Paper:

A Takeaway Based on Crowdsensing Path Planning Method: Using Idle Bicycles Resources for Delivery

Hongrui Zeng[1,2,3], Yonghua Xiong[1,2,3] and Jinhua She[4]

[1]School of Automation, China University of Geosciences, Wuhan 430074, China
[2]Hubei key laboratory of Advanced Control and Intelligent Automation for Complex Systems, Wuhan 430074, China
[3]Engineering Research Center of Intelligent Technology for Geo-Exploration, Ministry of Education, Wuhan 430074, China
[4]School of Engineering, Tokyo University of Technology, Tokyo 192-0982, Japan
E-mail:cugzhr947@163.com; xiongyh@cug.edu.cn; she@stf.teu.ac.jp

[Received 2024/06/29; accepted 2024/09/12]

Abstract: Due to the surge in takeaway demand, full-time delivery riders are facing challenges in completing their tasks on time. To address this issue. This study proposes an innovative takeaway delivery method based on Crowdsensing, which leverages the availability of idle bicycle riders to execute delivery tasks at a lower cost. To optimize the total delivery cost for merchants, we formulated an idle bicycle path planning model. This model, characterized by multiple riders, random locations, open deliveries, and intersecting pickup and delivery routes. To effectively address this path planning model, a takeaway delivery-based genetic algorithm was designed. Order-preserving initialization was utilized to construct the initial population. Adaptive mutation algorithm were introduced to overcome slow convergence and prevent the probability of falling into the local optimal solution. Experimental results indicate that the proposed genetic algorithm exhibits rapid convergence and high accuracy, successfully resolving the path planning model. Consequently, this method significantly reduces delivery costs and enhances delivery timeliness, demonstrating substantial application value.

Keywords: Crowdsensing, path planning, optimal algorithm, takeaway delivery

1. Introduction

With the increased demand in the takeaway market, the large number of orders has caused challenges to merchants, riders, and other groups [1]. The high number of orders not only increases the workload of riders but also causes delivery delays in some orders. The latter leads to a series of disputes. For example, merchants receive negative reviews from customers, orders are cancelled, and the rider's rewards are deducted. In this case, merchants often increases the number of riders under the traditional delivery system, which currently refers to platform delivery and merchant self-delivery. It is not suitable to use

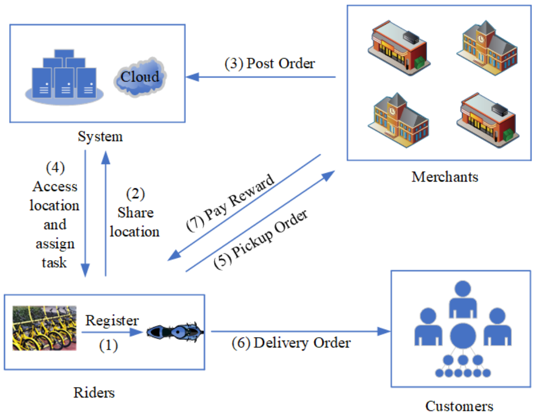

Fig. 1. Process of the takeaway delivery mode based on Crowdsensing

the traditional delivery mode because different merchants have various transaction amounts. In addition, a greater number of riders increases the delivery cost. With the widespread application of the Crowdsensing technology [2], which can better improve delivery delays, this study combined the idea of Crowdsensing in the field of takeaway delivery and proposed the takeaway delivery mode based on Crowdsensing, a novel delivery system. In this model, if there are a large number of orders, merchants can hire idle bicycle riders from the community, rather than full-time riders, to participate in delivery tasks at a lower cost. Idle bicycle resources refer to the bicycles and electric bicycles that are in idle period from shared platforms, individuals, enterprises, and other groups. In this novel mode, idle bicycle riders must be registered and pass relevant assessments in advance before they can be hired as system riders (referred to as "idle riders" hereafter). If idle riders are available, they should log into the system and share their current location. The system can assign delivery tasks to them. Figure 1 shows the process in this mode. First, the rider should register as an idle rider, then log into the system, and share their location with the system when they are idle. Merchants can post the orders to the system when the number of orders is large. Then, the system checks the number of available idle riders and assigns orders based on their locations. The rider should pick up the package from the merchant according to the task and deliver the order to the corresponding customer location. The task is finished when the idle rider reaches the location of the last customer. Finally, the merchant pays a reward to the idle rider. The path planning under the Crowdsensing takeaway delivery mode differs from that under the traditional takeaway delivery mode based on the following points: (1)No distribution center: The rider starts at a random location, and he/she can deliver nearby from the current location. (2)Open delivery: The rider does not need to return to the starting point when completing the delivery task, thereby reducing transportation cost. (3)Multi-rider delivery: The system has more delivery plannings due to multiple riders participating in delivery.

2. Related works

This section summarizes the current status of research on Crowdsensing application and vehicle routing problem (VRP).

2.1. Application of Crowdsensing

Crowdsensing is now widely used in fields such as energy dispatching [3], disease prevention [4], urban monitoring [5], and disaster prediction [6]. Literature [7] addresses noise detection issues by collecting noise data through community-driven sensing infrastructure, providing a new paradigm for gathering urban contextual sensing data. Literature [8] introduces a mobile crowdfunding platform synthesized on the edge, tracking infected individuals' trajectories to assess potential risk groups and warn users through smart notifications. Recent research in the field of food delivery includes literature [9], which incorporates drones and crowdsourcing into existing delivery systems and introduces an efficient adaptive hyper-heuristic method to optimize the routing for mixed closed-open delivery vehicles. Literature [10] proposes an effective large-scale mobile crowd tasking model that improves delivery efficiency by constructing the problem as a network minimum cost flow issue and applying pruning techniques to reduce network size. Literature [11] establishes a food delivery network utilizing spatial crowdsourcing, recruiting city taxis for meal delivery. Literature [12] explores on-demand meal delivery with drones, presenting an iterative heuristic framework for dynamic task scheduling with random task arrivals and deadlines, optimized by a simulated annealing-based local search method.

The above approaches have yet to show ideal timeliness and cost efficiency in the face of delivery delays and cost issues in meal delivery. Therefore, this paper proposes a takeaway delivery mode based on Crowdsensing.

2.2. Vehicle Route Issue

The takeaway path planning issue belongs to the vehicle route problem, which refers to designed optimal routes for vehicles to serve customers under certain constraints, and the NP-hard problem [13]. Common route planning problems include the Vehicle Routing Problem with Time Windows (VRPTW), the Open Vehicle Routing Problem with Time Windows (OVRPTW), the Vehicle Routing Problem with Pickup and Delivery (VRPPD), and the Multi-Vehicle Routing Problem with Time Windows (MVRPTW) [14].

The typical VRPTW considers time constraints during route planning. In literature [15], set against an eco-friendly backdrop, the electric vehicle charging route planning with time windows is examined, a mixed-integer programming model is constructed, and an ant colony probability selection algorithm is proposed to find a satisfactory charging route. The typical VRPPD requires considering the order of picking up and then delivering goods during route planning. Literature [16], against the backdrop of consumer online shopping, addresses the vehicle routing problem with pickup and delivery, proposing a waiting strategy and validating its significance and application through experiments, providing a reference for logistics companies to establish operational strategies. The typical OVRPTW adopts an open delivery system based on the time window routing problem, where riders do not need to return to the starting point after delivery. Literature [17], in the context of third-party logistics and fuel consumption issues, proposes a mathematical model for the green open vehicle routing problem with time windows and devises a mixed tabu search algorithm incorporating various neighborhood search strategies, significantly reducing the total and emission costs. The typical MVRPTW considers time window constraints based on multi-rider route planning. Literature [18] introduces a two-phase hybrid algorithm with time windows and multiple vehicles for the vehicle routing problem of pick-up and drop-off, aimed at reducing the total travel cost.

It is evident that most VRP and its variant models are often resolved using intelligent optimization algorithms, but the algorithms mentioned above are not directly applicable to the meal delivery model based on Crowdsensing. To better solve this issue, the current study designed the TDGA based on takeaway delivery.

3. Description of the problem and modeling

In this section, we firstly posed and described the related problem using mathematical expressions, and then established the multi-rider, open-delivery VRP with time windows pickup and delivery (MROVRPTWPD) model. The details are shown in the following chapters.

3.1. Problem Description

This issue can be described by the network $G = (N, A)$. N represents a set of nodes,$N = N_O \cup N_r$, wherein N_O represents the set of rider nodes, and $N_r = N_r^+ \cup N_r^-$, wherein N_r^+ represents the set of merchant nodes and N_r^- represents the set of customer nodes. A represents the set of all edges, $A = A_r \cup E_r$, wherein $A_r = \{(i,j)\,|i,j \in N_r, i \neq j\}$ represents four situations (merchant node to the customer node edge set, customer node to the merchant node edge set, customer node to the customer node edge set, and merchant node to the merchant node edge set). $E_r = \{(o,j)\,|o \in N_O, j \in N_r\}$ represents the edge set of the rider from the starting position to the first merchant node.

Assuming that the time window for the rider to pick up package at the merchant node is $[m_{i^+}, n_{i^+}]$, the latest pick up time is $a_{i^+}(a_{i^+} > \mathrm{n}_{i^+})$, where i^+ represents the time of arrival at the merchant node. The time window for the rider to reach the customer node is $[m_{i^-}, n_{i^-}]$. The latest delivery time is $a_{i^-}(a_{i^-} > \mathrm{n}_{i^-})$. i^- indicates that the rider has reached the customer node moment. Assuming an order is $r = (i,j)$, $i \in N_r^+, j \in N_r^-$, the rider needs to first reach node i and then reaches the corresponding node j of node i. R is the set of r for each order, and there are K riders in this area.

3.2. Decision Variables

There are two decision variables in this model:

(1)Path decision variables:

$$X_{ij}^k = \begin{cases} 1, k\text{th rider goes from node } i \text{ to node } j \\ 0, \text{otherwise} \end{cases} \tag{1}$$

when the rider K travels from node i to node j, rider K travels along route (i,j), X_{ij}^k is equal to 1; otherwise is 0.

(2)Task decision variables:

$$Y_r^k = \begin{cases} 1, \text{assign order } r \text{ to rider } k \\ 0, \text{otherwise} \end{cases} \tag{2}$$

when rider K takes order r, Y_r^k is equal to 1; otherwise is 0.

3.3. Constraints

The model has the following constraints:

(1)Each order is completed by one rider only.

$$\sum_{k \in K} Y_r^k = 1, \forall r \in R \tag{3}$$

(2)Each customer node is only accessed once.

$$\sum_{k \in K} \sum_{j \in N_r} X_{ij}^k = 1, \forall i \in N_r \tag{4}$$

(3)The rider departs from the starting position at 0 moments.

$$T_o = 0, \forall o \in N_O \tag{5}$$

(4)Each rider starts from their starting position.

$$\sum_{o \in N_O} \sum_{i \in N_r} X_{oi}^k = 1, \forall k \in K \tag{6}$$

(5)The rider's service ends with the last customer.

$$\sum_{k \in K} \sum_{j \in N_r} X_{ij}^k = 0, \forall i \in N_r \tag{7}$$

(6)The riders have a limit on accepting orders.

$$\sum_{r \in R} Y_r^k \leq Q \tag{8}$$

(7)The vehicle has a maximum driving distance.

$$\sum_{i \in N_r} \sum_{j \in N_r} X_{ij}^k \cdot \mathrm{d}_{ij} \leq L, \forall k \in K \tag{9}$$

(8)The vehicle load must not exceed the maximum capacity.

$$\lambda_i \sum_{r \in R} Y_r^k \leq C \tag{10}$$

(9)The number of riders assigned to a task cannot exceed the maximum number of riders.

$$\sum_{k \in K} \sum_{o \in N_O} \sum_{i \in N_r} X_{oi}^k \leq K \tag{11}$$

(10)Riders cannot stay on the customer.

$$\sum_{i \in N_r} X_{ih}^k - \sum_{j \in N_r} X_{hj}^k = 0, \forall k \in K, \forall h \in N_r \tag{12}$$

3.4. Objective function

This section considers the constraints affecting the total delivery cost of merchants, and divides the total cost into fixed cost, empty cost, transportation cost, and time penalty cost. The specific objective functions of each part are as follows:

(1)Fixed cost refers to rider salaries, vehicle maintenance costs, etc.

$$F_1 = C_1 \sum_{k \in K} \sum_{o \in N_O} \sum_{i \in N_r} X_{oi}^k \tag{13}$$

(2)Idle cost refers to the time, manpower, and electricity costs incurred by a rider from the current location to the first merchant location node.

$$F_2 = C_2 \sum_{k \in K} \sum_{o \in N_O} \sum_{i \in N_r} X_{oi}^k d_{oi} \tag{14}$$

d_{oi} represents the distance from node o to node i.

(3)Transportation cost refers to the time, manpower, and electricity costs incurred by a rider from a merchant or customer node to another customer or merchant node.

$$F_3 = C_3 \sum_{k \in K} \sum_{i \in N_r} \sum_{j \in N_r} X_{ij}^k d_{ij} \tag{15}$$

d_{ij} represents the distance from node i to node j.

(4)Time penalty cost refers to the time penalty cost. If the rider fails to arrive at the delivery location within the specified time period (i.e., early or late), the merchant needs to pay a fine. If the rider arrives at the delivery location on time, the time penalty cost is 0.

$$F_4 = C_4 \left(\sum_{i \in N_r} \max\{m_i - T_i, 0\} + \sum_{i \in N_r} \max\{T_i - n_i, 0\} \right) \tag{16}$$

Assuming the specified delivery time period is (m_i, n_i), T_i is the moment when the rider reaches the customer node.

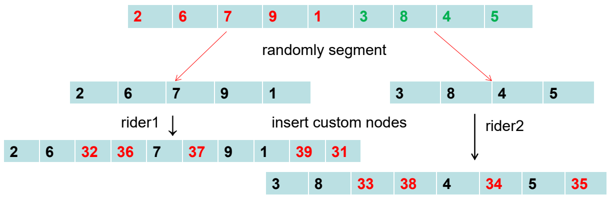

Fig. 2. Chromosome coding

(5)The opitimization goal of this model is to minimize the total cost.

$$\min F = F_1 + F_2 + F_3 + F_4 \quad \text{(17)}$$

4. Design of the TDGA

Traditional GAs have issues such as low convergence speed and poor local search ability when solving MROVRPTWPD issues. Thus, this study aimed to increase the accuracy and convergence speed of the GA by improving the algorithm construction, and this improved algorithm was called the TDGA.

4.1. Process of the TDGA

Based on the traditional GA, improvements have been made on population initialization, crossover, and mutation. The solution process of the designed algorithm is as shown in table 1:

4.2. Rules of chromosome coding

The delivery path adopts integer encoding. The chromosome is encoded as an integer string, and its length is based on the number of customers and available riders. Each string comprises several substrings, each of which is a genetic sequence representing the merchant and customer nodes that a rider passes through during the delivery process. During the encoding process, two constraints must be met, all nodes have a strict delivery order, and the order accepted by the rider has an upper limit. First, the merchant nodes are encoded, and the chromosomes are segmented to select as few riders as possible to complete the delivery task while meeting the maximum order quantity constraint that the riders can accept. Next, the corresponding customer nodes are inserted. If the node delivery order is met, the corresponding customer nodes of the merchant should be inserted into the chromosome segments to generate a feasible delivery plan. Figure 2 shows the coding steps for assigning 9 orders to 2 riders. Rider 1 completes the delivery tasks of 2, 6, 7, 9, and 1 and inserts corresponding customer nodes 32, 36, 37, 39, and 31 while meeting the abovementioned constraints. Rider 2 completes the delivery tasks of 3, 8, 4, and 5 and inserts corresponding customer nodes 33, 38, 34, and 35 while meeting the abovementioned constraints.

4.3. Method of order-preserving initialization

Set the population size and chromosome length, separate the merchant nodes from the customer nodes, and then randomly arrange the merchant nodes as the starting point for the rider. Randomly assign N merchant nodes to rider K, and finally insert the customer nodes into the merchant node in order based on the corresponding association between customers and merchants. If the current merchant's location is at the end of the merchant node sequence, it means that the merchant has no other customers. Simply insert its corresponding customer node at the end of the sequence. Otherwise, the corresponding customer node must be inserted in the adjacent merchant sequence. Finally, use as few riders as possible to serve all merchants and customer nodes and allocate them to the remaining riders in the same way. After all chromosomes are initialized, the final result is a matrix of $NIND$ rows and N columns, where each row is a chromosome. Each column represents a feasible route plan.

4.4. Method for calculating fitness

According to the optimization goal of this study, the calculation formula of population fitness function is designed as follows.

$$f = \frac{1}{F_1 + F_2 + F_3 + F_4} \quad \text{(18)}$$

The larger the fitness value of the chromosome, the higher the probability of being selected in the next iteration.

4.5. Method of the adaptive mutation algorithm

Since traditional mutation methods are prone to premature generation of local optimal solutions, an adaptive mutation algorithm is introduced. This algorithm adjusts the mutation probability in the algorithm process according to the fitness of the population, which can not only reduce the mutation rate to preserve good solutions but also prevent premature convergence. This method compares the current average fitness with the previous generation fitness and adjusts the mutation probability according to the results. If the current fitness is better than the previous generation's fitness, the mutation probability will be reduced. If the current fitness is less than the previous generation's fitness, the mutation probability will increase. If the current average fitness is consistent with the previous generation's fitness, the mutation probability will remain unchanged. The formula is as follows:

$$P_m = \begin{cases} P_m = P_m \times 1.01, Currfit < Lastfit \\ P_m = P_m \times 0.9, Currfit > Lastfit \\ P_m = P_m, Currfit = Lastfit \end{cases} \quad \text{(19)}$$

Next, randomly select two points for mutation, randomly select two merchant nodes and their corresponding customer nodes from different delivery plans, and perform gene exchange while keeping other genes unchanged. Add the mutated results to the next generation. Figure 3 depicts the specific steps.

Table 1. TDGA algorithm

TDGA algorithm
Ensure: Let vehicle capacity of each electric bicycle is C, delivery speed of per rider is V, number of available of riders is K. Set the number of iterations is 500, population size is 500, selection probability is 0.9, crossover probability is 0.95, mutation probability is 0.5. 1: Order-preserving initialization: separate the merchant nodes from the customer nodes, and randomly arrange the separated merchant nodes first; assign A merchant nodes to rider K, and finally insert the customer nodes according to the corresponding relationship between customers and merchants. The final result is a matrix of $NIND$ rows and N columns. 2: Calculate the fitness f of each individual and let the best individual is $bestcost$. 3: Use Stochastic Universal Sampling(Sus) method to select individuals in the current population to be the parent population, S_k, for the next generation. 4: Apply the OX crossover algorithm to randomly select a segment from each parent individual and create two offspring individuals, this will yield a hybrid population, C_k. 5: Use the adaptive mutation algorithm to adjust the mutation rate according to the fitness of individuals in the population, and then use the adjusted probability to mutate the selected individuals, which will yield a new population, M_k. And add them to the next generation. 6: Check if the crossover mutation operation improves the solution. If so, update the result, otherwise the algorithm will keep the current optimal result. 7: If the termination condition is met, output the best solution. Otherwise, let $K = K + 1$, and repeat step 1.

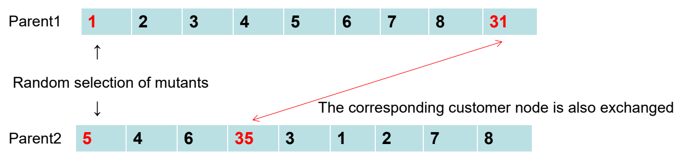

Fig. 3. Adaptive Mutation Algorithm process

5. Experiment and analysis

There are mainly two parts in this section. The first part explains the experimental data and parameter setting. The second part provides a detailed description of the experimental process.

5.1. Experimental data and explanation

The experimental data were selected from a specific shopping area within a 3-km radius at Furong District, Changsha City, Hunan Province, China. The data related to orders from the takeaway platform within a certain period of time were used as the test set. To validate the efficacy of the method proposed in this paper, two different scale test data were generated, including dataset A (10 riders, 30 orders) and dataset B (20 riders, 60 orders).

Table 2 depicts the information related to dataset A. Serial numbers 1-30 represent merchant nodes, and serial numbers 31-60 represent the customer nodes, where node 1 merchant corresponds to node 31 customer, and so on. M, C represent the merchant node and the customer node, respectively. X and Y represent the horizontal and vertical coordinates of the customer and merchant nodes, respectively. M_i and N_i represent the start and end windows of the customer's service time (minutes), respectively. A_i represents the maximum tolerance time (minutes). The distance between nodes (meters) is represented by the Euclidean distance.

The algorithm parameters in the experiment are set as follows: population size,$NIND = 500$; number of iterations, $MAXGEN = 500$; selection probability, $P_s = 0.9$; crossover probability, $P_c = 0.95$; mutation probability, $P_m = 0.5$; load capacity, $q = 10$; speed, $V = 20$.

5.2. Experimental results and analysis

In this section, we conducted algorithm comparison experiments in dataset A and dataset B to compare the TDGA algorithm with the GA algorithm to verify the effectiveness of the algorithm proposed in this paper. Table 3 shows the experimental results of the delivery mode based on Crowdsensing using TDGA.

As shown in Table 3, the order was delivered by eight riders, with a total cost ranging from 494 CNY to 531 CNY. In the eighth and ninth experiments, the time penalty cost was 0, which confirmed that all riders delivered on time under this delivery plan. However, the fixed cost under this plan was relatively high. Thus, it is not the optimal delivery plan. In the first experiment, the fixed cost was low. The number of riders selected was relatively small. The time penalty cost during delivery was relatively low. Therefore, based on this experiment, the total cost was lower. This study showed that the proposed delivery model based on Crowdsensing was feasible.

Table 4 shows the optimal comparasion result of the GA algorithm and TDGA algorithm. Based on the table 4, the minimum cost of the TDGA algorithm was significantly smaller than that of the GA algorithm. Further, it was not affected by the data size and number of experiments. In dataset A, the total cost of the TDGA algorithm decreased by 34.16. Meanwhile, in dataset B, the total cost of the TDGA algorithm decreased by 56.97.

Table 2. The information of dataset A

M	X	Y	M_i	N_i	A_i	C	X	Y	M_i	N_i	A_i
1	2725	2200	0	10	15	31	1925	2062.5	0	10	20
2	575	587.5	0	10	15	32	1987.5	412.5	0	10	20
3	1062	875	0	10	15	33	737.5	425	0	10	20
4	1650	1512.5	0	10	15	34	1837.5	2362.5	0	10	20
5	637.5	1012.5	0	10	15	35	1262.5	2075	0	10	20
6	1425	1887.5	0	10	20	36	1687.5	1650	0	10	25
7	1650	1512.5	0	10	20	37	2400	1537.5	0	10	25
8	2962.5	1150	0	10	20	38	862.5	2575	0	10	25
9	625	2562.5	0	10	20	39	1262.5	2075	0	10	25
10	2137.5	1987.5	0	10	20	40	2712.5	1737.5	0	10	25
11	1262.5	2637.5	5	15	25	41	2000	2350	0	20	30
12	875	1125	5	15	25	42	1612.5	2250	0	20	30
13	725	1450	5	15	25	43	1487.5	2225	0	20	30
14	1375	1850	5	15	25	44	2350	2600	0	20	30
15	1375	1850	5	15	25	45	1837.5	2362.5	0	20	30
16	2087.5	1250	5	15	30	46	1862.5	1425	0	30	40
17	1450	2700	5	15	30	47	2400	2375	0	30	40
18	2187.5	625	5	15	30	48	1437.5	1362.5	0	30	40
19	675	1975	5	15	30	49	325	2762.5	0	30	40
20	87.5	1125	5	15	30	50	750	1512.5	0	30	40
21	2300	2062.5	10	20	35	51	2762.5	2625	10	30	35
22	2800	2325	10	20	35	52	1925	1887.5	10	30	35
23	2650	2000	10	20	35	53	2250	2175	10	30	35
24	2825	2425	10	20	35	54	2012.5	1750	10	30	35
25	1887.5	1137.5	10	20	35	55	1825	2050	10	30	35
26	2675	1325	10	20	40	56	2350	1725	20	50	60
27	1512.5	1362.5	10	20	40	57	1462.5	2387.5	20	50	60
28	2625	2262.5	10	20	40	58	1800	1050	20	50	60
29	2625	2262.5	10	20	40	59	1687.5	1650	20	50	60
30	912.5	687.5	10	20	40	60	2275	737.5	20	50	60

Table 3. Total cost composition of the delivery mode based on Crowdsensing

Test number	F_1	F_2	F_3	F_4	Total cost
1	400	7.4554	76.7967	10.1629	494.4149
2	400	9.9056	73.4302	15.6948	499.0307
3	400	8.8002	77.6157	17.3641	503.7799
4	400	10.4566	69.4471	16.7056	496.6092
5	400	9.1401	77.1157	12.3911	498.6469
6	400	15.1066	74.1857	9.2747	498.567
7	400	11.2515	74.8855	10.1257	496.2627
8	400	13.5022	74.3717	17.6689	505.5427
9	450	4.7183	73.3156	0	528.0339
10	450	11.6734	69.972	0	531.6454

Table 4. Optimal cost components of the GA algorithm and TDGA algorithm in different datasets

Algorithm	DS	F_1	F_2	F_3	F_4	Total cost
GA	A	450	9.9551	77.445	263.5103	750.9104
TDGA	A	400	7.4554	76.7967	10.1629	494.4149
GA	B	800	22.733	176.0187	1703.6079	2702.3596
TDGA	B	800	35.9267	188.4217	733.0232	1757.3716

Therefore, the TDGA algorithm has evident advantages in solving this model. In addition, the TDGA algorithm has more prominent advantages when solving large-scale problems. Figure 4 and 5 shows the total cost of the two algorithms with the number of iterations: According to the Figure 4 and Figure 5, in both dataset A and dataset B simulation experiments, the TDGA algorithm had a better performance than the GA algorithm when optimizing to minimize the total delivery cost of merchants. In Figure 4, the TDGA algorithm achieved stability in the 190th generation. Meanwhile, the GA algorithm achieved stability in the 430th iteration. As the number of iterations increased, the total cost of the TDGA algorithm remained lower than the total cost of the GA algorithm. Simultaneously, by increasing the data size and conducting a second comparison, as shown in Figure 5, the cost of the TDGA algorithm was higher than the cost of the GA algorithm at the beginning of the iteration. Nevertheless, it immediately decreased to lower than the cost of the GA algorithm. The final results showed that the TDGA algorithm had good convergence speed and stability.

In summary, the TDGA algorithm not only provides better solutions but also overcomes the limitations of the traditional GA algorithm. After increasing the data size, the TDGA algorithm can obtain better results in terms of solution quality. Therefore, it is more suitable for solv-

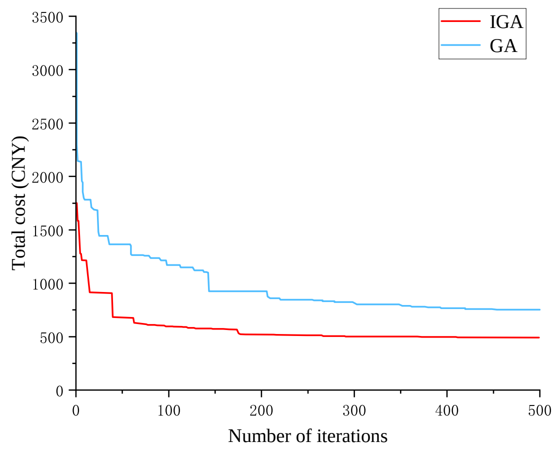

Fig. 4. Comparing the total cost of two algorithms with the number of iterations in dataset A

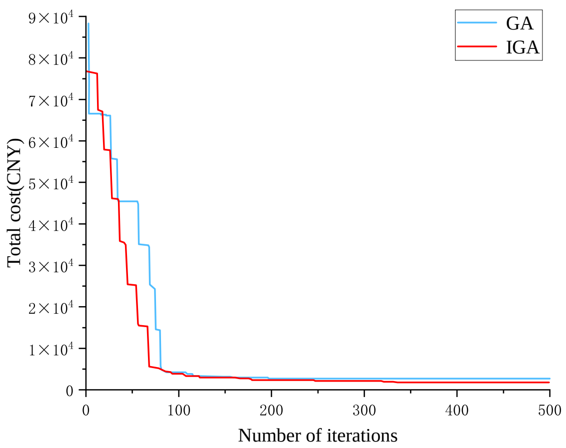

Fig. 5. Comparing the total cost of two algorithms with the number of iterations in dataset B

ing the path planning problem of large-scale orders in the delivery scenario.

6. Conclusion and future work

In this study, a takeaway delivery mode based on Crowdsensing was proposed to solve delivery delays in the current takeaway delivery system. Because this mode considers factors such as multiple riders, time windows, and open delivery, any conventional algorithms cannot be used. Hence, the TDGA was designed to solve this issue. The contributions of this paper can be summarized as follows:

(1)To address order delivery delays, a takeaway delivery mode based on Crowdsensing was proposed.

(2)To decrease the total merchant cost, the MROVRPTWPD model was constructed.

(3)To address the abovementioned path planning problem model, the TDGA was designed.

(4)The efficacy and superiority of the abovementioned method have been confirmed via extensive experiments.

The takeaway delivery mode based on Crowdsensing and the TDGA were tested and found to be effective. When tested on dataset A, the total cost of the TDGA algorithm decreased by 34.16%. In dataset B, the total cost decreased even higher at 56.97%.

However, more efforts should be exerted. That is, the vehicle path planning problem should be modeled under various dynamic constraints [19][20]. Moreover, this work must be extended to larger practical applications

References:

[1] Chang Y S, Lee H J. Optimal delivery routing with wider drone-delivery areas along a shorter truck-route. *Expert Systems with Applications*, 2018, 104:307−317.

[2] Hou S, Gao J, Wang C. Optimization Framework for Crowd-Sourced Delivery Services With the Consideration of Shippers' Acceptance Uncertainties. *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[3] Sisi Z, Souri A. Blockchain technology for energy-aware mobile crowd sensing approaches in Internet of Things. *Transactions on Emerging Telecommunications Technologies*, 2021: e4217.

[4] Sahraoui Y, Kerrache C A, Amadeo M, et al. A cooperative crowdsensing system based on flying and ground vehicles to control respiratory viral disease outbreaks. *Ad Hoc Networks*, 2022, 124: 102699.

[5] Ali A, Qureshi M A, Shiraz M, et al. Mobile crowd sensing based dynamic traffic efficiency framework for urban traffic congestion control. *Sustainable Computing: Informatics and Systems*, 2021, 32: 100608.

[6] Wang B, Sun Y, Liu D, et al. Social-aware UAV-assisted mobile crowd sensing in stochastic and dynamic environments for disaster relief networks. *IEEE Transactions on Vehicular Technology*,2019,69(1):1070−1074.

[7] Dutta J, Pramanick P, Roy S. NoiseSense: Crowdsourced context aware sensing for real time noise pollution monitoring of the city. *2017 IEEE international conference on advanced networks and telecommunications systems (ANTS)*, 2017: 1−6.

[8] Foschini L, Martuscelli G, Montanari R, et al. Edge-enabled mobile crowdsensing to support effective rewarding for data collection in pandemic events. *Journal of Grid Computing*, 2021, 19: 1−17.

[9] Huang H, Hu C, Zhu J, et al. Stochastic task scheduling in UAV-based intelligent on-demand meal delivery system. *IEEE Transactions on Intelligent Transportation Systems*, 2021, 23(8): 13040−13054.

[10] Wang Y, Zhang D, Liu Q, et al. Towards enhancing the last-mile delivery: An effective crowd-tasking model with scalable solutions. *Transportation Research Part E: Logistics and Transportation Review*, 2016, 93: 279−293.

[11] FooDNet: Toward an optimized food delivery network based on spatial crowdsourcing. *IEEE Transactions on Mobile Computing*, 2018, 18(6): 1288−1301.

[12] Stochastic task scheduling in UAV-based intelligent on-demand meal delivery system. *IEEE Transactions on Intelligent Transportation Systems*, 2021, 23(8): 13040−13054.

[13] Xu, D, Li, K, Zou, X, Liu, L. An unpaired pickup and delivery vehicle routing problem with multi-visit. *Transportation Research Part E*, 103, 218−247.

[14] Tkatek S, Bahti O, Lmzouari Y, et al. Artificial intelligence for improving the optimization of NP-hard problems: a review. *International Journal of Advanced Trends Computer Science and Applications*, 2020, 9(5).

[15] Zhang S, Chen M, Zhang W, et al. Fuzzy optimization model for electric vehicle routing problem with time windows and recharging stations. *Expert systems with applications*, 2020, 145: 113123.

[16] Park H, Son D, Koo B, et al. Waiting strategy for the vehicle routing problem with simultaneous pickup and delivery using genetic algorithm. *Expert Systems with Applications*, 2021, 165: 113959.

[17] Niu Y, Yang Z, Chen P, et al. Optimizing the green open vehicle routing problem with time windows by minimizing comprehensive routing cost. *Journal of cleaner production*, 2018, 171: 962−971.

[18] Bent R, Van Hentenryck P. A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers & Operations Research*, 2006, 33(4): 875−893.

[19] Fonseca-Galindo, J. C., de Castro Surita, G., Neto, J. M., de Castro, C. L., & Lemos, A. P. (2022). A multi-agent system for solving the dynamic capacitated vehicle routing problem with stochastic customers using trajectory data mining. *Expert Systems with Applications*, 195, 116602.

[20] Wang, Y., Zhe, J., Wang, X., Fan, J., Wang, Z., & Wang, H. (2022). Collaborative multicenter reverse logistics network design with dynamic customer demands. *Expert Systems with Applications*, 206, 117926.