Research Article

Research on Coordination and Optimization of Order Allocation and Delivery Route Planning in Take-Out System

Guofeng Sun,1 Zhiqiang Tian,1 Renhua Liu,1 Yun Jing,2 and Yawen Ma2

1 School of Traffic and Transportation, Lanzhou Jiaotong University, Lanzhou 730070, China
2 School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China

Correspondence should be addressed to Zhiqiang Tian; tianzq@mail.lzjtu.cn

Received 6 April 2020; Revised 15 June 2020; Accepted 20 June 2020; Published 23 July 2020

Academic Editor: Purushothaman Damodaran

Copyright ©2020 Guofeng Sun et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper studies the take-out route delivery problem (TRDP) with order allocation and unilateral soft time window constraints. The TRDP considers the order allocation and delivery route optimization in the delivery service process. The TRDP is a challenging version of vehicle routing problem. In order to solve this problem, this paper aims to minimize the total cost of delivery, builds an optimization model of this problem by using cumulative time, and adds time dimension in order allocation and path optimization dimensions. It can not only track the real-time location of delivery personnel but also record the delivery personnel to perform a certain task. The main algorithm is the dynamic allocation algorithm designed from the perspective of dispatch efficiency, and the subalgorithm is the improved genetic algorithm. Finally, some experiments are designed to verify the effectiveness of the established model and the designed algorithm, the order allocation and route optimization are calculated with/without the consideration of traffic jam, and the results show that the algorithm can generate better solution in each scene.

1. Introduction

At present, the take-out market in China continues to develop at an extraordinary speed, and the take-out platforms are competing in all aspects, such as high-quality supply and delivery experience. The COVID-19 has begun to erupt in all parts of the world and is spreading rapidly since the end of 2019. China, as one of the countries affected earlier by the epidemic, has adopted the measures of home quarantine during the fight against epidemic diseases, which plays a vital role in overcoming the epidemic. During the period of home quarantine, a large number of purchase and distribution of life materials and medicines were completed through the take-out. The mode of sale and delivery provides certain material support for residents, effectively alleviates the risk of infection of residents going out, and makes the best effect of blockade (isolation at home). It has been proved to be effective by practice in China to use a large number of daily necessities delivered by take-out staff to urban and rural residents during the epidemic period, which is worthy of reference for other countries that are affected by the epidemic and intend to adopt blockade (home isolation). Therefore, it is of great significance to research and apply the order (food, medicine, and other daily necessities) delivered by take-out staff.

Delivery timeliness and on-time rate, as important indicators of take-out delivery, are the core competitiveness of the delivery platform. Generally speaking, the most direct way to improve the delivery efficiency and punctuality rate of users is to equip more delivery personnel and expand the scale of transportation capacity. However, it also means the increase of distribution cost. Therefore, on the one hand, the delivery platform should pursue a good delivery experience; on the other hand, it is constrained by the human cost of distribution. How to improve the delivery rules, optimize the delivery route, and reduce the delivery cost while improving customer satisfaction has become a hot issue. In the current delivery problem, depending on the experience of delivery personnel to choose the delivery order and delivery path is easy to cause secondary delivery, which not only reduces the delivery efficiency but also increases the delivery cost and customer dissatisfaction, and less consideration is given to
the real-time distribution of delivery orders. Therefore, how to improve the delivery efficiency and customer satisfaction is very important for delivery companies, businesses, and customers.

In recent years, the issue of food delivery has become a hot topic of research. Li et al. [1] studied the task assignment and scheduling of food delivery from the perspective of game theory, aiming at maximizing the income of the dispatcher; Harman [2, 3] researched and developed the take-out order automatic management system, which realizes the interaction between customers and merchants through the order system and provides customers with the additional sales option service during the ordering process; Liu et al. [4, 5], using the spatial crowdsourcing technology, constructed the urban food distribution network and platform, and a new way to support delivery by taxi is studied. The optimization goal is to reduce the number of delivery by taxi, so as to ensure a higher income for the drivers participating in the delivery. The simulated annealing algorithm is designed for optimization, and the experiment has achieved good results in Chengdu, China. Liao et al. [6] proposed a multiobjective scheduling model for order combination and delivery path optimization in the process of meal delivery, in order to maximize customer satisfaction and balanced utilization of dispatchers in the region, and designed a two-stage solution algorithm composed of nondominated sorting genetic algorithm II (NSGA-II) and adaptive large neighbourhood search (ALNs). Jiang et al. [7] designed a dispatch system based on big data of take out logistics. The system can optimize the number of regional dispatchers, improve the efficiency and income of dispatchers, and increase customer satisfaction. The routing optimization problem in take-out delivery problem can be regarded as a variant of Vehicle Routing Problem (VRP). The general vehicle routing problem (VRP) refers to a certain number of customers, each of whom have different quantity of goods demand.,Q_hey distribution center refers to a certain number of customers, each of whom have different constraints, to achieve goals such as the shortest distance, lowest cost, and shortest time consumption under certain constraints [14]. However, the delivery problem does not have a unique delivery center. Businesses and customers correspond to each other, and there are soft time window requirements. After the delivery, the delivery personnel do not need to send the foods to the corresponding customers immediately. After the delivery, they do not need to return to the delivery starting point. They can continue the delivery with the current position as the starting point.

The research of vehicle routing problem has been relatively mature, so now it is basically focused on the problem with deformation and algorithm research. Chen et al. [15] studied different types of vehicle routing problems with virtual warehouse and time window and proposed a two-stage heuristic algorithm based on tabu search; Chen and Shen [16] proposed a hierarchical particle swarm optimization algorithm including the inner layer and outer layer for the impact of the location of the parking lot on the transportation cost, using the determined location of the depot to obtain the best establishment of the depot Location and the corresponding optimal vehicle path, in which the inner algorithm obtains the optimal vehicle path and the outer algorithm obtains the location of the parking lot. Liu et al. [17] in order to solve the problem of resource allocation and carbon emission in green vehicle path optimization, proposed a hybrid quantum immune algorithm based on cloud model (c-hqia), which can effectively improve the distribution efficiency and reduce the distribution demand Ben. Li et al. [18] comprehensively considered the vehicle routing and fleet size decision-making problems of various vehicle types when studying the multistage heterogeneous fleet scheduling problem, transformed the problem into a mixed integer programming model, and proposed a new heuristic algorithm based on greedy algorithm and simulated annealing algorithm. Ma and Sun [19] proposed a mutation ant colony algorithm for solving the freight vehicle routing problem, which has a better performance Martinovic et al. [20] proposed a new vehicle
routing problem considering single item pickup and delivery and proposed an iterative improved simulated annealing algorithm based on greedy random initial solution.

Although the research on the deformation and algorithm of the vehicle routing problem has been relatively mature, for the delivery path optimization problem, we need to consider the constraints of one-sided soft time window and the special requirements of first take and send; for the regional delivery system, we must consider the distribution between the delivery order and delivery personnel. Based on this, this paper establishes a delivery path optimization model based on cumulative time and designs the shortest path order allocation algorithm and the path optimization algorithm based on the improved ant colony algorithm.

The major contributions of this study are as follows: (1) for allocation and dispatch path optimization modelling, the cumulative time modelling method is used to establish the order allocation and dispatch path optimization model; (2) a dynamic optimization algorithm is designed to solve the model in this paper.

Section 2 describes the problem of order allocation and path optimization in the take-out system. Section 3 is the construction of the optimization model. Section 4 is the algorithm design. Section 5 summarizes the research results, and Section 6 discusses the future research.

2. Problem Statement and Notations

2.1. Problem Statement. The problem of order allocation and delivery path optimization in the take-out system is that when the take-out order is generated, the order is assigned to a specific courier according to the current location and distribution status of all couriers. After receiving the new task, the courier starts from their current location and goes to the merchant to pick up the goods under the condition of time window constraint and then sends it to the corresponding customer, or continues to the next merchant node to pick up goods and then to complete the delivery task at the same time. Generally speaking, a one-sided soft time window constraints for the courier get to the merchant to pick up the meal and to the customer’s home to deliver the order, as shown in Figure 1.

In the order allocation stage, we need to consider the impact of the orders already dispatched to every courier and allocate the order tasks from the aspects of delivery efficiency and service ability of all couriers. When a new order is designated to a certain courier, the originally planned delivery route of this courier may be affected rather than the most effective one. As a result, a rescheduling phase is necessary to obtain the optimal path based on current situation, which reflects the dynamic coupling characteristic between order allocation and delivery route optimization in the take-out delivery problem. In the delivery route optimization stage, it is aimed to find an optimal route with the minimum generalized cost, including the travel distance, the waiting cost, and the time penalty cost of all couriers. In Figure 2, a small example is used to illustrate the interaction process of order allocation and route optimization in the delivery problem.

Figure 2 shows a simplified delivery problem. The node pairs (1, 2), (3, 4), and (5, 6) are the orders that have been generated before and assigned to the courier \( v \) according to certain distribution rules, in which the nodes with odd number correspond to the merchant node while the nodes with even number correspond to the customer node. As can be seen from the comparison, with the emergence of new order (7, 8) at a certain time, the optimal delivery route of courier \( v \) changed from (1-2-3-4-5-6) to (1-2-3-7-4-5-8-6). When the system has more than one take-out couriers, the result of order allocation and route optimization are changed dynamically.

According to the characteristic of this problem, the set of merchant and the set of customer both have a unilateral soft time window constraint, that is, the courier can arrive at the merchant or serve the customer early but not late than the expect time window, otherwise a penalty cost associated to the delay time will be used to affect the generation of delivery route.

Without losing generality, this paper makes the following assumptions:

(1) The number of couriers inside the service region is known and fixed, which can meet the demand of order dispatch
(2) All orders are compatible with any courier, that is, each order can be dispatched to any courier
(3) The delivery interruption caused by vehicles and couriers is not considered

2.2. Notations, Variables, and Formulation

(1) General Notations

\[ V \]: set of couriers
\[ M \]: set of businesses
\[ P \]: set of customers
\[ O \]: set of take-out couriers starting location nodes
\[ K \]: set of take-out orders
\[ N \]: set of space-time nodes
\[ A_p \]: set of take-out delivery optimizes arcs in space-time networks
\[ v \]: index of courier
\[ m \]: index of business
\[ p \]: index of customer
\[ k \]: index of order
\[ \Psi_p \]: set of service arc couriers to customers \( p \)
\[ (i, t), (j, s) \]: space-time nodes, \( (i, t), (j, s) \in N \)
\[ (i, t; j, s) \]: a space-time arc from head node \( (i, t) \) to tail node \( (j, s) \), \( (i, t; j, s) \in A \)
\[ c \]: penalty coefficient of courier arriving late at business node or customer node
\[ d_{i,t; j,s} \]: the distance from node \( (i, t) \) to node \( (j, s) \)
\[ l_j \]: the latest arrival time specified by node \( j \)
(2) Space-Time Node Sets
- \(N^k_D\): set of take-out orders leaving the node
- \(N^k_A\): set of take-out orders arriving at the node
- \(N^V_D\): set of courier leaving the node
- \(N^V_O\): set of origin location nodes of courier

(3) Other Declaration Symbols
- Area: the search area is composed of the location of the order businesses node and the radius of 3 km
- \(D^k_{mp}\): the distance between the couriers and the businesses node in the search area of order \(k\)
- \(D^k_{mp}\): the distance between businesses node \(m\) and customer node \(p\) in order \(k\)
- \(S_v\): the distance completes orders of courier \(v\)
- \(C_v\): capacity limitation of courier \(v\).

(4) Decision Variables
- \(x^v_{i,t,j,s}\): 1 if the courier goes from node \((i,t)\) to node \((j,s)\); 0 otherwise.
3. Optimization Model Construction of Delivery Coordination

In order to optimize the delivery route of take-out, it is necessary to consider the distribution of take-out tasks and the optimization of the delivery route. Set up the take-out delivery time-space network, as shown in Figure 3, and use the cumulative time to build the model [21], so as to accurately express the take-out delivery route optimization problem with two sets of variables.

In order to represent specific nodes with temporal and spatial attributes, we define them as \((i, t)\) and \((j, s)\), which represent the departure node and arrival node of the order, respectively, and define a node set \(N = N^{KD} \cup N^{KA}\) based on delivery orders, as shown below:

(1) Delivery order departure node \((i, t) \in N^{KD}\), where \(i\) and \(t\) represents the starting position and time of a particular take-out order \(k\), respectively, and satisfies \(\lambda(i, t) = k\), indicating the coupling relationship between the departure node \((i, t)\) and the take-out order \(k\).

(2) Take-out order arrival node \((j, s) \in N^{KA}\) denotes that, for a particular take-out order \(k\), \(j\) and \(s\) represent the arrival location and time of take-out order \(k\), respectively, and similarly satisfy \(\lambda(j, s) = k\).

(1) Objective Function. The objective of this optimization model is to minimize the generated total cost of take-out delivery, including travel cost and time penalty cost, that is, the objective function can be described as:

\[
\min Z_1 = \sum_{v \in V} \sum_{(i, t, j, s) \in A_v} d_{ij} x_{ij}^v + c \sum_{v \in V} \sum_{(i, t, j, s) \in A_v} \max(s - l_{ji}, 0).
\]

(1)

(2) Delivery Start Node Constraints. From the current location, the first node that the take-out delivery courier visits must be the merchant node:

\[
\sum_{(i, t, j, s) \in A_v} x_{ij}^v = 1, \quad i \in O, j \in M, \forall v \in V.
\]

(2)

(3) Delivery End Node Constraints. When the take-out delivery courier visits his last customer, it can be considered that the take-out dispatch process is completed and there is no need for him to return to the starting point or the merchant nodes:

\[
\sum_{(i, t, j, s) \in A_v} x_{ij}^v = 0, \quad i \in P, j \in O, \forall v \in V.
\]

(3)

(4) Service Frequency Constraints. In a task set, for the same courier, each customer is served exactly once:

\[
\sum_{(i, t, j, s) \in A_v} x_{ij}^v = 1, \quad \forall p \in P, \forall v \in V.
\]

(4)

(5) Order Task Coverage Constraints. Each order task can be delivered by only one take-out delivery courier. Task coverage constraints can be represented by route dispatch variable \(x_{ij}^v\) and order dispatch variable \(y_k^v\):

\[
\sum_{(i, t, j, s) \in K, \lambda(i, t) = k} x_{ij}^v = y_k^v, \quad \forall k \in K, \forall v \in V,
\]

(5)

(6) Order Distribution Constraints. For each order, only one take-out delivery courier can be assigned:

\[
\sum_{v \in V} y_k^v = 1, \quad \forall k \in K, \forall v \in V.
\]

(6)

(7) Variable Constraints. The take-out delivery couriers choice of each route is a 0-1 variable, and whether to assign order \(k\) to courier \(v\) is also a 0-1 variable:

\[
x_{ij}^v \in \{0, 1\} \quad (i, t, j, s) \in A_v,
\]

\[
y_k^v \in \{0, 1\}, \quad \forall k \in K, \forall v \in V.
\]

(7)

4. Solution Algorithms

The delivery order allocation and dispatch route optimization studied in this paper is a dynamic process. In the solution stage, it is necessary to determine the allocation result of the subsequent new orders according to the assigned orders and their completion status of the delivery courier, as shown in Figure 4. Based
on this, we design an order dynamic allocation algorithm embedded in a path optimization subalgorithm.

First, orders generated early are assigned to couriers with the consideration of distance between the courier and the merchant and the customer, as well as the delivery ability of the couriers, and the current best delivery route is calculated according to all orders allocated to each courier. Then, the new order allocation is carried out based on the former order allocation result, and finally the optimal route after adding the new order is recalculated and this process will be repeated until all orders are allocated to the corresponding courier.

The allocation of new order in the take-out system is a dynamic process, which is handled with the passage of time, and the current position and remaining tasks of all the couriers should be also considered.

In this paper, from the perspective of dispatch efficiency, the evaluation function of order allocation is established based on the sum of the follow-up operation time of all the couriers in the region after receiving the new order, and the current optimal allocation scheme is determined according to the evaluation function.

Let order allocation evaluation function $T(k, 1)$ represent the total remaining delivery time of the system after the allocation of order $k$, namely,

$$T(k, 1) = \min\{F(v_u, k, 1)\}, \quad v_u \in V_{Area}$$

The equation $F(v_u, k, 1)$ represents the remaining delivery time of the entire system after assigning order $k$ to courier $v_u$, which is equal to the sum of the delivery time of courier $v_u$ after assigning order $kT(v_u, k, 1)$ and the delivery time of all other couriers not allocated order $T(v_u, k, 0)$.

$$F(v_u, k, 1) = T(v_u, k, 1) + \sum_{v_u} T(v_u, k, 0), \quad v_u \in V_{Area}$$

The delivery time of courier $v_u$ not allocated order $kT(v_u, k, 0)$ is related to status of the courier in the previous stage:

$$T(v_u, k, 0) = \begin{cases} \max\{T(v_u, k - 1, 1) - \Delta_{k,k-1}\}, & \text{if Allocate order } k - 1 \text{ to } v_u, \\ \max\{T(v_u, k - 1, 0) - \Delta_{k,k-1}\}, & \text{otherwise,} \end{cases}$$

where $t_{k-1}$ and $t_k$ are the time of order $k - 1$ and order $k - 1$ appears, and $\Delta_{k,k-1}$ represents the time span from $t_{k-1}$ to $t_k$:

$$\Delta_{k,k-1} = t_k - t_{k-1}.$$ 

Although the probability is small, it is possible to generate two or more orders at the same time. In this case, the way we deal with it is sorting these orders by random sequence, and then allocate the new orders to couriers one by one.

In order allocation and delivery route coordination optimization algorithm of the take-out system, the main algorithm is dynamic order allocation algorithm based on dispatch efficiency, and the subalgorithm is route optimization algorithm based on current optimal. For a new delivery order, we first try to assign it to a courier in the region, use the delivery route optimization subalgorithm to optimize the delivery route of this courier in real time, and calculate his remaining delivery time and the corresponding system remaining delivery time.

Repeat the process to get all the possible allocation schemes, and then, based on the evaluation function, the greedy strategy is adopted to select the optimal scheme to use. Continue the allocation process of the subsequent orders according to the sequence of their appearing time until all the orders have been allocated. The specific flow chart of the algorithm is shown in Figure 5.
4.1. Main Algorithm: Dynamic Order Allocation Algorithm

Step0: Initialization; Initialize the real-time location of all couriers and the information of all orders, let $k = 1$, and $T(v_i, k, 0) = 0$ for each courier.

Step1: For order $k$ in the system, search all the couriers in the Area to form the courier set $V_{Area}$.

Step2: Execute $(v_i = 1, V_{Area}, v_i++)$, and calculate $T(v_i, k, 1)$ and $F(v_i, k, 1)$.

Step3: Perform Sub-Algorithm of route optimization to calculate the current optimal route of courier $v_i$.

Step4: According to the evaluation function $T(k, 1)$, assign order $k$ to the corresponding courier who has the best function value.

Step5: Capacity judgment. If the capacity $C_v$ of courier $v_i$ does not exceed its maximum capacity after the order allocation in Step 4, the allocation of order $k$ is completed; if not, return to Step 2.

Step6: Judge whether the condition $K < K_{max}$ is met, if so, execute $k++,$ update all $T(v_i, k, 0)$ according to $\Delta_{k,k-1}$ and return to Step 1; otherwise, the algorithm meets the termination condition and output the result.

Figure 5: Flow chart of dynamic coordination optimization algorithm.
of order allocation and the optimal delivery route for each courier.

4.2. Subalgorithm: Current Optimal Route Optimization Algorithm. In order to find the optimal delivery route for the allocated but not yet completed orders for a certain courier, a genetic algorithm considering the special characteristics of the problem is designed in this paper. The flow of the algorithm is as follows:

Step 0 (chromosome representation): the chromosomes are coded in natural number, i.e., the corresponding codes of orders O1 and O2 are 1, 2 and 3, 4, respectively, where the odd numbers 1, 3 represent the merchant nodes while the even numbers 2, 4 represent the customer node. Since the delivery stage of the order must occur after the picking up stage, the customer node (even number) corresponding to an order in the chromosome must be behind the merchant node (odd number), and the numbers used to represent the merchant and customer are sorted by the order generation time sequence.

In the delivery system, the order delivery problem must be guaranteed to take first and send back, that is to say, for an order, in the chromosome, the corresponding point of the merchant must appear before the corresponding point of the customer. All of them can get reasonable offspring, which can satisfy the order’s first pick-up and back delivery.

Step 1 (initial population selection): the initial population is randomly generated in existing individuals according to the constraints.

Step 2 (cross operator): green denotes the intersection of genes, with the first half of the offspring being the first half of the corresponding parent and the second half in the order of the remaining elements in the other parent. All solutions must also satisfy one constraint, that is, for five orders, the business order is 1, 3, 5, 7, 9; the customer order is 2, 4, 6, 8, 10; the access order constraints of 1 before 2, 3 before 4, 5 before 6, 7 before 8, and 9 before 10 are required.

In this paper, a visit sequence-based coding is designed to solve the delivery route optimization problem. In the process of crossover, the method of generating two children from two parents is adopted. The specific operation of crossover is shown in Figure 6.

First, two parent chromosomes are randomly selected from the population according their fitness function, and a crossing position is generated randomly either, as shown in the red line in the figure. Since the chromosomes are divided into the first half and the second half by the crossing position, an effective crossover operation is designed here to ensure the feasibility of the two generated children. For child 1, the first half of father 1 was copied directly, and then the missing genes were supplemented to the second half in the order in which they appeared in father 2. Similarly, this operation can be used again to generate child 2. In this mode of crossover, as long as two parents are feasible, the offspring generated by their hybridization will also be feasible. That is, the constraint of visiting the merchant nodes (1, 3, 5, 7, 9) before their corresponding customer nodes (2, 4, 6, 8, 10) can be easily satisfied.

Step 3 (mutation operator): to maintain the diversity of population, the local changes of chromosome are generated by mutation operator. First, a chromosome is selected randomly from the population. Then, in order to keep the feasibility of chromosome after mutation, a variation mode based on node pair exchange is designed, that is, select two orders randomly and exchange the visit sequence of their merchant nodes correspondingly, as well as the customer nodes, so as to ensure that, for any chromosome after mutation, each customer node must appear after its corresponding merchant node. The specific mutation process is shown in Figure 7.

Step 4 (fitness function): the fitness function is the objective function in this paper, that is, the total delivery cost of all orders.

5. Numerical Experiments

5.1. A Simple Example. To verify the validity of the take-out order allocation and delivery route optimization model established in this paper, taking Anning District of Lanzhou City Gansu Province in China as an example, 13 orders with 26 nodes are selected, and the latitude and longitude of each node are obtained through the Golden Map API. Assuming the following take-out delivery scenario, the location of two couriers and the location of the merchants and customers are shown in Figure 8. The detailed information is shown in Table 1, where the merchants and customers correspond to the subscript. For example, the merchant node and the customer node of order O5 are m5 and c5, respectively.

In this paper, take-out order dispatch and route optimization with/without considering traffic jam are calculated and compared, in which the congestion of urban road network is reflected by the speed of couriers. If there is no traffic jam in each road section of the road network, the delivery speed of the courier is constant throughout the trip. If there is traffic jam in some road sections, the courier will pass the congested road at a lower speed.

The computing environment of the numerical experiment in this paper is Intel Core i7-7500u CPU, with 16 GB memory and windows 10 operating system, and the algorithm proposed is programed by MATLAB R2019a software. Assuming that the vehicle dispatching speed is 24.5 km/h and 35 km/h when the congestion is considered or not, the penalty cost for overtime dispatching is 1 yuan/min, and the dispatching fee is 0.1 yuan/km. The calculation termination condition is the iteration times, which is set as 100.

When considering the congestion situation in a specific period of time, it is assumed that the 30th min and...
Cross position

First half
Second half

Figure 6: A crossover operator diagram.

Figure 7: A mutation operator diagram.

Figure 8: Take-out node map location.
Table 1: Information tables for each node.

| ID    | Positional coordinates         | Waiting time (min) | Time window | Order generation time |
|-------|--------------------------------|--------------------|-------------|-----------------------|
| Courier1 | 103.723339, 36.108555              | 0                   | —           | —                     |
| Courier2 | 103.731286, 36.101268              | 0                   | —           | —                     |
| O1 | m1 103.725745, 36.100809  | 1 | [0, 15] | 11:30 |
|      | c1 103.728406, 36.106348  | 0 | [0, 35] | 11:30 |
| O2 | m2 103.725745, 36.100809  | 1 | [0, 15] | 11:35 |
|      | c2 103.728406, 36.106348  | 0 | [0, 35] | 11:35 |
| O3 | m3 103.723948, 36.101437  | 0 | [1, 19] | 11:41 |
|      | c3 103.709617, 36.090871  | 1 | [0, 40] | 11:41 |
| O4 | m4 103.724538, 36.096487  | 3 | [2, 26] | 11:50 |
|      | c4 103.699834, 36.094989  | 2 | [5, 51] | 11:50 |
| O5 | m5 103.730761, 36.099808  | 1 | [2, 31] | 11:55 |
|      | c5 103.694254, 36.098516  | 2 | [10, 50] | 11:55 |
| O6 | m6 103.730208, 36.097688  | 2 | [4, 40] | 12:00 |
|      | c6 103.697386, 36.101689  | 1 | [10, 63] | 12:00 |
| O7 | m7 103.720466, 36.104070  | 0 | [3, 48] | 12:09 |
|      | c7 103.697247, 36.106487  | 0 | [2, 31] | 12:09 |
| O8 | m8 103.723229, 36.098421  | 2 | [4, 50] | 12:15 |
|      | c8 103.717396, 36.103198  | 1 | [6, 63] | 12:27 |
| O9 | m9 103.714362, 36.107033  | 1 | [6, 63] | 12:27 |
|      | c9 103.749840, 36.087507  | 2 | [5, 72] | 12:27 |
| O10 | m10 103.745770, 36.098516 | 2 | [3, 70] | 12:38 |
|      | c10 103.741933, 36.097061  | 1 | [10, 75] | 12:38 |
| O11 | m11 103.747262, 36.098949 | 1 | [4, 83] | 12:48 |
|      | c11 103.739824, 36.104035  | 1 | [5, 90] | 12:48 |
| O12 | m12 103.748163, 36.098390  | 1 | [3, 88] | 12:54 |
|      | c12 103.743054, 36.102341  | 1 | [10, 93] | 12:54 |
| O13 | m13 103.730095, 36.097775  | 2 | [2, 95] | 13:00 |
|      | c13 103.758739, 36.087851  | 2 | [6, 104] | 13:00 |

Table 2: Matrix of congested road sections.

| m1  | m2  | m3  | m4  | m5  | m6  | m7  | m8  | m9  | m10 | m11 | m12 | m13 | p1  | p2  | p3  | p4  | p5  | p6  | p7  | p8  | p9  | p10 | p11 | p12 | p13 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1   | 1   | 0   | 1   | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 0   | 1   | 1   | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 0   | 0   | 1   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 0   | 1   | 0   | 1   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 0   | 1   | 0   | 0   | 1   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 1   | 0   | 1   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 1   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 0   | 1   | 1   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 1   | 1   | 1   | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |

10 Mathematical Problems in Engineering
56th min of the delivery system are the start and end time of traffic jam, and the congestion section matrix is shown in Table 2.

Table 3 shows the results of order allocation with/without traffic jam, and the corresponding delivery route are shown in Figures 9 and 10, respectively.

|                  | Regardless of traffic jam |          | Considering traffic jam |          |
|------------------|---------------------------|----------|-------------------------|----------|
| Allocated order  | (O3, O5, O6, O7, O8, O9, O10, O11) |          | (O1, O2, O4, O12, O13)  |          |
| Delivery route   | m3-c3-m5-c5-m6-m8-m7-c6-m9-c7-c9-c8-m10-m11-c10-c11 |          | m3-c3-m5-c5-m7-m8-m6-m9-c6-c7-c8-c9-m10-m11-c10-c11 |          |
|                  | m1-c1-m2-c2-m4-c4-m12-m12-m13-c13 |          | m1-c1-m2-c2-m4-c4-m12-m12-m13-c13 |          |
| System cost      | 6.2631                    |          | 6.4298                  |          |

Table 3: Order allocation result.

Figure 9: Order assignment and delivery route diagram regardless of road congestion. (a) Delivery route of courier 1. (b) Delivery route of courier 2.

Figure 10: Order assignment and delivery route diagram considering road congestion. (a) Delivery route of courier 1. (b) Delivery route of courier 2.
| Courier | Positional coordinates |
|---------|------------------------|
| 1       | 103.723339, 36.108555 |
| 2       | 103.731286, 36.101268 |
| 3       | 103.728457, 36.098558 |
| 4       | 103.746996, 36.092802 |
| 5       | 103.719530, 36.109986 |
| 6       | 103.707771, 36.095021 |
| 7       | 103.725882, 36.108267 |
| 8       | 103.725882, 36.108267 |
| 9       | 103.717899, 36.099737 |
| 10      | 103.753691, 36.092594 |
| 11      | 103.701592, 36.104453 |
| 12      | 103.726826, 36.099876 |

| Table 5: Information tables for each node. |

| ID     | Positional coordinates | Waiting time (min) | Time window | Order generation time |
|--------|------------------------|--------------------|-------------|-----------------------|
| O1     | m1 103.725745, 36.100809 | 1                  | [0, 15]     | 11:30                 |
|        | c1 103.728406, 36.106348 | 0                  | [0, 35]     | 11:30                 |
| O2     | m2 103.725745, 36.100809 | 1                  | [0, 15]     | 11:32                 |
|        | c2 103.728406, 36.106348 | 0                  | [0, 35]     | 11:32                 |
| O3     | m3 103.723948, 36.101437 | 0                  | [1, 19]     | 11:35                 |
|        | c3 103.709617, 36.090871 | 1                  | [0, 40]     | 11:35                 |
| O4     | m4 103.724538, 36.096487 | 3                  | [2, 18]     | 11:36                 |
|        | c4 103.699833, 36.090897 | 2                  | [5, 45]     | 11:36                 |
| O5     | m5 103.730671, 36.099808 | 1                  | [2, 19]     | 11:37                 |
|        | c5 103.694254, 36.094989 | 2                  | [10, 50]    | 11:37                 |
| O6     | m6 103.730208, 36.097688 | 2                  | [4, 17]     | 11:38                 |
|        | c6 103.697386, 36.106189 | 1                  | [10, 55]    | 11:38                 |
| O7     | m7 103.720466, 36.101407 | 0                  | [3, 17]     | 11:40                 |
|        | c7 103.697247, 36.106423 | 0                  | [0, 50]     | 11:40                 |
| O8     | m8 103.723229, 36.098421 | 2                  | [4, 19]     | 11:42                 |
|        | c8 103.717396, 36.103198 | 1                  | [0, 35]     | 11:42                 |
| O9     | m9 103.714362, 36.107033 | 1                  | [6, 18]     | 11:44                 |
|        | c9 103.749845, 36.087507 | 2                  | [5, 40]     | 11:44                 |
| O10    | m10 103.745770, 36.098516 | 2                  | [3, 21]     | 11:45                 |
|        | c10 103.741933, 36.097061 | 1                  | [10, 50]    | 11:45                 |
| O11    | m11 103.747262, 36.098949 | 1                  | [4, 23]     | 11:46                 |
|        | c11 103.739824, 36.104035 | 1                  | [5, 40]     | 11:46                 |
| O12    | m12 103.748163, 36.098390 | 1                  | [3, 30]     | 11:47                 |
|        | c12 103.743054, 36.102341 | 1                  | [10, 50]    | 11:47                 |
| O13    | m13 103.730095, 36.094775 | 2                  | [2, 40]     | 11:48                 |
|        | c13 103.758739, 36.087851 | 2                  | [6, 65]     | 11:48                 |
| O14    | m14 103.727427, 36.099113 | 2                  | [0, 45]     | 11:50                 |
|        | c14 103.725023, 36.110278 | 0                  | [0, 65]     | 11:50                 |
| O15    | m15 103.730259, 36.107157 | 3                  | [0, 45]     | 11:52                 |
|        | c15 103.744507, 36.096062 | 1                  | [0, 65]     | 11:52                 |
| O16    | m16 103.748112, 36.090513 | 2                  | [1, 49]     | 11:54                 |
|        | c16 103.739273, 36.095229 | 3                  | [0, 80]     | 11:54                 |
| O17    | m17 103.727083, 36.099945 | 3                  | [2, 47]     | 11:56                 |
|        | c17 103.696957, 36.105008 | 2                  | [5, 85]     | 11:56                 |
| O18    | m18 103.732319, 36.105146 | 2                  | [2, 47]     | 11:58                 |
|        | c18 103.715839, 36.105979 | 0                  | [10, 80]    | 11:58                 |
| O19    | m19 103.715754, 36.102927 | 0                  | [4, 49]     | 12:00                 |
|        | c19 103.715153, 36.111041 | 1                  | [10, 85]    | 12:00                 |
| O20    | m20 103.720732, 36.100153 | 0                  | [3, 48]     | 12:02                 |
|        | c20 103.703909, 36.096894 | 0                  | [0, 80]     | 12:02                 |
| O21    | m21 103.733177, 36.105840 | 2                  | [4, 51]     | 12:04                 |
|        | c21 103.731203, 36.094813 | 4                  | [0, 80]     | 12:04                 |
From the calculation results, it can be found that the order allocation of the two couriers are the same when the traffic jam is considered or not. Courier 1 is assigned 8 orders (O3, O5, O6, O7, O8, O9, O10, O11) while courier 2 is assigned 5 orders (O1, O2, O4, O12, O13). In the first scenario, the congestion is not considered and the delivery route of courier 1 is m3-c3-m5-c5-m6-m8-m7-c6-c9-c8-m10-m11-c10-c11 and that of courier 2 is m1-c1-m2-c2-m4-c4-m12-c12-m13-c13. In the second scenario, when there is a traffic jam, the delivery route of courier 1 is changed to m3-c3-m5-c5-m7-m8-m6-m9-c7-c9-m10-m11-c10-c11, whereas the delivery route of courier 2 is still m1-c1-m2-c2-m4-c4-m12-c12-m13-c13. For the system cost, the one considered traffic jam is slightly higher than the other one.

5.2. Large-Sized Numerical Experiments. This section aims to conduct more numerical experiments to examine the capability of the proposed model and algorithm. In this paper, a large-scale calculation example composed of 12 couriers and 40 orders is designed. The information of couriers and orders is shown in Tables 4 and 5.

In this paper, the calculation parameters and the running environment of the large case are the same as those of the small one, and the proportion of congested road is about 50%. The results of take-out order allocation and path optimization considering traffic jam or not are shown in Tables 6 and 7, respectively.

It can be found that, in both condition with or without traffic jam, there are some differences between the result of order allocation and route optimization. The couriers 1, 6, and 11 are not assigned any delivery orders when the traffic status is good, while the couriers 1, 7, and 11 are not assigned any tasks when traffic congestion is happened. The reason for this result is that the objective function of the problem here we studied is to minimize the total system dispatch cost, but not the delivery cost or income balance of single courier.

For this numerical test, the system dispatch cost considering traffic jam or not is 41.1540 yuan and 55.0888...
It can be found that the computing time is almost the same for the same scale cases. For the average computing time of each order, the one of large-scale examples is significantly longer than that of simple examples. With the increase of the number of couriers and tasks, the scale of the road network is correspondingly expanded, resulting in the sharp increase of the feasible delivery routes. Therefore, the search efficiency of the algorithm in each iteration is also reduced. Although the computing time of large-scale examples has increased, it can still basically meet the application needs in practical according to this scale. In the design of this paper, the connection between nodes is simplified, and all nodes are all connected, which leads to the delivery route obtained.

| Table 6: Results of order allocation and delivery route without considering congestion. |
|---|---|---|
| Courier | Allocated order | Delivery route |
| Courier 1 | — | — |
| Courier 2 | (O14, O21, O24, O32, O33, O37, O38, O39, O40) | m14-c14-m21-c21-m24-c24-m32-c32-m33-c33-m38-c38-m39-c39-m40-c40 |
| Courier 3 | (O2, O15, O29) | m2-c2-m15-c15-m29-c29 |
| Courier 4 | (O28) | m28-c28 |
| Courier 5 | (O3, O4, O6, O7, O8) | m3-m8-m4-m6-m7-c3-c4-c6-c7 |
| Courier 6 | — | — |
| Courier 7 | (O9, O10, O11, O12, O13, O16, O17, O20, O25, O26, O27) | m9-m13-m10-m11-m12-c13-c9-m16-c10-c12-c11-m25-m27-m17-m26-m20-c16-c25-c20-c17-c26 |
| Courier 8 | (O5, O18, O19, O22) | m5-c5-m19-m18-m22-c18-c19-c22 |
| Courier 9 | (O23, O34, O35, O36) | m23-c23-m35-m36-c35-c34-c36 |
| Courier 10 | (O30, O31) | m30-m31-c31-c30 |
| Courier 11 | — | — |
| Courier 12 | (O1) | m1-c1 |

| Table 7: Results of order allocation and delivery route considering congestion. |
|---|---|---|
| Courier | Allocated order | Delivery route |
| Courier 1 | — | — |
| Courier 2 | (O11, O12, O33) | m12-m11-c12-c11-m33-c33 |
| Courier 3 | (O2, O15, O32, O36, O39) | m2-c2-m15-c15-m32-m36-c36-c32-c39 |
| Courier 4 | (O19, O22, O23, O25, O26, O28) | m19-c19-m22-m23-m25-c23-c22-c25-m26-c26-m28-c28 |
| Courier 5 | (O3, O4, O6, O7, O8, O27) | m3-m8-m4-m6-m7-c3-c4-c8-m27-c27 |
| Courier 6 | (O10, O13, O21, O24, O29, O30, O31) | m21-m24-m13-c21-m10-c24-c10-m31-m29-m30-c31-c29-c30-c13 |
| Courier 7 | — | — |
| Courier 8 | (O5, O9, O17, O20) | m5-m9-c5-c9-m17-m20-c20-c17 |
| Courier 9 | (O23, O35, O40) | m23-c23-m35-m40-c35-c40 |
| Courier 10 | (O16, O38) | m16-c16-m38-c38 |
| Courier 11 | — | — |
| Courier 12 | (O1, O14, O18, O34, O37) | m1-c1-m14-c18-c14-m34-m37-c34-c37 |

| Table 8: Comparison of numerical test calculation time. |
|---|---|---|
| Type of road network congestion | Total computing time (s) | Average computing time per order (s) |
| Simple example | Regardless of road network congestion | 2 | 0.15 |
| | Consider road network congestion | 2 | 0.15 |
| Large-sized numerical experiments | Regardless of road network congestion | 112 | 2.8 |
| | Consider road network congestion | 114 | 2.85 |
by the algorithm being far larger than the actual problem. In practical application, when the number of paths in physical network is reduced, whereas the performance of large-scale commercial computing server is higher, the efficiency of the algorithm will be greatly improved.

6. Conclusions

In this paper, order dispatching and route coordination optimization of the take-out system is studied. A route optimization model based on the minimum cost of system dispatch is established to deal with the complexity of space-time network of the take-out dispatch system. Considering the real-time dynamic characteristic of order distribution and route optimization, the coupling relationship between order allocation and route optimization is established based on the delivery process. The main algorithm designed in this paper includes two core parts, one of which is the order allocation algorithm based on the highest efficiency of system dispatch and the other is the route optimization algorithm based on the improved genetic algorithm. A numerical experiment is constructed to verify the effectiveness of the model and the algorithm, and the result of two different traffic status are calculated. The computing results show that the algorithm not only has a better operation efficiency but also can reduce the dispatching cost of the delivery system, therefore has great practical significance.

Our current research studies from the perspective of system optimization, which is very necessary in some emergency scenarios, such as the epidemic of COVID-19. In the further research, we will pay more attention to the equilibrium of the order allocation, and continue to study the interactive solution strategies of order allocation and route optimization to realize the real-time optimization of large-scale dispatch problem in the urban system.

Data Availability

All data included in this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This study was jointly supported by grants from the National Natural Science Foundation of China (Grant no. 71761023) and Natural Science Foundation of Gansu Province, China (Grant no. 18JR3RA110).

References

[1] M. Li, J. Zhang, and W. Wang, “Task selection and scheduling for food delivery: a game-theoretic approach,” in Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), IEEE, Abu Dhabi, UAE, February 2018.

[2] B. Harman, “Systems and methods for take-out order analytics,” U.S. Patent No. 9105041, 2015.

[3] B. Harman, “Systems and methods for take-out order management,” U.S. Patent No. 9805351, 2017.

[4] Y. Liu, B. Guo, H. Dai, Z. Yu, D. Zhang, and C. Chen, “Poster: FooDNet: optimized on demand take-out food delivery using spatial crowdsourcing,” in Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking, ACM, Salt Lake City, UT, USA, October 2017.

[5] Y. Liu, B. Guo, C. Chen et al., “FooDNet: toward an optimized food delivery network based on spatial crowdsourcing,” IEEE Transactions on Mobile Computing, vol. 18, no. 6, pp. 1288–1301, 2019.

[6] W. Liao, L. Zhang, and Z. Wei, “Multi-objective green meal delivery routing problem based on a two-stage solution strategy,” Journal of Cleaner Production, vol. 258, Article ID 120627, 2020.

[7] F. Jiang, M. Xu, and D. Cui, “Scheduling system based on takeaway logistics big data,” Big Data Research, vol. 3, no. 1, pp. 109–115, 2017.

[8] H. Dai, J. Tao, H. Jiang, and W. Chen, “O2O on-demand delivery optimization with mixed driver forces,” IFAC-PapersOnLine, vol. 52, no. 13, pp. 391–396, 2019.

[9] Y. Shi, M. Chen, T. Qu, and W. Liu, “Vehicle routing problem of an innovative B2C and O2O joint distribution service,” Procedia CIRP, vol. 83, pp. 680–683, 2019.

[10] J. Li, Y. Zheng, B. Dai, and J. Yu, “Implications of matching and pricing strategies for multiple-delivery-points service in a freight O2O platform,” Transportation Research Part E: Logistics and Transportation Review, vol. 136, Article ID 101871, 2020.

[11] Z. He, G. Han, T. C. E. Cheng, B. Fan, and J. Dong, “Evolutionary food quality and location strategies for restaurants in competitive online-to-offline food ordering and delivery markets: an agent-based approach,” International Journal of Production Economics, vol. 215, pp. 61–72, 2019.

[12] M. Reihaneh and G. Ahmed, “A branch-and-price algorithm for a vehicle routing with demand allocation problem,” European Journal of Operational Research, vol. 272, no. 2, pp. 523–538, 2019.

[13] K. Braeckers, K. Ramaekers, and I. Van Nieuwenhuyse, “The vehicle routing problem: state of the art classification and review,” Computers & Industrial Engineering, vol. 99, pp. 300–313, 2016.

[14] G. B. Dantzig and J. H. Ramser, “The truck dispatching problem,” Management Science, vol. 6, no. 1, pp. 80–91, 1959.

[15] H.-K. Chen, H.-W. Chou, and C.-Y. Hsu, “The linehaul-feeder vehicle routing problem with virtual depots and time windows,” Mathematical Problems in Engineering, vol. 2011, Article ID 759418, 15 pages, 2011.

[16] R.-M. Chen and Y.-M. Shen, “Novel encoding and routing balance insertion based particle swarm optimization with application to optimal CVRP depot location determination,” Mathematical Problems in Engineering, vol. 2015, Article ID 743507, 11 pages, 2015.

[17] X.-H. Liu, M.-Y. Shan, R.-L. Zhang, and L.-H. Zhang, “Green vehicle routing optimization based on carbon emission and multiobjective hybrid quantum immune algorithm,” Mathematical Problems in Engineering, vol. 2018, Article ID 8961505, 9 pages, 2018.

[18] B. Li, X. Yang, and H. Xuan, “A hybrid simulated annealing heuristic for multistage heterogeneous fleet scheduling with fleet sizing decisions,” Journal of Advanced Transportation, vol. 2019, Article ID 5364201, 19 pages, 2019.
[19] J. Ma and G. Sun, “Mutation ant colony algorithm of milk-run vehicle routing problem with fastest completion time based on dynamic optimization,” *Discrete Dynamics in Nature and Society*, vol. 2013, Article ID 418436, 6 pages, 2013.

[20] G. Martinovic, I. Aleksic, and A. Baumgartner, “Single-commodity vehicle routing problem with pickup and delivery service,” *Mathematical Problems in Engineering*, vol. 2008, Article ID 697981, 17 pages, 2008.

[21] H. Niu, X. Zhou, and X. Tian, “Coordinating assignment and routing decisions in transit vehicle schedules: a variable-splitting Lagrangian decomposition approach for solution symmetry breaking,” *Transportation Research Part B: Methodological*, vol. 107, pp. 70–101, 2018.