k = 10 \) means the market 2. The correspondences between the route indexes and the real places are shown in Fig. 4. Without exceeding the maximum working hours, the distribution route \( L_k = (0,6,10,2,7,3,8,0) \) means that a vehicle will pick up food from supplier 1 and market 2 after leaving the garage. After that, food is delivered to communities 2 and 1, and then picked up food at market k and the food obtained from market 1 is delivered to community 2. Finally, return to the garage and end the trip. It can be found that the indexes of \( L_k, k \) and \( L_k, k \) both represent community 2, so these repeated indexes can express multiple visits to the same place.

5.2. The fitness of food sources

There are two steps to calculate the fitness of each food source. First, the food source is converted into the actual distribution scheme by Algorithm 1: Conversion \( (X_i, q) \rightarrow (Z, h_i, q) \), and then calculate \( f(X_i) = f(Z) \). The food source fitness function is based on the formula (1) with the addition of correction \( \lambda \):

\[
f(X_i) = \lambda \times \sum_{v \in C} \sum_{c \in C} \left[ \alpha \sum_{d \in D} u_{dik} \theta_{dik} + \beta \sum_{d \in D} u_{dik} \theta_{dik} \right]
\]

(15)

Since vehicles need to go to markets or the supplier to pick up food before they go to communities to deliver food, the route with the second index (the first place visited after the garage) as a community is invalid. Therefore, this invalid route is restricted by setting \( \lambda \). If the second index of \( X_{i, k} \) represents a community, then \( \lambda = 0.1 \); otherwise \( \lambda = 1 \).

5.2.1. The conversion of food sources

First, the food source \( X_i \) needs to be converted into the corresponding distribution scheme \( Z(X_i) \). \( X_i \) consists of candidate routes for \( K \) vehicles: \( X_i = (X_i, 1, \ldots, X_i, k) \). Because the PEABCTS algorithm uses progressive construction solutions (Section 5.3) when the solution has not been entirely constructed, the vehicle number in the food source is smaller than the practical vehicle number: \( K' \leq K \). The first step is to initialize the determined schedule \( Q_{\text{limit}} \). The second step is to derive the delivery food amount of \( k \)-th vehicle \( U_k = (u_{ijk}), \forall i, j \in V \) by the formulas (8)-(10),(12),(14) according to \( L_k \), \( T_{\text{max}} \) and all unfinished distribution tasks \( q = (q_l), \forall i, j \in V \). The calculation of schedule \( T_k \) should be based on \( Q_{\text{limit}} \) to avoid contact with the previous vehicle at the same place. So, the schedule \( T_k \) can be calculated by \( U_k, L_k, Q_{\text{limit}} \) and formulas (2)-(4), (7), (14). Because Algorithm 2: Derivation uses the greedy principle to pick up the food that needs to be delivered, there may be an overloading situation: the food of community \( i \) has been picked up in markets, but the time of arriving community \( i \) exceed the maximum working hours. Overloading causes too much food to be picked up in the markets or suppliers on the early route, resulting in a waste of time and capacity. Overloading will also shorten the practical route length (Algorithm 3: Access). Therefore, in Algorithm 2: Derivation, the practical route length \( h_i' \) needs to be input to avoid overloading. However, setting a smaller practical route length \( h_i' \) may cause the vehicle to end the distribution prematurely before the maximum working hours. The third step is to adjust the practical route lengths gradually. The adjusting method is that the \( h_i' \) is gradually increased based on the practical route length in the first calculation. The condition of \( t_{h_i} \leq T_{\text{max}} \) means the actual practical route length may be larger than \( h_i' \). So, the method is increasing \( h_i' \) until gradually \( h_i' \geq T_{\text{max}} \). The distribution scheme of route length \( h_i = 1 \) to make full use of the maximum working time without overloading. In conclusion, the calculation of the pickup and delivery amount is a process of gradually modifying the practical route length. The fourth step is to update the determined schedule \( Q_{\text{limit}} \) and unfinished distribution tasks after deriving the distribution scheme of a vehicle. Algorithm 1: Conversion shows the pseudocode of the process of transforming the food source into the distribution scheme.

<table>
<thead>
<tr>
<th>( L_{i,k} )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( l_{i,k} )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

Fig. 4. Corresponding transformation of indexes.
Algorithm 1. Conversion.

```
1. U_k = δ_k
2. for i = 1 to K
3. δ_k = δ_k + δ_i
4. Derivation(U_k, δ_k, δ_k) = (U_k', δ_k')
5. Access(U_k', δ_k') = (U_k', δ_k')
6. while δ_k' ≤ T_max
7. δ_k = δ_i + δ_k
8. Derivation(U_k, δ_k, δ_k') = (U_k', δ_k')
9. end
10. δ_k = δ_k - 1
11. (U_k', δ_k') = δ_k
12. (U_k, q) = q
13. end
```

Output: Z = (U_k', δ_k'); i = 1, ..., K

Algorithm 2. Derivation.

```
1. f_k = 0; δ_k = 0
2. for i = 1 to K
3. δ_k = δ_i + δ_k
4. if C < δ_k ≤ C + θ + ε
5. for j = i + 1 to δ_k
6. if 0 ≤ δ_j ≤ C
7. δ_k' = min(δ_k, max(0, δ_j - ε))
8. δ_k' = δ_k' + δ_j - δ_k
9. δ_k = δ_k + δ_j
10. end
11. end
12. if 0 ≤ δ_j ≤ C
13. δ_k' = δ_k' + δ_j
14. δ_k = δ_k + δ_j
15. end
16. for δ_j in δ_k
17. if δ_j > δ_j
18. δ_k' = max[δ_k', max(0, δ_j - ε)]
19. δ_k = δ_k - ε
20. end
21. end
22. end
23. end
24. end
25. end
26. U_k' = (U_k', δ_k'); i = 1, ..., K
```

Output: U_k', δ_k'}

5.2.2. Delivery scheme and schedule

After determining the routes, the corresponding delivery scheme and schedule need to be derived through Algorithm 2: Derivation. The principle of calculating the delivery scheme is: when a vehicle reaches a market, only the amount of food that needs to be delivered to the communities to be visited is picked up until the vehicle is filled. The food for communities that are not within the effective working length of the route will not be picked up. When the vehicle reaches a community, all the food of this community loaded on the vehicle is dropped off. The calculation principle of the schedule is: at the same place, the courier who arrives earlier performs the pickup and delivery operation first, and the courier who arrives later waits for the previous courier to complete the operation and then leaves this place.

Algorithm 2 introduces the waiting time to make the schedule more accurate. We describe the waiting time before the k-th vehicle arrives at place j after place i as \( t_{ij} \) (in h). Fig. 5 describes the calculation rule of waiting time. Vehicles 1 and 2 visit the locations i and i + 1. Vehicle 1* indicates the virtual situation with no waiting time. If there is no waiting time, the time interval between the time \( t_{ij} \) for vehicle 1* to arrive location i and the time for vehicle 2 to leave is not enough. It is believed that there will be contact between these two couriers. Since \( t_{ij} \geq t_{i+1} \), the later courier requires waiting for a period until \( t_{ij} \geq t_{i+1} + \phi \) and then pick up or drop off the foods.

5.2.3. Practical route length

The total number of places that the k-th vehicle can reach within the maximum working hours is called practical route length \( h_k \). Then use Algorithm 3: Access to check whether the distribution process is within the maximum working hours.

Fig. 5. Waiting time in contactless distribution.
Algorithm 3. Access.

Input: $T_0$, $T_1$, $T_{max}$

1. $L_0 = 0$
2. for $i = 1$ to $(V + 1) \cdot \alpha$
3. if $L_0 \geq T_{max}$
4. if $0 < L_0 \leq C$
5. $L_i = L_i + 1$ if $L_i \leq C$
6. $L_i = L_i + 1$
7. end
8. end
9. else
10. $L_i = L_i + 1$
11. end
12. end

Output: $L_i$

5.3. Progressive construction solution

The length of distribution routes of $K$ vehicles in a food source is $[(V + 1) \cdot \omega] \cdot K$. If the length of randomly generated routes is too long, it will cause subsequent neighborhood transformation to search in a larger domain, which is not conducive to the convergence of the algorithm. Because of Hypothesis 1: vehicles have the same capacity, the PEBACTS algorithm uses a progressive approach to gradually add vehicles into the food source until the number of vehicles is finally needed. The initial food source only contains one vehicle. Subsequently, according to the increase in the number of iterations, the new food source adds a randomly generated route of the new vehicle based on the corresponding old food source $X_k$ (Algorithm 4: Construct). This method of progressively constructing new food sources is to gradually expand the solution domain while ensuring that the new food sources always have good fitness and to achieve the purpose of improving the search efficiency of the algorithm. Especially in the case of a large number of vehicles, this method avoids the algorithm wasting a lot of calculation time on food sources of the algorithm. Especially in the case of a large number of vehicles, this method of progressively constructing new food sources is to gradually expand the solution domain while ensuring that the new food sources always have good fitness, and improves the efficiency of the algorithm.


Input: $X_k$, $A_k$, $A_{max}$, $F$

1. $L = (L_0, L_1, \ldots, L_0) = X_k$
2. for $y = 1$ to $F$
3. $C_y = K \cdot (K + 1)/2$; $S_y = K(K + 1)/2$
4. if $a > C_y/K$ then
5. Randomly generate route $L_{y+1}$
6. $Y = L \cup L_{y+1}$
7. end
8. end
9. end

Output: $X_y = 1, 2, \ldots, F$

5.4. Crossover operator

The EABC uses the neighborhood operators to find new food sources in the employed bee stage and compare them with the original food sources. Neighborhood operators have no direction, so new food sources may not be better than old ones. In order to make new food sources better, this paper uses crossover operators to search for new food sources based on the current optimal food source. Since OX and PMX are two types of crossover operators with good results for solving routing problems (Larranaga et al., 1999), this paper uses OX and PMX (shown in Fig. 6) as crossover operators. Cross the route $X_{k}$ of each vehicle in the food source $X_k$ with the route $X_{k'}$ of each vehicle in the current optimal food source $X_{o}$, respectively, to generate two new food sources $X_{k'}^1$ and $X_{k'}^2$. After comparing the two new food sources with the old food source, the better food source is retained. Because the first and last indexes of the route are garages, the crossover operator does not include the first and last indexes. In addition, in order to improve the efficiency of crossover operator, the position of the first index of the crossover operator cannot be outside the practical route length, i.e., $1 < M2 < (V + 1 - \omega); 1 < M1 < h_k$, where $h_k$ represents the practical route length of $X_{k}$. Otherwise, the crossover operators will not change the actual distribution scheme.

5.5. Tabu search operator

In order to increase the ability of the artificial bee colony algorithm to search locally in the later period, a tabu search operator is embedded in the onlooker bee stage. The tabu search algorithm (Glover, 1986) is an extension of the local neighborhood search. The optimization ability of the tabu search operator is stronger than the neighborhood of the onlooker bee stage in the original EABC. The tabu search operator uses formula (15) as the adaptation function. The tabu length is $2^*$.

Algorithm 5. Tabu search.

Input: $X_k$

1. for $k = 1$ to $K$
2. $L_k = X_k$
3. $L_{k_{min}} = \min L_k^* [L_{k1}, \ldots, L_{kN}]$.
4. while $W < K$
5. Neighborhood transformation operators: $(L_{k_{min}}, h_k) \rightarrow (L_{k_{i}}, n = 1, \ldots, N)$
6. Find the optimal solution $L_{k_{i}}$ and optimal solution $L_{k_{i}}^*$ which is not in the Tabu list from $L_{k_{min}}$.
7. if $f(L_{k_{i}}) > f(L_{k_{i}}^*)$
8. $L_{k_{min}} = L_{k_{i}}^*$; $W = W + 1$
9. else
10. $L_{k_{min}} = L_{k_{i}}$; $W = W + 1$
11. end
12. Update Tabu List
13. end
14. $X_k = L_{k_{min}}$
15. end

Output: $X_k = (X_k), k = 1, 2, \ldots, K$

Neighborhood transformation operators choose one of three common neighborhood operators: random swaps of subsequences, random insertions of subsequences, and reversing subsequences (Fig. 7). Before the neighborhood transformation operators, five parameters $M1$, $M2$, $\Delta 1$, $\Delta 2$, and $k$ need to be randomly generated to determine the specific index position of the neighborhood transformation, where $M2 > M1, 0 < M2 - M1$, and $1 < k < K$. $M1$ represents the starting position of the neighborhood transformation; $M2$ represents the end position of the neighborhood transformation; $\Delta 1$ represents the length of the first sequence; $\Delta 2$ represents the length of the second sequence; $k$ represents the sequence index of the vehicle that performs the neighborhood transformation operators. Like the
crossover operators, the neighborhood transformation operators do not include the first and last indexes of the route sequence, and the positions of the first indexes of the crossover operators are within the practical route length. Fig. 5 shows the process of neighborhood transformation operators. Only the positions of M1, M2, Δ1, and Δ2 in random swaps of subsequences are marked in Fig. 5. The five parameters of the other neighborhood transformation operators are the same as that in random swaps of subsequences: M1 = 3, M2 = 9, Δ1 = 1, Δ2 = 2. The specific process of each neighborhood transformation operator is detailed below.

1) Random swaps of subsequences

The starting point of the first sequence is i = M1, its length is Δ1; the terminal point of the second sequence is j = M2, and its length is Δ2. Then the positions of the two sequences are reversed, i.e., (2,3) is swapped with (6,7,8) in Fig. 7(a). The positions of the other points are unchanged. In the condition of Δ1 = Δ2 = 0, random swaps of subsequences become random swaps of points. Just swap the point of i = M1 with the point of j = M2, i.e., (2) is swapped with (8) in Fig. 7(b). The positions of the other points are unchanged.

2) Random insertions of subsequences

The starting point of the subsequence is j = M2, its length is Δ2, and it is inserted into the position of i = M1, i.e., (6,7,8) is inserted into the position of Ly,k(3) in Fig. 7(c). The positions of the other points are unchanged. In the condition of Δ1 = Δ2 = 0, random insertions of a subsequence become random insertions of points. Just insert the point of j = M2 to the position of i = M1, i.e., (8) in Fig. 7(d) insert to the position of Ly,k(3). The positions of the other points are unchanged.

3) Reversing subsequences

The terminal point of the subsequence is selected as j = M2, and its length is Δ2. Then the sequence is reversed, i.e., the (6,7,8) is reversed to (8,7,6) in Fig. 7(e). The positions of the other points are unchanged.

---

**Fig. 6. Crossover operators.**

**Fig. 7. Neighborhood transformation operators.**
The tabu search operator is different from the ordinary tabu search algorithm in three aspects:

1) The tabu search operator takes \( X_p \) as input instead of the randomly generated initial solution.

2) Randomly select \( X_{p,k} \) in \( X_p \) by parameter \( k \) before performing neighborhood transformation operators, which can shorten the time required to perform neighborhood transformation operators in \( X_p \) directly.

3) The tabu length, the number of candidate solutions, and the termination condition all have adaptive mechanisms, which can enhance the effectiveness when solving problems of different scales.

6. Numerical study

The methodology proposed in this paper is applied to nine numerical simulation cases with different scales. In the following subsections, the parameter settings of numerical simulation will be described, and the effectiveness of the algorithm will be analyzed. After that, the distribution effect of contactless joint distribution will be analyzed. Finally, the effectiveness of the algorithm will be analyzed. After that, the distribution parameter settings of numerical simulation will be described, and the effectiveness of distribution are discussed.

6.1. Numerical simulation

Firstly, three kinds of regions with different characteristics were randomly generated in a square area of 9km\(^2\): a region with a few large communities (L), a region with many small communities (S), and a mixed region with both large communities and small communities (M). As shown in Fig. 8, the garage (yellow pentagon) is in the center of the region, the supplier (blue rhombus) is located near the garage, the communities (red squares), and markets (green circles) are evenly located throughout the region. The sizes of communities and the sizes of markets are random numbers. For example, the sizes of communities in the case of L1 are random numbers between 5 and 10. With the three ratios of supportive supply for each region, the nine numerical cases are generated (see Table 2). We assume that the total food demand is constant, so if the supportive supply of food is abundant, residents' demand for on-demand food will be small. For example, if the ratio of supportive supply is 0.2, it means that the supportive supply can meet 20% of the food demand of the residents, and the remaining 80% of the demand is on-demand food. In numerical simulations, in order to ensure that the communities and markets in different-scales regions have the same size, the number of communities \( \times \) the average size of communities = constant value (75); the number of markets \( \times \) the average size of markets = constant value (6).

Because each community has a certain area, the distances between the communities and the markets cannot be too small. The food demand from markets \( \times \) the average size of markets = constant value (6).

For example, if the ratio of supportive supply is 0.2, it means that the supportive supply can meet 20% of the food demand of the residents, and the remaining 80% of the demand is on-demand food. In numerical simulations, in order to ensure that the communities and markets in different-scales regions have the same size, the number of communities \( \times \) the average size of communities = constant value (75); the number of markets \( \times \) the average size of markets = constant value (6).

In order to compare the distribution effect of each case, the total demand of residents in each case needs to be controlled by formula (17) to be a fixed value, 500.

\[
q_j = \frac{q_{ij}^*}{500}, \forall i \in C, j \in B \cup S \tag{17}
\]

Set other parameters as \( \alpha = 1, \beta = 2, Q = 50, T_{\text{max}} = 3(\text{h}), r_{\text{load}} = 1/60(\text{h}), v = 10(\text{km/h}), \phi = 1/6(\text{h}), T_{\text{life}} = 24(\text{h}), K = 3, c_j = 0, \forall i, j \in V \). When the value of \( \omega \) is not set properly, it will result in an inefficient distribution scheme, so it is necessary to dynamically set \( \omega \) according to formula (18) so that the vehicle can always visit the places within \( T_{\text{max}} \), without causing too many useless points in \( L_{x,k} \).

\[
\omega = \frac{\sum_{i \in C, j \in B} q_{ij}^*}{Q \cdot v} \times 1.2 \tag{18}
\]

6.2. Algorithm performance analysis

Our computational experiments are carried out on a laptop powered by an Intel Core i5 CPU @ 2.6 GHz with a total RAM of 4 GB. In order to prove that PEABCTS can solve the contactless joint distribution model efficiently,

![Fig. 8. The three regions with different characteristics.](image-url)
the performance of PEABCTS is compared with EABCTS, PEABC, EABC, and TS. The information of each algorithm is shown in Table 3.

The neighborhood operators used in EABCTS, PEABC, EABC, and TS are the same as the neighborhood operators used in the tabu search operation (Section 5.5). The computational time of every algorithm in different cases is approximately consistent with PEABCTS (Table 4), running for 20 times. Fig. 9 summarizes the results of nine numerical cases for five algorithms through box plots. In each box, the junction of light red and dark red indicates the median value, the edges are the 25th and 75th percentiles, the whiskers extend to the most extreme non-outlier values, and outliers are plotted individually as black points. The values of upper limits are marked. The "Gap" is the gap between the 25th and 75th percentiles, which is calculated by the formula (19).

\[ \text{Gap} = P_{75\%} - P_{25\%} \]

In Fig. 9, we can find that PEABCTS has obtained the best upper limit of five numerical cases, the best median and the best lower limit of total nine numerical cases. By comparing the five algorithms, we can find that the calculation results of EABC and TS are relatively poor. However, EABCTS combines the advantages of both, which has achieved better upper limits, medians, and lower limits than EABC and TS in the nine numerical cases. After comparing PEABCTS and EABC, it can be found that the mechanism of progressive solution construction can effectively improve the development ability of the algorithm, but it will increase the Gaps in the results. However, comparing PEABCTS and EABCTS, the mechanism of progressive solution construction can reduce the Gaps in the results. After embedding the tabu search operator in PEABC, the convergence ability of PEABCTS is further improved, and a smaller Gap than PEABC is obtained.

In conclusion, embedding the tabu search operator or the mechanism of progressive construction solution can raise the upper limit of results and the Gap. Only PEABCTS with both the tabu search operator and the mechanism of progressive construction solution can improve the upper limit of results and reduce the value of Gap when solving the vehicle routing problem of contactless joint distribution service.

6.3. Contact frequency analysis

This section compares the contactless joint distribution service with two distribution services of supportive supply and on-demand supply and analyzes the impact of the contactless joint distribution service on reducing the contact frequency. For simplicity, we write “the independent distribution service” to refer to the two distribution services of supportive supply and on-demand supply in the following sections. The supportive food distribution model and the on-demand food distribution model are established. The constraints and objective functions of the supportive food distribution model and the on-demand food distribution model are consistent with the contactless joint distribution model, except for the following two differences:

1) Since there are no contactless constraints in the supportive supply service and the on-demand supply service, the constraint (7) is removed.
2) The supportive food distribution model adds \( \sum_{j \in V} q_{ijk} = 0, \forall i, j \in B, \forall k \in K \) to prohibit vehicles from distributing on-demand food. The on-demand food distribution model adds \( \sum_{j \in V} q_{ijk} = 0, \forall i, j \in S, \forall k \in K \) to prohibit vehicles from distributing supportive food.

In the independent distribution service, a vehicle can only choose one between the supportive supply and on-demand supply for delivery. Therefore, the vehicles are divided into two parts to complete the distribution tasks of on-demand supply and supportive supply respectively (formulas (20), (21)).

The number of vehicles allocated to supportive supply
\[ \text{Delivery Efficiency} = \frac{P}{D} \]

Supply Rate
\[ = \frac{\sum_{i \in C} \sum_{j \in C} q_{ijk}}{\sum_{i \in C} \sum_{j \in C} v_{ijk}} \]

The contactless joint distribution service and the independent distribution service are respectively applied to nine numerical cases, and each numerical case is calculated for 10 times to obtain the distribution function with the optimal objective function (Table 5). The total distance traveled by the vehicles, and the contact frequency in the distribution scheme is calculated. The contactless joint distribution service has a higher supportive rate. In other words, the contactless joint distribution service can focus more on the delivery services with higher weight coefficient, while the independent distribution service does not have such advantages. The size of the supportive rate depends on the number of vehicles allocated to the supportive supply. In nine numerical cases, the contactless joint distribution service can increase both the supportive rate and supply rate.

First, comparing the average value of the results of nine numerical cases, it can be found that the contactless joint distribution service has higher residents’ satisfaction, shorter traveling distance, and leads to higher delivery efficiency, which increased 60.6% on average than the independent distribution service. In the same case, the independent distribution service has a higher contact frequency, which is 6.78 on average. The contact frequency of case M3 is 12 times, which indicates that each courier has contacted 4 times within three hours of working time. During the transmission of the COVID-19 virus, the administrator should avoid such high contact frequency.

"L1–L3" means the average values of the results of three numerical cases L1, L2, and L3. The same applies to “M1–M3” and “S1–S3”. It can be found that for a region with a few large communities, though the residents'
satisfaction is high, the independent distribution service will generate more contact frequency between the couriers. Therefore, it is necessary to use the contactless joint distribution service to avoid contact. For three types of regions, the contactless joint distribution service can improve residents' satisfaction, the supply rate, the supportive rate, and the delivery efficiency. It is proved that the contactless combined distribution model can be applied to regions with different kinds of communities.

6.4. The role of human-computer interaction

In order to verify that the interface parameters $K$ and $T_{\text{max}}$ of human-computer interaction can help administrators to formulate suitable distribution schemes in many different situations. In this section, different $K$ and $T_{\text{max}}$ values are selected for comparison in the case of M2. Running 10 times for each situation to get the optimal distribution scheme. Set the number of iterations $A_{\text{max}} = 20$. First set $T_{\text{max}} = 3$, and choose six situations of the number of vehicles $K = \{1,2,3,4,5,6\}$. Fig. 10(a) indicates that as the number of vehicles increases, resident's satisfaction (P) increases. However, the larger the number of vehicles, the smaller the improvement in residents' satisfaction. The delivery efficiency of each vehicle gradually decreases as the number of vehicles increases. Then set $K = 3$, and choose six situations of the maximum working hours $T_{\text{max}} = \{2,2.5,3,3.5,4,4.5,5\}$, the trend of the results is similar to the results of the different number of vehicles: with the increase of the maximum working hours, the delivery efficiency of each car decreased gradually (Fig. 10(b)). There are two inflection points of delivery efficiency, both in Fig. 10(a) and (b). Therefore, two vehicles and a maximum working time of four hours are a suitable choice in the case of M2.

It can be seen from Fig. 10(a) that the supply rate increases as the number of vehicles increases, but the trend of increasing the support rate gradually slows down with the increase in the number of vehicles. As can be seen from Fig. 10(b) that the supply rate increases slowly with the increase in the maximum working hours, while the supportive rate stabilizes at a high level. As a result, if administrators need to increase the supply rate, it is recommended to increase the number of vehicles instead of extending...
the maximum working hours; if the supportive rate is to be increased, it is recommended to extend the maximum working hours instead of increasing the number of vehicles.

7. Conclusions

In order to address the issue of food distribution during the enclosed management period of gated communities under the influence of COVID-19, this paper proposes a new contactless joint distribution service based on two independent services: supportive supply service and on-demand supply service. In order to solve the contactless distribution model, a PEABCTS algorithm is developed in this paper. Compared with five algorithms, it is proved that the mechanism of progressive construction solution and the tabu search operation can improve the efficiency of the enhanced artificial bee colony algorithm. Applying to three kinds of regions with different scales and three different ratios of supportive supply, the PEABCTS algorithm shows strong adaptability.

Through the comparative experiments of the two independent distribution models and the contactless joint distribution model, it is found that the contactless joint distribution service can reduce the contact frequency between the couriers, reduce the risk of infectious diseases, and finally protect the safety of the couriers during COVID-19 pandemic. In addition, the contactless joint distribution service can improve residents' satisfaction, supportive food distribution rates, food distribution rates and distribution efficiency.

Set the parameters of the maximum working time and the number of vehicles as a human-computer interaction interface. These two parameters

Table 5

The results of two kinds of services.

<table>
<thead>
<tr>
<th>Case</th>
<th>Contactless joint distribution service</th>
<th>Independent distribution service</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P Supply rate</td>
<td>Supportive rate</td>
</tr>
<tr>
<td>L1</td>
<td>342.6 52%</td>
<td>97%</td>
</tr>
<tr>
<td>L2</td>
<td>438.5 51%</td>
<td>100%</td>
</tr>
<tr>
<td>L3</td>
<td>514.8 56%</td>
<td>85%</td>
</tr>
<tr>
<td>M1</td>
<td>328.8 51%</td>
<td>91%</td>
</tr>
<tr>
<td>M2</td>
<td>426.3 53%</td>
<td>91%</td>
</tr>
<tr>
<td>M3</td>
<td>509.7 53%</td>
<td>88%</td>
</tr>
<tr>
<td>S1</td>
<td>265.2 38%</td>
<td>84%</td>
</tr>
<tr>
<td>S2</td>
<td>369.5 40%</td>
<td>93%</td>
</tr>
<tr>
<td>S3</td>
<td>427.0 46%</td>
<td>72%</td>
</tr>
<tr>
<td>L1-L3</td>
<td>431.96 53.3%</td>
<td>94.3%</td>
</tr>
<tr>
<td>M1-M3</td>
<td>421.60 52.4%</td>
<td>90.0%</td>
</tr>
<tr>
<td>S1-S3</td>
<td>353.89 41.2%</td>
<td>82.7%</td>
</tr>
<tr>
<td>Mean</td>
<td>402.49 48.95%</td>
<td>89.02%</td>
</tr>
</tbody>
</table>

Fig. 10. The impacts of two interface parameters.
can help administrators to develop a more effective distribution scheme for different environments and conditions based on the occurrence of inflection points. Through numerical experiments, it is found that if the administrators need to increase the supply rate, the number of vehicles can be increased; if the supportive rate needs to be increased, the maximum working time can be extended.

The contactless joint distribution service is a crucial issue for the daily food supply of closed gated communities in response to the COVID-19 epidemic and can be extended to other major public health emergency application scenarios. Future research can develop in several directions. The research and development of the dynamic joint distribution model of the changes in the demand of residents in closed communities is an important topic. Another direction is to consider improving food distribution in the communities.

Declaration of competing interest

None.

Acknowledgments

Our deepest gratitude goes to the anonymous reviewers for their careful work and thoughtful suggestions, which have helped us to substantially improve this paper.

Funding

This work was supported by the National Natural Science Foundation of China (Grant No. 50908235).

CRediT authorship contribution statement

Dawei Chen: Methodology, Software, Validation, Formal analysis, Data Curation, Writing - original draft, Visualization; Shuangli Pan: Investigation, Validation, Formal analysis, Data Curation; Qun Chen: Conceptualization, Resources, Writing - review & editing, Supervision, Project administration, Funding acquisition; Jiahui Liu: Software, Formal analysis, Investigation, Data curation, Visualization.

References


