

Article

Not peer-reviewed version

Dynamic Scheduling Optimization of Automatic Guide Vehicle for Terminal Delivery under Uncertain Conditions

[Qian Qian Shao](#), [Jia Wei Miao](#)^{*}, [Peng Hui Liao](#), [Tao Liu](#)

Posted Date: 16 August 2024

doi: 10.20944/preprints202408.1160.v1

Keywords: AGV; take-out terminal delivery; rolling scheduling; stochastic programming; routing combination; NSGA-III



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: 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.

Article

Dynamic Scheduling Optimization of Automatic Guide Vehicle for Terminal Delivery under Uncertain Conditions

Qian Qian Shao, Jia Wei Miao *, Peng Hui Liao and Tao Liu

Shenyang Jianzhu University

* Correspondence: 3231135293@qq.com

Abstract: As an important part of urban terminal delivery, automated guided vehicle (AGV) has been widely used in the field of takeout delivery. Due to the real-time generation of takeout orders, the delivery system is required to be extremely dynamic, so the AGV needs to be dynamically scheduled. At the same time, the uncertainty in the delivery process (such as the meal preparation time) further increases the complexity and difficulty of AGV scheduling. Considering the influence of these two factors, the method of embedding stochastic programming model into rolling mechanism is adopted to optimize the AGV delivery routing. Specifically, To handle real-time orders under dynamic demand, an optimization mechanism based on rolling scheduling framework is proposed, which allows the AGV route to be continuously updated. Different from most VRP models, open chain structure is used to describe the dynamic delivery path of AGV. In order to deal with the impact of uncertain meal preparation time on route planning, a stochastic programming model is formulated with the purpose of minimizing the expectation of order timeout rate and the total customer waiting time. In addition, an effective path merging strategy and after-effects strategy are also considered in the model. In order to solve the proposed mathematical programming model, a multi-objective optimization algorithm based on NSGA-III framework is developed. Finally, a series of experimental results demonstrate the effectiveness and superiority of the proposed model and algorithm.

Keywords: AGV; take-out terminal delivery; rolling scheduling; stochastic programming; routing combination; NSGA-III

1. Introduction

Because of its convenient and fast characteristics, take-out has become an essential part of the daily life of the working class. According to the data, the market size of online take-out terminal delivery industry (China) will grow to 941.74 billion yuan in 2022, with a year-on-year growth of 19.8% [1]. Such large-scale market growth will inevitably bring huge terminal delivery [2], which is also an important reason why AGV is widely used in the field of take-out delivery. However, effective AGV scheduling can improve efficiency, reduce costs, and enhance enterprise competitiveness [3]. Therefore, as an important part of food delivery, AGV scheduling can effectively improve the service level of customers and the competitiveness of food delivery companies.

In the process of AGV delivery, it will face off peak hour and peak hour [4–6]. In the off peak hour, AGV can basically meet the deliver demand due to the relatively smaller order volume. However, in the peak hour, due to the rapid expansion of order size, the number of AGVs could not meet the surging delivery demand, which in turn led to the extension of customer waiting time. In addition, long delivery times can reduce customer trust in the platform. Although increasing the number of AGVs can meet the delivery demand during peak hours, there will be excessive idle AGVs during off-peak hours, which will lead to a significant increase in costs.

Due to the real-time and random nature of online orders, AGVs need to be constantly dynamically scheduled for route scheduling during the delivery and pick-up. Because the takeout platform adopts rolling scheduling framework based on order pool to assign orders to delivery personnel [18–23,25,26], the route of AGV delivery is also optimized on a rolling basis. From the microscopic route structure level, the insertion of new orders is always carried out in the process of AGV operation, and this insertion will inevitably have an impact on the planned route of AGV. Therefore, in dynamic scheduling, most routes based on previous decisions cannot be fully executed.

Due to variety of complex factors, the meal preparation time of the merchant is uncertain. At the same time, the time when the AGV will reach the customer is also uncertain. Therefore, this uncertainty will cause difficulties for AGV scheduling. In order to improve delivery efficiency, the uncertainty of meal arrive time should be fully considered in the route planning of AGV.

In practice, combined meal pickup and combined meal delivery are widely used as operational modes to improve delivery efficiency. Specifically, different orders with a small order time span and the same pickup location can be combined for pickup; Different orders with the same customer can be combined for delivery. This mode of operation can significantly reduce the number of redundant driving routes. Obviously, route consolidation increases the difficulty of AGV route planning.

The remainder of this paper is organized as follows. In Section 2, the relevant literatures are introduced. In Section 3, detail problems are described and a mathematical description of the model is given. We then outline our metaheuristic in Section 4. In Section 5 presents results from numerical experiments, and the last section provides conclusions and future work.

2. Literature Review

At present, the academic research on AGV scheduling mainly focuses on the application scenarios of manufacturing industry, warehouse, workshop and container terminal, and its application in takeout delivery has not been any research results. Considering the similarity of scheduling principle, AGV route planning model in other fields still has reference significance in the field of take-out delivery. On the other hand, the field of non-AGV delivery is rich in research results related to delivery models such as vehicle routing problem (VRP). Therefore, in terms of improving delivery efficiency and delivery quality, it still has reference value for AGV takeout delivery.

The transportation organization models discussed in the current research results mainly fall into two categories: one is the centralized transport organization model with unified cargo delivery hub, which mainly adopts the VRP variant model for scheduling optimization. The other type is the decentralized transport organization model without unified cargo delivery hub, which mainly adopts non-VRP variant model for scheduling optimization.

In recent years, most scholars have used VRP variant models to conduct research on pick up and delivery route optimization [6–19,24,25,27]. In the field of AGV research, reducing the driving distance and the delivery cost are the core issues [8–17]. In order to improve delivery efficiency, Cheng et al. [9] and Gao et al. [10] set the minimum completion time as the goal and proposed a solution for the multiple AGVs dispatching problem (AGVDP). Chen et al. [12] focused on optimizing the material handling and delivery processes of AGVs, aiming to minimize energy consumption and increase customer satisfaction. In the study of non-AGV takeout delivery, although there have been studies on takeout routing planning based on known order information [6], these methods are mainly applicable to pre-order (static situation) and are difficult to cope with real-time changes in order demand. In order to deal with the randomness of customer ordering and the uncertainty of meal preparation time, Ulmer et al. [7] proposed a time buffer using parametric cost function approximation (CFA). Zheng et al. [27] adopted the fuzzy set theory to obtain the time interval for meal preparation. In addition, considering the randomness of customer demand, Reyes et al. [18] and Yildiz et al. [19] adopted rolling scheduling decision for takeout delivery under uncertain demand environment (i.e., randomness of customer order).

During the peak hours, some scholars used non-VRP variant models to conduct research on pick up and delivery route optimization [5,20–23,26]. However, at present, there is no research using non-VRP model to solve AGV scheduling problem. As for the study of non-AGV takeout delivery, Liu et

al. [5] studied the optimization of crowdsourcing delivery route in a deterministic environment, which did not consider dynamic scenarios. Rolling scheduling decisions are more suitable in dynamic situations. Considering the random generation of order demand and carrying passengers, Du et al. [20] proposed a crowdsourcing framework for taxi carrying passengers and delivering takeout at the same time. The framework adopts the method of rolling cycle repeat matching for passengers and food parcels. Considering the real-time dynamics of the order and the inertial action of delivery when the algorithm is involved, Zachary et al. [21] planned the pick up and delivery path based on the rolling framework. Sun et al. [22] established the coupling relationship between order allocation and path planning based on the delivery process through the real-time dynamics of order allocation and path optimization. A dynamic optimization algorithm is designed to solve the model. In order to optimize delivery routes, Tu et al. [23] proposed an online dynamic optimization framework with order collection time window based on order dynamics and delivery status. Yu et al. [26] established a real-time order allocation and path optimization model by taking into account the real-time and timeliness of orders.

Based on demand uncertainty. In the study of AGV, Liu et al. [17] solved the scheduling problem of AGV by considering the uncertainty of the replenishment time of workstation. However, in the study of non-VRP takeout delivery, only a small number of scholars have considered the uncertainty factors in the online order delivery problem [7,24,27]. Considering the randomness of customer orders and the uncertainty of food preparation time, a time buffer setting using parametric cost function approximation (CFA) was proposed by Ulmer et al. [7]. In order to deal with the uncertainty of meal preparation time of merchants, Zheng et al. [24] used the fuzzy set theory to obtain the time interval for meal preparation. Zheng et al. [27] established a Gaussian mixture model combining the distribution of uncertain service time. A hybrid distribution estimation algorithm is developed to solve the problem.

In the field of takeout delivery, order combination refers to the combined pickup and combined delivery of multiple orders. At present, order combination is only discussed in the centralized transportation organization model. On this basis, this paper introduces the concept of action combination. In order to optimize the efficiency of delivery, order combination is considered under the decentralized transportation organization model. In the academic research of AGV, the principle of one single match is mainly adopted, but the related content of action combination has not been fully considered. However, in the field of non-AGV takeout delivery, scholars have paid attention to the issue of order combination (i.e., bundling and unified distribution of multiple orders) [7,18,19,25]. In addition, Reyes et al. [18] and Yildiz et al. [19] considered the combined take-up of action combination in VRP mode. Liao et al. [25] proposed to combine delivery a group of orders that meet the time window and load requirements. At the same time, Principal Component Analysis (PCA) and K-means cluster were adopted to decide whether to combine orders.

In route planning, after-effect refers to the consideration of the impact of the previous decision on the current decision. For the study of AGV, no scholars have specifically discussed the issue of after-effect. In the field of non-AGV takeout delivery, some scholars have paid attention to the importance of after-effect. By predicting the distribution of order demand in the future period, they consider the future order demand in the scheduling process [7,21]. Based on the order quantity data of each restaurant, Ulmer et al. [7] set a time buffer to consider future order demand and generate the probability of customers placing orders. Zachary et al. [21] studied the fairness and decentralized decision making of delivery personnel to accommodate the problem of matching new orders with delivery personnel.

Based on the above description of existing study, we integrate our research work into Table 1. In Table 1, "One-time decision" refers to not considering whether a new customer appears, but only considering whether the customer's needs fluctuate. "Rolling decision" means to consider not only the dynamics, but also the inertia routings of delivery personnel. Y means to consider the uncertain factors of the meal preparation time of restaurants, N means not to consider the uncertain factors of the meal preparation time of restaurants. c represents order combination, i.e. under the centralized transportation organization model, C represents action combination, that is, under the decentralized

transportation organization model. *f* represents the scheduling after-effect based on prediction. *F* represents the scheduling after-effect based on no prediction. *S* represents a single objective, *S** represents the normalization of multiple objectives into a single objective, *M* represents multiple objectives.

According to Table 1, the following conclusions can be made: (1)Aiming at the uncertainties in takeaway delivery, some scholars adopt stochastic programming to make one-time scheduling decisions. At present, there is no relevant research to deal with uncertain scheduling optimization under dynamic conditions based on rolling framework. (2)As for the combination problem in takeout delivery scheduling, the existing studies only discussed the order combination under the centralized transportation organization mode, while the action combination under the decentralized transportation organization mode was not mentioned. (3)For the consideration of after-effect, the existing results are based on statistical models for demand forecasting. This method performs well in macro order demand forecasting, but is difficult to predict micro discrete events (e.g., customers, meal delivery times, etc.). Considering the above limitations, the innovations of this paper are as follows: (1)In order to achieve AGV scheduling optimization, a stochastic programming model is embedded in the rolling frame to solve the uncertain control problem under dynamic conditions in takeout delivery. (2)According to the characteristics of decentralized transportation organization, this paper proposes a method of integrating open chain structure and order combination mechanism, so that AGV can realize action combination in the process of dynamic route updating. (3)An after-effect control method is proposed, which does not rely on historical data and demand forecasting. At the same time, the delivery system can adapt to the changes of dynamic environment through flexible real-time adjustment.

Table 1. related literatures.

Model type	author	model	scheduling framework	uncertain times	action consolidation	look-forward	Multi-objective/ Single-objective
	Xin et al. (2020) [8]	MVPD	One-time decision	N			S
	Cheng et al. (2023) [9]	MAGV-SP	One-time decision	N			S*
	Gao et al. (2023) [10]	MADP	One-time decision	N			S*
	Qiu et al. (2015) [11]	EHARP	One-time decision	N			S*
	Chen et al. (2023) [12]	AGVEESC	One-time decision	N			M
	Xin et al. (2022) [13]	ARP	One-time decision	N			
	Zou et al. (2020) [14]	AGVDP	One-time decision	N			S*
	Li et al. (2018) [15]	AGVSP	One-time decision	N			S*
	Wang et al. (2022) [16]	MARP	One-time decision	N			S
	Liu et al. (2023)[17]	AGVDP	One-time decision	Y			M

	Xue et al. (2021)[6]	The pickup and delivery problem,	One-time decision	N			S
	Ulmer et al. (2021)[7]	DPDP, SDDP	One-time decision	Y	c	f	S
	Reyes et al. (2018)[18]	MDRP	Rolling decision	N	c		S
	Yildiz et al. (2019)[19]	DVRPs	Rolling decision	N	c		S
	Zheng et al. (2020) [24]	PDP	One-time decision	Y			S*
	Liao et al. (2020)[25]	GMDRP,	Rolling decision	N	c		M
	Zheng et al. (2022) [27]	on-demand food delivery	One-time decision	Y			S*
Non-VRP	Liu et al. (2018) [5]	Spatial CrowdSourcing	One-time decision	N			S; S*
	Du et al (2019)[20]	Spatial CrowdSourcing	Rolling decision	N			M
	Zachary (2019) [21]	VFCDP	Rolling decision *	N		f	S
	Sun et al. (2020) [22]	TRDR	Rolling decision	N			S
	Tu et al. (2019)[23]	Spatial CrowdSourcing	Rolling decision	N			S*
	Yu et al (2022)[26]	DRVRP	Rolling decision	N			M
	本文		Rolling decision*	Y	C	F	M

3. Problem Formulation

3.1. Problem Description

3.1.1. Take-Out Delivery Flow

The take-out delivery process involves four stakeholders: customers, AGV, and restaurants. The connection among them is established by the take-out platform. Figure 1 shows the flow chart of take-out delivery. Customers place orders on the take-out platform through mobile APP (Step 1, Information flow). After the platform receives orders, it sends the order information to the restaurant (Step 2, information flow). The restaurants provide feedback to the platform on whether to accept the order in a short period of time. (Step 3, information flow). After receiving feedback, the platform sends assigns orders to appropriate AGV (Step 4, information flow). The corresponding AGV goes to the restaurant nodes to collect the food based its working state (Step 5, physical flow). Then, the AGV delivers the food to the customer's location within a period (Step 6, physical flow). Finally, they provide feedback to the take-out scheduling platform (Step 7 information flow).

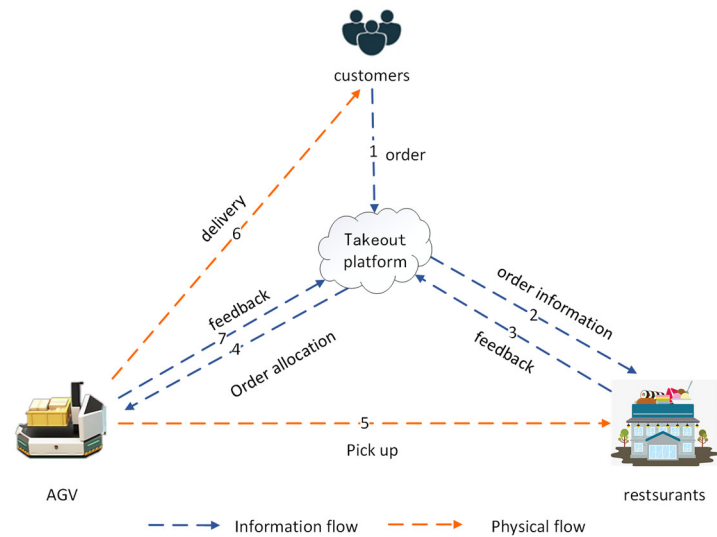


Figure 1. The order dispatching process.

3.1.2. Real-Time Dynamic and Rolling Scheduling

As universally acknowledged, the order time in take-out delivery exhibits significant variability. The AGV scheduling platform needs to continuously receive and assign orders.

Therefore, it is necessary to adjust the AGV's path constantly. In other words, when a new order is inserted, the algorithm should intervene to re-plan the allocation path of the AGVs. The insertion time of the new order is called the decision time of rolling scheduling. AGV should not immediately change the current action when performing delivery operations upon receipt of a new order to ensure consistency of operation. This action being carried out by the AGV is called inertia action. Therefore, after each rolling schedule, the starting time of the readjusting path should be the end time of the inertial action. It is important to note that the completion of the current inertia of the AGV does not equal the completion of the current order. Figure 2 shows an example of the order task sequence and routing combination based on the rolling scheduling framework.

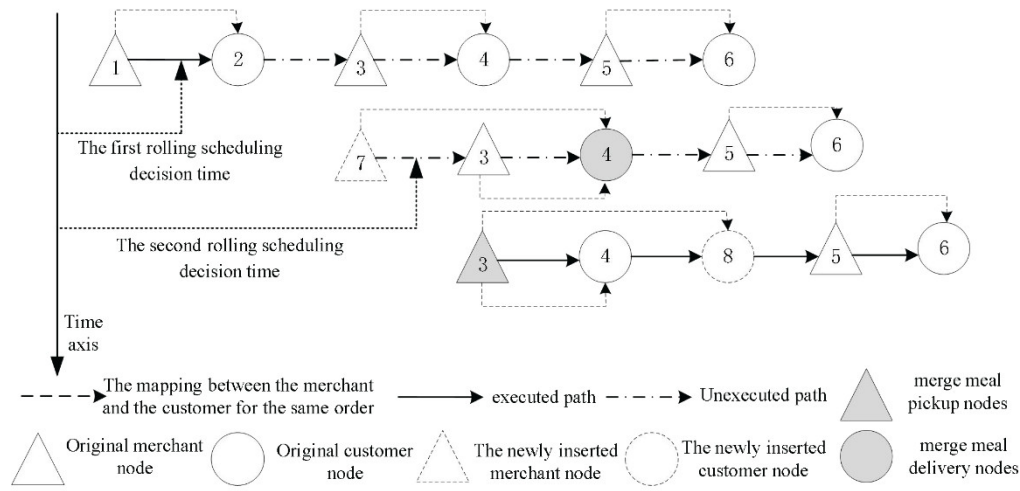


Figure 2. Rolling pick-up and delivery path sequence and routing combination.

3.1.3. Uncertain Preparation Times

The process of AGV completing delivery tasks involves numerous uncertain factors, including traffic resistance, meal preparation time of restaurants, time of AGV serving customers, etc. Among them, the uncertainty of meal preparation time of restaurants is the factor that has the greatest impact on the system [24]. Figure 3 shows the difference between the expected delivery time and the actual

delivery time. PT1 represents the travel time for AGV to pick up order 1 from restaurant node 1. PT2 represents the travel time for AGV to pick up order 2 from restaurant node 2. DT1 represents the travel time of AGV to customer node 1. DT2 represents the travel time of AGV to customer node 2. WT1 represents the waiting time of AGV at restaurant node 1. WT2 represents the waiting time of AGV at restaurant node 2. ST represents the time of AGV serving customers. If the food arrives at the restaurant node too late, the waiting time of the AGV at the restaurant node will become longer, which will lead to the order timeout, that is, the actual arrival time of the AGV is later than the latest delivery time of the order, and the customer satisfaction will decline. Therefore, it is necessary to consider the uncertainty of the arrival time, which can better guarantee the robustness of the system.

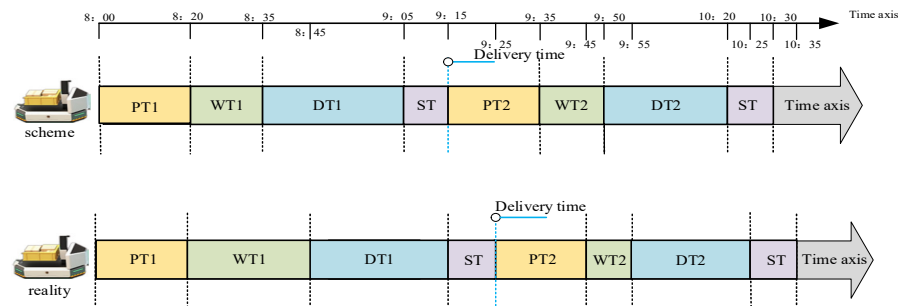


Figure 3. Comparison of expected delivery time and actual delivery time.

3.1.4. Routing Combination

In terms of order delivery, for different orders with a small order time span and the same restaurants node, the AGV can wait slightly to pick up the food together; For different orders with the same customer involved, the AGV can unify the collection and delivery. This mode of operation is known as action consolidation (AC). Figure 4 shows the rolling pick-up and delivery path considering routing combination. The original path of the AGV is 1-2-3-4-5-6. A new order is received at customer node 2 (a new restaurant node 7 is inserted). Since restaurant node 7 and restaurant node 3 correspond to the same customer node 4, the AGV will merge food delivery at customer node 4. After the first rolling dispatching, the path is changed to 1-2-7-3-4-5-6 (the executed path is 1-2). A new order is received on restaurant node 7 (a new customer node 8 is inserted). Since customer node 8 and customer node 4 correspond to the same restaurant node 3, the AGV merges food pick-up at the customer nodes 3. After the second rolling scheduling, the path is changed to 1-2-7-3-4-8-5-6 (the executed path is 1-2-7). Although taking AC decision into account in route planning will increase the complexity of optimization, it can significantly improve delivery efficiency and customer satisfaction.

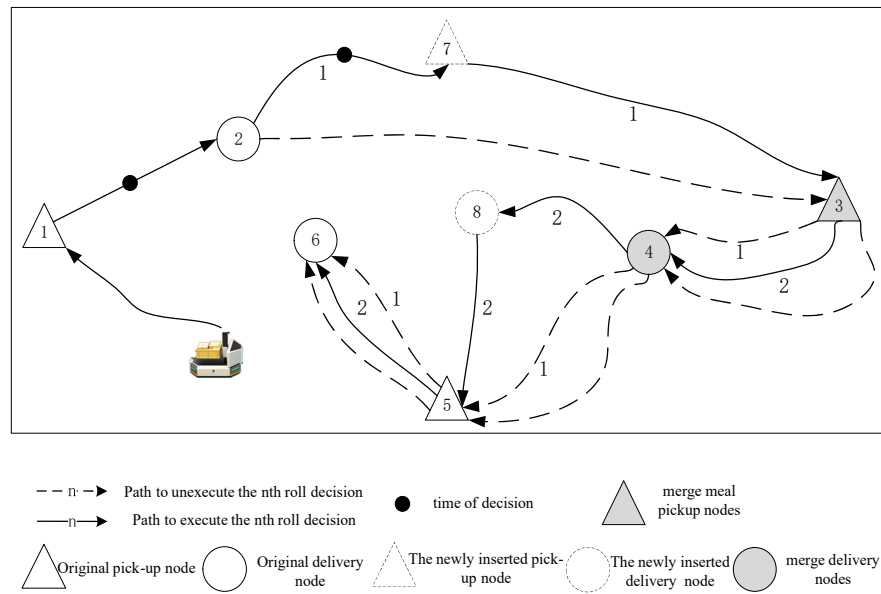


Figure 4. Rolling pick-up and delivery path considering routing combination.

3.1.5. Look-Forward

In real life, route planning is often considered based on look-forward. AGV generally gives priority to the delivery of orders that are closer to their current location, and orders that are relatively far away will be slightly shelved. The purpose is to wait for future orders that may be able to combine with current orders.

3.2. Modeling

To solve the problem studied, the following multi-objective rolling pick-up and delivery route planning model (MO-RPDRP) is developed.

The model is based on the following assumptions:

- The inertial action currently being executed cannot be changed.
- The preparation time of the restaurant is subject to Gaussian distribution.
- The AGV's meal pick-up time occupies a small amount of time in the entire delivery process, so it is not considered.

In MO-RPDRP model, N represents the order set involved in the current rolling period. N_1 represents the current set of orders that have been picked up but not delivered. If the inertial action of AGV is food picking up, the order involved is also included N_1 . If the inertial action is food delivery, the order is considered completed, it is not included in N_1 . N_2 represents the current order set that have not been picked up. I represents the set of restaurant nodes. J Represents the set of customer nodes. A represents the set of all nodes. If the order is one-to-one mapping to both the restaurant nodes and the customers, $|N_1| + 2|N_2| = |A|$. If there are multiple orders for the same restaurant node or multiple orders for the same customer, $|N_1| + 2|N_2| > |A|$. If AC strategy is not executed, the length of pick-up and delivery path is $|N_1| + 2|N_2|$. Otherwise, the actual length of pick-up and delivery path is in the interval $[|A|, |N_1| + 2|N_2|]$. To make orders and restaurant nodes, orders and customers are mapped one-to-one, the virtual restaurant nodes and virtual customer nodes need to be set. The set of virtual restaurant nodes is represented as $I^{virtual}$, it and the set I forms the set of restaurant node expansion I' , $I' = I \cup I^{virtual}$. The number of the virtual restaurant node set $I^{virtual}$ continues with the

number of the restaurant node set I . The set of virtual customer nodes is represented as $J^{virtual}$, it and the set J forms the set of customer node expansion J' , $J' = J \cup J^{virtual}$, the number of virtual customer node set $J^{virtual}$ continues with the number of the customer node set J . The set of all node expansion is represented as $A' = I' \cup J'$. For $n \in N_1$, it has a unique customer node corresponding to it. For $n \in N_2$, the only restaurant node can be found, and it has a customer node corresponding to it. The length of the virtual open chain (pick-up and delivery path) is $|A'|$.

The essence of MO-RPDRP decision is the sequence of all nodes in the set A' and the identification of AC strategy. The decision variable x_{ij} is set as 0-1 variable. If the AGV travels from node i to node j , the decision variable x_{ij} is equal to 1. Otherwise, it is equal to 0. Decision variable x_{ij} can execute AC strategy while deciding the sequence of virtual nodes. The identification of AC strategy is reflected by 0-1 variables p_i^1 and p_j^2 . Other sets, parameters and variables are shown in Table 2 for details.

Table 2. Sets, parameters, and variables.

Sets	description
N_1	the current set of orders that have been picked up but not delivered
N_2	the current order set that has not been picked up
N	set of orders involved in the current rolling period, $N = N_1 \cup N_2$, $n \in N$
I	set of restaurant nodes, $I = \{1, 2, \dots, I \}$
$I^{virtual}$	set of virtual restaurant nodes, $I^{virtual} = \{ I + 1, I + 2, \dots, N_2 \}$
J	set of customer nodes, $J = \{ N_2 + 1, N_2 + 2, \dots, N_2 + J \}$
$J^{virtual}$	set of virtual customer nodes, $J^{virtual} = \{ N_2 + J + 1, N_2 + J + 2, \dots, N_1 + 2 N_2 \}$
A	set of all nodes, $A = I \cup J$
I'	set of restaurant nodes expansion, $I' = I \cup I^{virtual}$
J'	set of customer nodes expansion, $J' = J \cup J^{virtual}$
A'	set of all nodes expansion, $A' = I' \cup J'$
map_i^1	virtual (actual) restaurant node i mapping of the actual restaurant node, $i \in I'$
map_j^2	virtual (actual) customer node j mapping of the actual customer node, $j \in J'$
α_n	the restaurant node number corresponding to the order n , if the restaurant node number corresponding to the order n is i , then there is $i = \alpha_n$, and there is an inverse mapping $n = \arg \alpha(i)$, $i \in I', n \in N$
β_n	the customer node number corresponding to the order n , if the customer node number corresponding to the order n is j , then there is $j = \beta_n$, and there is an inverse mapping $n = \arg \beta(i)$, $j \in J', n \in N$

parameters	
V	maximum capacity of delivery vehicle
t_{ij}^d	travel time from node i to node j , $i \in I', j \in J'$
s_0	number of the initial node
T_0	the ready time of the AGV
r_n	order time for order n , $n \in N$
b_n	the latest delivery time for order n , $n \in N$
v_n	the volume occupied by commodities for order n , $n \in N$
t_n^p	meal preparation time for order n , it follows the Gaussian distribution, $t_n^p \sim (\mu_i, \sigma_i^2)$, $n \in N$
t_n^s	customer service time for order n , $n \in N$

Decision variable	
x_{ij}	if the deliverer travels from node i to node j , the decision variable is equal to 1, otherwise, x_{ij} is equal to 0, $i \in A' \cup s_0, j \in A'$
Other variable	
s_l	the number of the l th node to be visited, if the sequence of the node i in the path is l , $i = s_l$, and there is an inverse mapping $l = \arg s(i)$, $l \in \{1, 2, \dots, A' \}$, $i \in A'$
p_l^1	if the l th node and the next node merge to pick-up food, p_l^1 is equal to 1, otherwise, p_l^1 is equal to 0, $l \in \{1, \dots, A' -1\}$
p_l^2	if the l th node and the next node merge to deliver food, p_l^2 is equal to 1, otherwise, p_l^2 is equal to 0, $l \in \{1, \dots, A' -1\}$
a_l	time to arrive s_l , $l \in \{1, 2, \dots, A' \}$
d_l	take-out volume carried on arrival at s_l , $l \in \{1, 2, \dots, A' \}$
λ_l	the number of unfinished orders before visit s_l , $l \in \{1, 2, \dots, A' \}$
g_n	the completion time of order n , $n \in N$
ω	the timeout rate of the order

Objective function 1 minimizes the expected order timeout rate, ω is the timeout rate for all orders.

$$\min F_1 = E(\omega) \quad (1)$$

Objective function 2 minimizes the expected total customer waiting time. From the perspective of the customer, the customer needs the AGV to deliver the meals as soon as possible. The derivation of the completion time g_n of the order n is shown in (14).

$$\min F_2 = E\left(\sum_{n \in N} (g_n - r_n)\right) \quad (22)$$

Objective function 3 maximizes scheduling after-effect. A look-forward objective is introduced to adjust the impact of this decision on subsequent scheduling. See 3.1.5 for detailed description. In terms of scheduling, orders are completed intensively in the early stage, while orders are completed relatively dispersed in the later stage. In the objective function, it is expressed as (3). Figure 5 shows the relationship between the number of remaining nodes and the number of unfinished orders.

$$\min F_3 = E\left(\sum_{l=1}^{|A'|} \lambda_l (a_l - a_{l-1})\right) \quad (3)$$

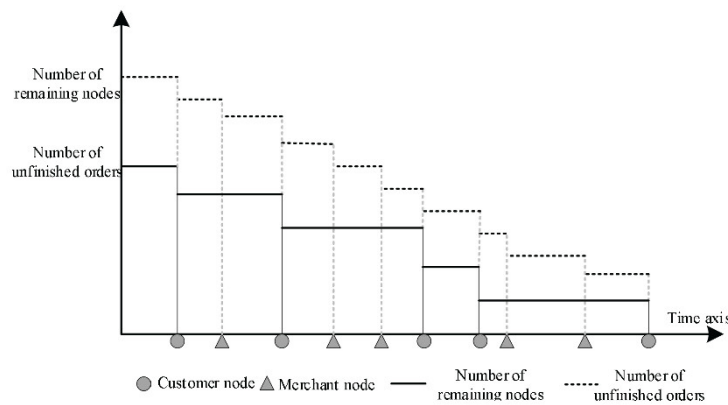


Figure 5. Changes in the number of orders and nodes to be delivered.

Constraints (4) and (5) delineate deduction constraints governing the sequential visitation order of order nodes. The access sequence s_l of the l th node is deduced by decision variables x_{ij} . Obviously, $s_{|A'|}$ is the last access node.

$$s_1 = \sum_{j \in A'} jx_{s_0j} \quad (4)$$

$$s_{l+1} = \sum_{j \in A'} jx_{s_lj}, \forall l \in \{1, 2, \dots, |A'| - 1\} \quad (5)$$

Constraints (6) and (7) are constraints on the volume of take-out carried by the AGV. The volume of take-out carried upon arrival cannot exceed the maximum capacity of the AGV. If the AGV passes through the restaurant node s_l (pick-up stage), the take-out volume is increased. The take-out volume d_{l+1} carried by the AGV upon arrival s_{l+1} is the sum of d_l and $v_{\arg \alpha(s_l)}$. If the AGV passes through the customer node s_l (the delivery stage), the take-out volume is reduced. The takeaway volume d_{l+1} carried by the AGV on arrival s_{l+1} is the difference between d_l and $v_{\arg \alpha(s_l)}$.

$$d_0 = \sum_{n \in N_1} v_n$$

$$d_{l+1} = \begin{cases} d_l + v_{\arg \alpha(s_l)} & s_l \in I' \\ d_l - v_{\arg \beta(s_l)} & s_l \in J' \end{cases}, \forall l \in \{1, \dots, |A'| - 1\} \quad (6)$$

$$d_l \leq V, \forall l \in \{1, \dots, |A'| - 1\} \quad (7)$$

Constraint (8) imposes limitations on the visiting sequence of order nodes. For the same order, the AGV should visit the restaurant node α_n first, then the customer node β_n is visited. It is expressed as that the time when the AGV arrive at the restaurant node $a_{\arg s(\alpha_n)}$ should be less than the time when they arrive at the customer node $a_{\arg s(\beta_n)}$.

$$a_{\arg s(\alpha_n)} < a_{\arg s(\beta_n)}, \forall n \in N \quad (8)$$

Constraints (9), (10) and (11) are the AC identification constraints of order.

$$p_l^1 = \begin{cases} 1, & \text{map}_{s_l}^1 = \text{map}_{s_{l+1}}^1, s_l, s_{l+1} \in I', \forall l \in \{1, \dots, |A'| - 1\} \\ 0 & \end{cases} \quad (9)$$

$$p_l^2 = \begin{cases} 1, & \text{map}_{s_l}^2 = \text{map}_{s_{l+1}}^2, s_l, s_{l+1} \in J', \forall l \in \{1, \dots, |A'| - 1\} \\ 0 & \end{cases} \quad (30)$$

$$p_l^1 + p_l^2 \leq 1, \forall l \in \{1, \dots, |A'| - 1\} \quad (41)$$

Constraint condition (14) is the constraint of the completion time of the order, g_n is the sum of $a_{\arg s(\beta_n)}$ and t_n^s . The derivation of a_l is shown in (13). If there is a combination of routings between s_l and s_{l-1} (merged meal pick-up or delivery), a_{l-1} is equal to a_l . If s_{l-1} is the restaurant node, which does not produce merged meal pick-up with the last node, a_l is the sum of a_{l-1} and $t_{s_{l-1}s_l}^d$. If s_{l-1} is the restaurant node, which produces merged meal pick-up with the last node, set_l is expressed as the sequence set of s_{l-1} and last merged meal restaurant node. The expression is (12). set_l is the sequence set of s_{l-1} and last merged meal restaurant node. a_{l-1} is the maximum sum of the ordering time in set_{l-1} and the meal preparation time of the restaurant. The restaurant's meal preparation time $t_{\arg \alpha(s_w)}^p$

follows the Gaussian distribution. In other words, for different restaurants, they obey the Gaussian distribution with different expectation μ_i , and variance σ_i^2 . If s_{l-1} is the customer node, it does not produce merged meal delivery with the last node, a_l is the sum of a_{l-1} , $t_{\arg \beta(s_l)}^s$, and $t_{s_{l-1}s_l}^d$. If s_{l-1} is the customer node, it produces merged meal delivery with the last node, a_{l-1} is equal to a_{l-2} . The service time of the same customer node is same, so the calculation of a_l is not affected.

$$set_l = \{l' \mid \prod_{l'=1}^{l-1} p_{l'}^1 = 1\} \cup l \quad (52)$$

$$a_l = T_0 + t_{s_0 s_1}^d$$

$$a_l = \begin{cases} a_{l-1} & p_{l-1}^1 + p_{l-1}^2 = 1 \\ \max(a_{l-1}, \max_{w \in set_{l-1}} (r_{\arg \alpha(s_w)} + t_{\arg \alpha(s_w)}^p)) + t_{s_{l-1}s_l}^d & p_{l-1}^1 + p_{l-1}^2 = 0 \\ a_{l-1} + t_{\arg \beta(s_{l-1})}^s + t_{s_{l-1}s_l}^d & p_{l-1}^1 + p_{l-1}^2 = 0, s_{l-1} \in J' \end{cases} \quad (63)$$

$$g_n = a_{\arg s(\beta_n)} + t_n^s, \quad \forall n \in N \quad (74)$$

Constraint condition (15) is the quantity constraint of uncompleted orders.

$$\lambda_l = |N|$$

$$\lambda_{l+1} = \begin{cases} \lambda_l & s_l \in I' \\ \lambda_l - 1 & s_l \in J' \end{cases} \quad (85)$$

Constraint conditions (16) - (19) are open chain structure constraints. To maintain the flow balance between intermediate nodes, (16) - (17) traverses all restaurant nodes and customer nodes. (18) and (19) are open chain endpoint constraints, the two endpoints of the path are not connected. (20) is the constraint to eliminate subloop.

$$\sum_{i \in A' \cup s_0} x_{ij} = 1, \quad \forall j \in A' \quad (96)$$

$$\sum_{j \in A'} x_{ij} = 1, \quad \forall i \in (A' - s_{|A'|}) \cup s_0 \quad (107)$$

$$\sum_{i \in A' \cup s_0} x_{s_{|A'|}i} = 0 \quad (118)$$

$$\sum_{i \in A'} x_{is_0} = 0 \quad (129)$$

$$\sum_{i \in H} \sum_{j \in H} x_{ij} \leq |H| - 1, \quad \forall H \subset A', 2 \leq |H| \leq |A'| - 2 \quad (20)$$

The established MO-RPDRP is a stochastic programming model. Three objective functions in the model are derived as follows.

$$\begin{aligned}
\min F_1 &= E(\omega) \\
&= E\left(\frac{\sum_{n \in N} \max(0, \text{sgn}(g_n - b_n))}{|N|}\right) \\
&= \frac{1}{|N|} \sum_{n \in N} E(\max(0, \text{sgn}(g_n - b_n))) \\
&= \frac{1}{|N|} \sum_{n \in N} \int_0^{+\infty} \text{sgn}(g_n - b_n) f(g_n - b_n) d(g_n - b_n) \\
&= \frac{1}{|N|} \sum_{n \in N} \int_0^{+\infty} f(g_n - b_n) d(g_n - b_n)
\end{aligned} \tag{213}$$

$f(g_n - b_n)$ represents the probability density function of $g_n - b_n$

$$\begin{aligned}
\min F_2 &= E\left(\sum_{n \in N} (g_n - r_n)\right) \\
&= E\left(\sum_{n \in N} g_n\right) - \sum_{n \in N} r_n
\end{aligned} \tag{22}$$

$$\begin{aligned}
\min F_3 &= E\left(\sum_{l=1}^{|A|} \lambda_l (a_l - a_{l-1})\right) \\
&= \sum_{l=1}^{|A|} \lambda_l E(a_l - a_{l-1}) \\
&= \sum_{l=1}^{|A|} \lambda_l (E(a_l) - E(a_{l-1}))
\end{aligned} \tag{23}$$

Obviously, according to (21) - (23), the core of the derivation of three objective functions is the derivation of $E(a_l)$. So we derive the expression (24) of $E(a_l)$ according to the recursive form of (13).

$$a_l = \begin{cases} a_{l-1} & p_{l-1}^1 + p_{l-1}^2 = 1 \\ \max(a_{l-1}, \max_{w \in \text{set}_{s_{l-1}}} (r_{\arg \alpha(s_w)} + t_{\arg \alpha(s_w)}^p)) + t_{s_{l-1}s_l}^d & p_{l-1}^1 + p_{l-1}^2 = 0 \\ a_{l-1} + t_{\arg \beta(s_{l-1})}^s + t_{s_{l-1}s_l}^d & s_{l-1} \in I', p_{l-1}^1 + p_{l-1}^2 = 0 \end{cases} \tag{24}$$

Obviously, when $p_{l-1}^1 + p_{l-1}^2 = 1$, $E(a_l) = E(a_{l-1})$ when $p_{l-1}^1 + p_{l-1}^2 = 0$ and $s_{l-1} \in J'$, $E(a_l) = E(a_{l-1}) + t_{\arg \beta(s_{l-1})}^s + t_{s_{l-1}s_l}^d$.

When $p_{l-1}^1 + p_{l-1}^2 = 0$ and $s_{l-1} \in I'$, $\forall l > 1$, due to the different forms of a_{l-1} under l . When the AGV did not pass through the restaurant node before arriving at s_{l-1} , a_{l-1} is a definite constant. When the AGV passes through the restaurant node before arriving at s_{l-1} , a_{l-1} is a random variable.

1) If the AGV did not pass through the restaurant node before arriving at s_{l-1} , a_{l-1} is a constant. The expectation of a_l can be obtained by referring to the derivation of $E(a_2)$.

2) As can be seen from the above, if the AGV pass through the restaurant node before arriving at node s_{l-1} , a_{l-1} is a random variable. Obviously, the original function of the $E(a_l)$ integrand in the expression is a non-elementary function [29], it is difficult to solve this problem by directly finding the original function.

Therefore, we choose to use the discretization method to estimate the expectation of a_l [28]. In other words, the meal preparation time obeying the Gaussian distribution is transformed into a series of discrete time points. The expectation of a_l is estimated by using the rules and linear operations

in Tables 3–5. Obviously, the time when the AGV arrive at any node is a time set, and the time when they leave any node is also a time set.

Table 3. Comparison rule 1.

Input: (1) Time set of AGV arriving at restaurant node i : $A_{\arg s(i)}^{arrival}$
(2) The meal preparation time set of restaurant node i : $C_{\arg \alpha(i)}^{dining}$
Output: The set of times when the AGV leaves restaurant node i : $Z_{\arg s(i)}^{departure}$

$$Z_{\arg s(i)}^{departure} = []$$

For $a_{\arg s(i)}^{arrival} \in A_{\arg s(i)}^{arrival}$

For $c_{\arg \alpha(i)}^{dining} \in C_{\arg \alpha(i)}^{dining}$

If $a_{\arg s(i)}^{arrival} \in c_{\arg \alpha(i)}^{dining}$

$$Z_{\arg s(i)}^{departure} = Z_{\arg s(i)}^{departure} \cup a_{\arg s(i)}^{arrival} \in$$

else

$$Z_{\arg s(i)}^{departure} = Z_{\arg s(i)}^{departure} \cup c_{\arg \alpha(i)}^{dining}$$

End if

End for

End for

Table 4. Comparison rule 2.

Input: (1) Time set of AGV arriving at restaurant node i : $A_{\arg s(i)}^{arrival}$
(2) Total consolidated orders at restaurant node i : W
(3) The meal preparation time set of restaurant node i : $C_{\arg \alpha(i)w}^{dining}$
Output: The set of times when the AGV leaves restaurant node i : $Z_{\arg s(i)}^{departure}$

$$Z_{\arg s(i)}^{departure} = []$$

$$w=1$$

For $a_{\arg s(i)}^{arrival} \in A_{\arg s(i)}^{arrival}$

$$Z_{\arg s(i)}^{departure} = F(w, W, a_{\arg s(i)}^{arrival}, C_{\arg \alpha(i)w}^{dining}, Z_{\arg s(i)}^{departure})$$

End

$$F(W, a_{\arg s(i)}^{arrival}, C_{\arg \alpha(i)w}^{dining}, Z_{\arg s(i)}^{departure})$$

Input: (1) Current level: W
(2) Total order number: W
(3) Time set of AGV arriving at restaurant node i : $a_{\arg s(i)}^{arrival}$
(4) The meal preparation time set of restaurant node i : $C_{\arg \alpha(i)w}^{dining}$
(5) The set of times when the AGV leaves restaurant node i : $Z_{\arg s(i)}^{departure}$

Output: Result

```

value = aargs(i)arrival
If w < W
    For cargα(i)dining ∈ Cargα(i)wdining
        value2 = max(value, cargα(i)dining)
        Zarg s(i)departure = F(w + 1, W, value2, Cargα(i)(w+1)dining, Zarg s(i)departure)
    End
Else
    For cargα(i)dining ∈ Cargα(i)wdining
        value2 = max(value, cargα(i)dining)
        Zarg s(i)departure = Zarg s(i)departure ∪ value2
    End
End
Result = Zarg s(i)departure

```

Table 5. selection rule.

```

Input:(1)The set of times when the AGV leaves restaurant node i : Zarg s(i)departure
      (2) The upper limit of time node density : γ
Output: The set of times when the AGV leaves restaurant node after deletion Zarg s(i)departure'

```

```

Zarg s(i)departure' = []
num = []
q' = ⌊ (max(Zarg s(i)departure) - min(Zarg s(i)departure)) / γ ⌋
If ⌊ Zarg s(i)departure ⌋ > q'
    kdel = ⌊ Zarg s(i)departure ⌋ - q'
    krem = mod(⌊ Zarg s(i)departure ⌋ / kdel)
    m = ⌊ ⌊ Zarg s(i)departure ⌋ / kdel ⌋
    n = (Zarg s(i)departure - krem) / m
    If krem > 0
        Randomly generate a 0-1 q' vector with length m ,The sum of elements in
        vector is krem
        For i in vector
            If i = 1
                num = numi ∪ n + 1
            Else
                num = numi ∪ n
            End if
        End for
        Divide the [max(Zarg s(i)departure), min(Zarg s(i)departure)] into m intervals based on the number
        in num
        Randomly delete a time point within every interval
        Union of deleted intervals is Zarg s(i)departure'
    Else
        Divide the [max(Zarg s(i)departure), min(Zarg s(i)departure)] evenly into m intervals
        Randomly delete a time point within every interval
        Union of deleted intervals is Zarg s(i)departure'

```

```

End if
Else
 $Z_{\arg s(i)}^{\text{departure}} = Z_{\arg s(i)}^{\text{departure}}$ 
End if

```

By adopting the above discretization method, on the one hand, the difficulty of directly finding the original function is avoided in the process of finding the expectation of a_l ; On the other hand, since this problem is a dynamic problem based on the rolling frame, the discretization method can reduce the time complexity. Besides, the calculation efficiency can be improved by adjusting the parameters τ 、 γ .

Obviously, we need to modify the objective function. For objective function 1:

$$\begin{aligned}
 \min F_1 &= E(\omega) \\
 &= \frac{1}{|N|} \sum_{n \in N} E(\max(0, \text{sgn}(g_n - b_n))) \\
 &= \frac{1}{|N|} \sum_{n \in N} \sum_{a_{\arg s(\beta_n)}^{\text{arrive}} \in A_{\arg s(\beta_n)}^{\text{arrive}}} \frac{\max(0, \text{sgn}(a_{\arg s(\beta_n)}^{\text{arrive}} + t_n^s - b_n))}{|A_{\arg s(\beta_n)}^{\text{arrive}}|}
 \end{aligned} \tag{25}$$

For objective function 2:

$$\begin{aligned}
 \min F_2 &= E\left(\sum_{n \in N} (g_n - r_n)\right) \\
 &= \frac{1}{|N|} \sum_{n \in N} \sum_{a_{\arg s(\beta_n)}^{\text{arrive}} \in A_{\arg s(\beta_n)}^{\text{arrive}}} \frac{(a_{\arg s(\beta_n)}^{\text{arrive}} + t_n^s)}{|A_{\arg s(\beta_n)}^{\text{arrive}}|} - E\left(\sum_{n \in N} r_n\right) \\
 &= \frac{1}{|N|} \sum_{n \in N} \sum_{a_{\arg s(\beta_n)}^{\text{arrive}} \in A_{\arg s(\beta_n)}^{\text{arrive}}} \frac{(a_{\arg s(\beta_n)}^{\text{arrive}} + t_n^s)}{|A_{\arg s(\beta_n)}^{\text{arrive}}|} - \sum_{n \in N} r_n
 \end{aligned} \tag{26}$$

For objective function 3:

$$\begin{aligned}
 \min F_3 &= E\left(\sum_{l=1}^{|A|} \lambda_l (a_l - a_{l-1})\right) \\
 &= \sum_{l=1}^{|A|} \lambda_l (E(a_l) - E(a_{l-1}))
 \end{aligned} \tag{27}$$

$$E(a_l) = \frac{1}{|A_l^{\text{arrive}}|} \sum_{a_l^{\text{arrive}} \in A_l^{\text{arrive}}} a_l^{\text{arrive}}$$

4. The Proposed AC-NSGA-III Algorithm

4.1. NSGA-III

The constructed MO-RPDRP model is essentially a Traveling Salesman Problem (TSP) with complex constraints and AC strategy. Obviously, it falls into the category of NP-hard problems, which does not have an exact polynomial-time algorithm. Furthermore, the 3 objective functions considered in the model trade off each other, so this problem is a multi-objective optimization problem. To solve the MO-RPDRP model, we propose the AC-NSGA-III algorithm, which based the NSGA-III[30], one of the multi-objective intelligent optimization algorithms. We have a different approach to algorithm encoding. The flowchart of the algorithm is shown in Figure 6.

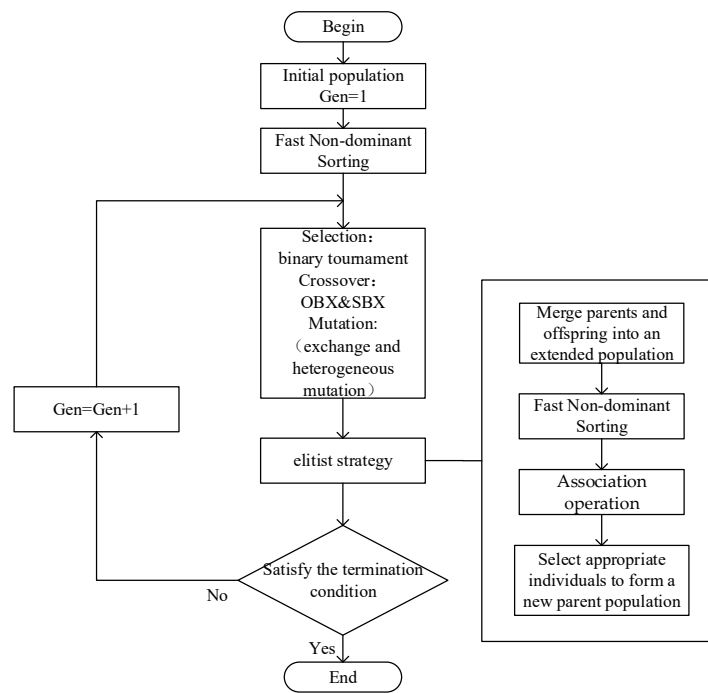


Figure 6. algorithm flow chart.

4.2. Operator Design

4.2.1. IRC Coding

Chromosome encoding is a pivotal component of the AC-NSGA-III algorithm. To ensure that the generated chromosome encodings correspond to feasible solutions and to prevent infeasible solutions resulting from crossover operations that violate model constraints (such as delivery priority, vehicle capacity limits, etc.), a decoding rule based on a cellular array structure was designed.

The cellular array Q represents a chromosome, $Q=[x,y]$. The design of the chromosome structure leverages the concept of constructive heuristic methods, constructing the solution on an order-by-order basis. The variable x represents the construction sequence of the generated orders.

The variable y is composed of two components, $y = \begin{bmatrix} y^1 \\ y^2 \end{bmatrix}$, where the values indicate the insertion positions of the corresponding nodes. Specifically, y^1 denotes the periodic insertion positions of the restaurant nodes associated with x , and y^2 denotes the periodic insertion positions of the customer nodes associated with x . Both x and y^2 are one-dimensional arrays of length $|N|$, while y^1 is a one-dimensional array of length $|N2|$.

The values in y^1 and y^2 exhibit periodic reducibility, meaning they are handled cyclically. When these values exceed the total number of available insertion positions, the insertion position for the corresponding order node is determined by the remainder of this value divided by the total number of insertion positions. If the remainder is zero, the insertion position is the last available slot.

The decoding process for the chromosome involves inserting the restaurant and customer nodes corresponding to the order into the delivery route according to the values specified in y^1 and y^2 . This ensures that the resulting solutions adhere to the model constraints while facilitating effective order construction and scheduling.

```

Input: Q; Q= [x, y] %chromosome
      x;% the construction sequence of the generated orders
      y=[y1,y2];%arrays representing node insertion locations, including periodic insertion
      locations for pickup and customer nodes
      N;%The number of orders involved
Output: RESULT;%node access order

```

```

L=[];%initially empty set, storing the temporary path (to be inserted)
pick();%function of getting the restaurant node
delivery();%function of getting the customer node
P();%function of getting the list of insertable locations
insertNode();%node insertion function
i=1;
L=[pick(x(i)),delivery(x(i))];
for i = 2:N
    MNode = pick(x(i));% gets the restaurant node
    n1=P(MNode,L)%gets the list of insertable locations
    % calculate the insertion location of the restaurant node
    if mod(y1(i), length(n1))=0
        insertIndex=n1(end);
    else
        insertIndex = n1((mod(y1(i), length(n1))));
    L = insertNode(L, MNode, insertIndex); % inserts the node at the insertion position
    CNode = delivery(i);% gets the customer node
    n2=P(CNode,L)%gets the list of insertable locations
    % calculate the customer node insertion location
    if mod(y2(i), length(n2))=0
        insertIndex=n2(end);
    else
        insertIndex = n2((mod(y1(i), length(n2))));
    L = insertNode(L, CNode, insertIndex); % insert a node at the insertion position
end
RESULT=L

```

1) In this process, AC strategy must be followed, and node insertion must not violate constraint (9), meaning that if the actual positions represented by adjacent order nodes in L are the same, they should be combined for pickup or delivery.

2) Node insertion must not violate constraints (6) - (8), meaning it should not breach the delivery volume constraints or the order node and visit sequence restrictions.

3) Since the decoded results are real number sequences, the open chain structure constraint and sub-loop elimination constraints (16) - (20) can be naturally avoided. The positions filtered by this restriction rule are the actual insertable positions.

Figure 7 shows an example of IRC, which includes three orders involving restaurant nodes 1, 2, 3 (denoted as M1, M2, M3) and customer nodes 1, 2, 3 (denoted as C1, C2, C3). $x = (2, 1, 3)$, $y = \begin{bmatrix} 5, 7, 5 \\ 3, 5, 8 \end{bmatrix}$. In the encoding, the process starts with the restaurant node 2 (M2) and customer node 2 (C2) associated with the first order, forming the path M2-C2. Then, according to the value 5 in y^1 and the total number of available insertion positions is 2, $5 \% 2 = 1$, so the restaurant node M1 is inserted into the first available position. The visit sequence after insertion is M2-M1-C2.

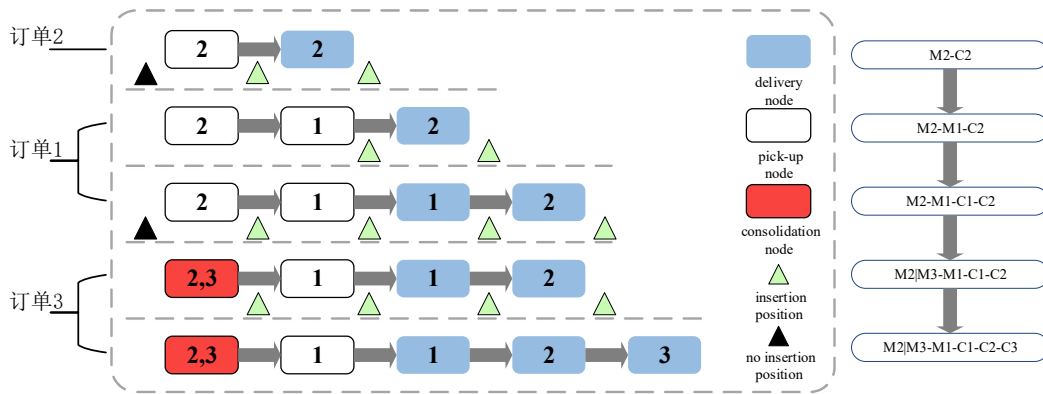


Figure 7. Decoding structure and insertion restriction rules.

Next, according to the value 5 in y^2 , the customer node C1 associated with order 1 is inserted. Based on constraint 8, the restaurant node 1 must be visited before the customer node 1, so the total number of available insertion positions is 2, $5 \bmod 2 = 1$, so the customer node C1 is inserted into the first available position. The visit sequence after insertion is M2-M1-C1-C2.

Finally, according to the value 5 in y^1 , the restaurant node M3 associated with order 3 is inserted. Since M3 and M2 correspond to the same actual pickup location, they are merged for pickup. Similarly, C3 is inserted following the same principle, and thus the complete delivery and pickup route for all orders is formed.

4.2.2. Initial Population

The population size is N , the initial population P_0 is generated randomly.

x is a randomly generated sequence. Every random positive integer in y^1 , y^2 is not greater than $|A^i| - 1$.

4.2.3. Non-Dominant Hierarchical Ranking of Population

Non-dominated sorting effectively distinguishes individuals' relative advantages in multi-objective optimization, aiding in the selection and retention of high-quality individuals, thereby maintaining population diversity and evolutionary effectiveness. Multi-objective optimization algorithms generate selection pressure through population sorting, encouraging the population to continuously approach the Pareto front.

4.2.4. Reference Point Evaluation System

In order to evaluate the population diversity, the reference point selection method of NSGA-III is adopted [31].

4.2.5. Binary Tournament Selection

Two individuals from the population are randomly selected to compete: If the non-dominant ranks of them are different, low-ranking individual is selected to be put into the mating pool. Otherwise, the individual whose associated reference points with smaller number of niches is select to be put into the mating pool. The selection is repeated N times.

4.2.6. OBX&SBX Mixed Crossover

For x , order-based crossover (OBX) is used, ensuring that no elements are missing after crossover, with a crossover probability of P_c . It is showed in Figure 8. In order to better deal with the sparsity of individual space in the process of dealing with multi-objective optimization problems, simulated binary crossover (SBX) [32] is adopted for y^1 , y^2 . The non-integer solution generated by SBX was rounded in the way of rounding off.

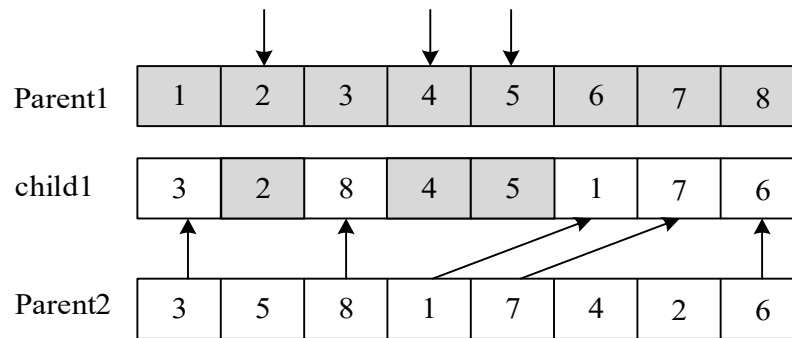


Figure 8. Order-Based Crossover, OBX.

4.2.7. Mutation Operator

Exchange mutation is adopted for x . When an exchange mutation is applied to chromosomes based on binary or integer, two genes are randomly selected. Then, their values are exchanged. When the random number is less than the given mutation rate P_m , chromosome mutation occurs [33]. Non-uniform variation was adopted for y^1 , y^2 . The original gene value was randomly disturbed. The result was taken as the new gene value. After every locus is calculated with the same probability, it is equivalent to the entire solution vector making a slight change in the solution space. In the operation of non-uniform mutation from $X = x_1 x_2 x_3 \dots x_k \dots x_l$ to $X' = x_1 x_2 x_3 \dots x'_k \dots x_l$, if the value range of mutation point x_k is $[U_{\min}^k, U_{\max}^k]$, the new gene value x'_k will be determined by (28), where t is the current iteration number.

$$x'_k = \begin{cases} x_k + \Delta(t, U_{\max}^k - x_k), & \text{if } \text{random}(0,1) = 0 \\ x_k - \Delta(t, x_k - U_{\min}^k), & \text{if } \text{random}(0,1) = 1 \end{cases} \quad (28)$$

4.2.8. Elite Strategy

Parent population P_t and offspring population Q_t are merged into a population R_t

Fast non-dominant hierarchical sorting: individuals in R_t are divided into ranks $(F_{P_1}, F_{P_2}, \dots)$. A new population of S_t is constructed from F_{P_1} . When F_{P_L} is put in it, $S_t = N$ or $S_t > N$.

Association operation: the line between the origin and the reference point is taken as the reference line. The individuals in population $P_{t+1}(P_{t+1} = S_t / F_{P_L})$ are associated with the reference points on the nearest reference line. The number of individuals associated with each reference point (i.e., the number of niches) is calculated, it is denoted as ρ_j (the niches number of the j th reference point is ρ_j).

Individual retention operation: elite individuals from F_{P_L} are selected. Then, they enter P_{t+1} . In order to maintain the diversity of the population, the reference point of ρ_j with the minimum is denoted as j (if ρ_j of multiple reference points is equal and minimum, one of them is denoted as $\bar{j}$ at random).

4.2.9. Termination Condition of the Algorithm

The convergence algebra threshold is set as θ , the inverse generation distance is introduced to identify the convergence algebra of the algorithm [34]. IGD is the average value of the minimum distance between the point (feasible solution) on the front surface of each theoretical Pareto and the non-dominated solution set obtained by the algorithm. Similarly, the distance between the non-dominated solution set of two successive generations of populations can be expressed. Since the elite strategy of AC-NSGA-III makes the non-dominated solution set of the progeny population not inferior to that of the parent population, the IGD value is calculated based on the non-dominated solution set of the progeny population. β is defined as threshold value. In successive generations θ , when the IGD between the parent generation and the child generation is smaller than β , the algorithm converges. The calculation formula of IGD is shown in (29). Where Q_t is the child non-dominant solution set, P_t is the parent non-dominant solution set, and $d(v, Q_t)$ is the minimum distance from v to P_t of the individual in Q_t .

$$IGD(Q_t, P_t) = \frac{1}{|Q_t|} \sum_{v \in Q_t} d(v, P_t)$$

(29)

5. Experiments

5.1. Basic Data

In order to explore the effectiveness of the proposed model and algorithm. According to the one-month actual investigation of a delivery enterprise, it is calculated that the number of orders received by AGVs is the largest during the peak hour (11:00-13:00) in one day. Therefore, the information of the top four orders of peak hour in the month is selected as examples, and the details of the examples are shown in Appendix A. The design of fixed parameters in the model is shown in Table 6. The hardware environment of the numerical experiment is Intel i9-12900K/KS CPU, DDA4-3600MH-32G(16G*2) memory, 970-EVO-Plus-NVMe-M.2-1T solid state disk, windows 10 operating system. The proposed algorithms are implemented by Python language.

Table 6. Values of some parameters in the numerical experiment.

parameters	value
v	68.45 dm^3
T_0	11: 00
t_n	5min

N	10
P_c	0.8
P_m	0.06

4.2. Ex5.2. Experimental Design

- 1) Comparison of optimization effects considering uncertainty factors.
Under the same algorithm conditions, to verify the superiority of the proposed model, the uncertainty model considering the meal preparation time and the model without considering the meal preparation time are compared.
- 2) Comparison of order combination modes.
Under the same algorithm conditions, to verify the superiority of the proposed order combination method, the AC strategy is compared with other order combination strategies (OCS [25] and NSDP [35]).
- 3) Comparison of look-forward optimization effects.
Under the same algorithm conditions, to verify the cumulative optimization effect of look-forward in the rolling optimization process, the model considering objective function 3 is compared with the model without considering objective function 3.

5.3. Comparison of Results

Comparison of optimization effect considering uncertainty factors.

To evaluate the effectiveness of the proposed model, numerical experiments were conducted on the deterministic model (model-C) and the Uncertainty model (Model-Unc) under each example. The two models were run 100 times respectively. The data characteristics of total customer waiting time and order timeout rate were obtained as shown in the figure below.

Figure 9 is box chart, showing the median, lower quartile, upper quartile, minimum and maximum values. In the figure, the dashed lines represent the expected values of the solutions of model-C and Model-Unc under each example. As can be seen from the figure, on the one hand, the expected value of the solution under Model-Unc is better. On the other hand, the model-C solution has a larger fluctuation range than Model-Unc solution, the unfeasible rate of model-C solution is higher. The uncertainty model proposed by us has a good immune effect to uncertainty factors. In other words, the solution of Model-Unc has good robustness under uncertain environment.

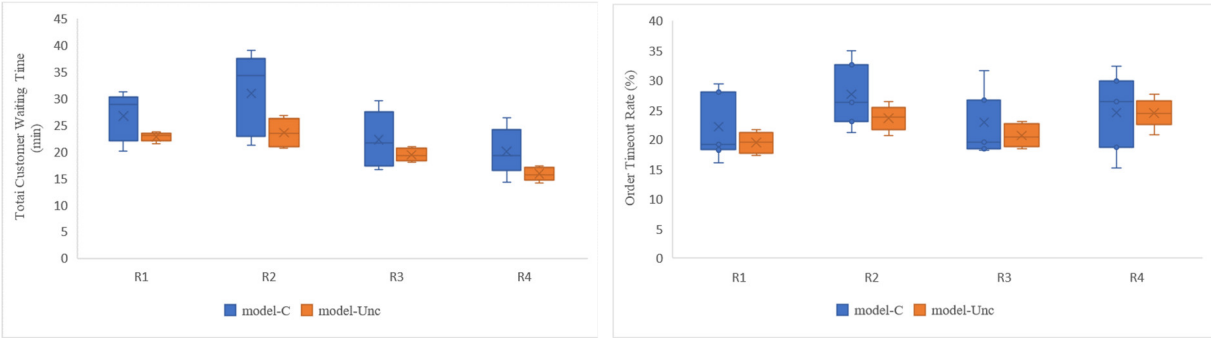


Figure 9. Experimental result--a.Total customer waiting time(left) b.Order timeout rate(right).

To better reflect the optimization effect of the proposed model under uncertain environment, A1 is taken as an example. The mean value of the meal preparation time remained unchanged. As the standard deviation of the meal preparation time increased, the standard deviation trend of the solutions of model-C and Model-Unc was calculated. The experimental results are shown in the figure below.

In Figure 10, as the standard deviation of meal preparation time increases, the standard deviation of total customer waiting time under Model-Unc is much smaller than that under model-C. When the standard deviation of meal preparation time of restaurants is 1.2, the scheduling scheme

solved by Model-Unc reduces the standard deviation of total customer waiting time by 38.1% compared with model-C. The experimental results of the order timeout rate are similar, it will not be described here.

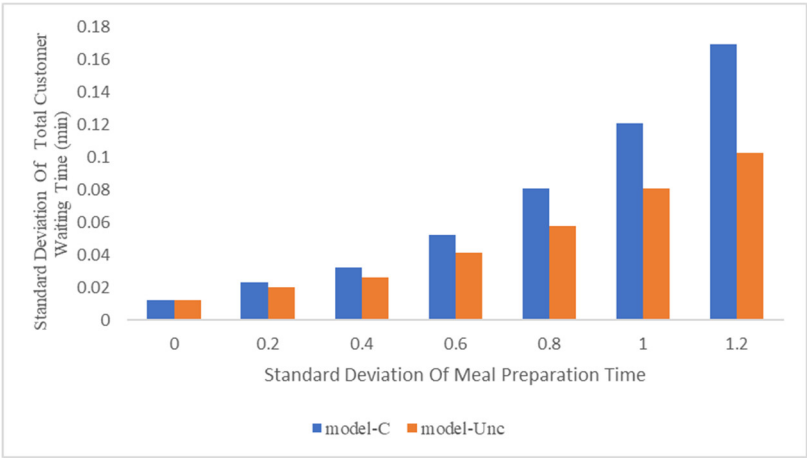


Figure 10. Experimental result (Total customer waiting time).

Comparison of order combination patterns.

The solution of the MO-RPDRP model is a three-dimensional pareto front. In order to enable the pareto front to be located on a two-dimensional plane, the look-forward objective dimension is removed in the following experiments. In this experiment, the model considers the look-forward objective.

In this experiment, the rolling scheduling with a large number of remaining unfinished orders is taken as an example. It makes the comparative effect of Pareto frontier more obvious under different order combination modes. Figure 11 shows the change of the number of remaining unfinished orders under each rolling schedule of model-1 (four examples). In each group of examples, the 7th and 8th rolling scheduling have the largest number of remaining orders, so they are selected as examples. Figure12 takes A1 as an example, it shows Pareto frontiers of three order modes under model-1. It can be seen from the figure that the Pareto frontier of the model in AC strategy is significantly better than the other two.

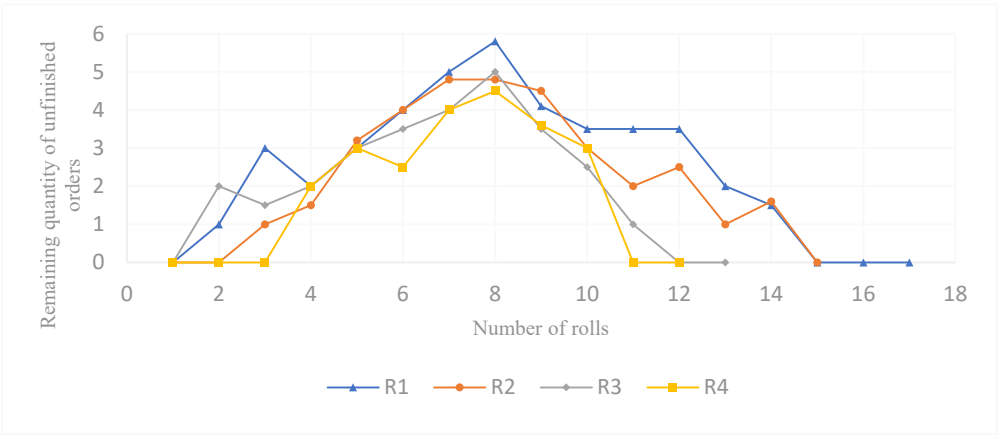


Figure 11. Change in the number of remaining unfinished orders.

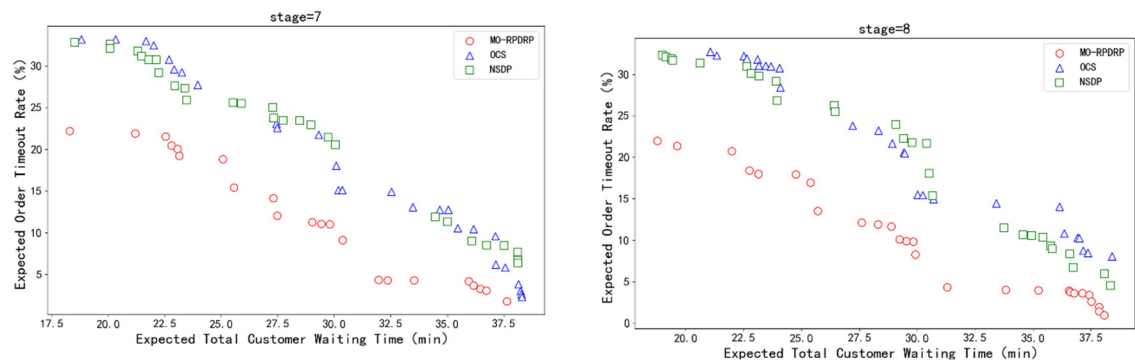


Figure 12. Superiority comparison of order combination mode.

Comparison of look-forward optimization effects.

The model considering the look-forward objective is denoted as Model-1. A model that does not consider the look-forward objective is denoted as Model-2.

Taking A1 as an example, to compare the solutions of mode-1 and mode-2, the orders of the 4th, 7th, 11th and 15th rolling scheduling are selected as experimental examples. The solution extracted from the pareto frontier obtained by mode-1 is represented as solu-1. The solution extracted from the pareto frontier obtained by mode-2 is represented as solu-2. The experimental results are shown in the figure below.

In Figure 13, on the fourth rolling scheduling, although the solutions extracted from Pareto front of model-2 dominate the solutions extracted from Pareto front of model-1. With the increasing of rolling schedule times, the scheduling after-effect of look-forward objective increases gradually. Pareto frontier obtained by model-1 considering look-forward objective is significantly better than Pareto frontier obtained by model-2 without look-forward objective. The solutions extracted from the Pareto front of model-2 no longer dominate the solutions extracted from the Pareto front of model-1.

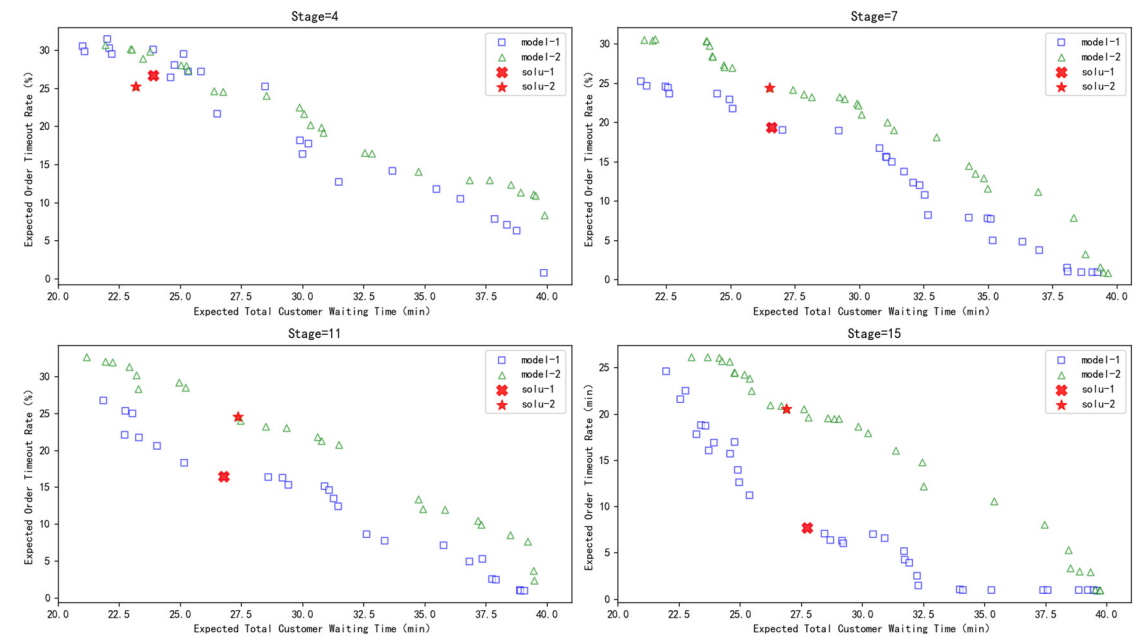


Figure 13. Superiority comparison of look-forward.

6. Conclusions

In this paper, we considered the uncertainty of the meal preparation time, the AGV route optimization problem was addressed. Based on the realistic constraints, such as order node access sequence restriction and vehicle volume, a MO-RPDRP model integrating AC strategy and node sequence is developed. In the model, customer waiting time and order timeout rate are taken into account. Meanwhile, an improved NSGA-III algorithm is designed. In the numerical experiment, the actual survey data of a regional delivery enterprise during the peak hours are taken as examples. The results show that: (a) Compared with the deterministic model, the solution obtained by MO-RPDRP model has good robustness and better optimization effect under uncertain environment; (b) The Pareto frontier of the model is better under AC strategy.

As far as we know, based on the uncertain environment, the literature considering order combination for rolling optimization of take-out, pick-up and delivery path is still in initial stage. Further research in related areas can be carried out in the future. For example, to achieve higher delivery efficiency, the collaborative optimization problem of multiple AGV is studied.

Appendix A

Table A1. Example 1 (A1) .

The number of rolling schedules	Order time	Restaurant preparation time is subject to distribution expectations	Restaurant preparation time follows the standard deviation of the distribution	Restaurant node number	Customer node number	The latest delivery time
(1)	11: 00: 22	2	0.5	1	14	11: 30
(2)	11: 19: 58	3	0.2	2	15	11: 45
(3)	11: 23: 19	2.4	1	3	16	11: 53
(4)	11: 25: 24	2.8	0.8	3	17	11: 55
(5)	11: 31: 25	2	0.6	4	18	12: 00
(6)	11: 47: 14	3.4	2	5	19	12: 00
(7)	11: 51: 54	5	1.6	6	20	12: 20
(8)	11: 56: 42	5	1.5	6	21	12: 25
(9)	11: 59: 32	5	0.8	6	22	12: 30
(10)	12: 05: 49	2	1	7	23	12: 35
(11)	12: 11: 36	4	0.4	8	24	12: 40
(12)	12: 20: 20	2	0.6	9	24	12: 55
(13)	12: 27: 14	5.4	1.7	10	25	12: 55
(14)	12: 30: 28	3	1.3	11	26	13: 10
(15)	12: 31: 54	6.2	1	12	27	13: 15
(16)	12: 39: 42	6.2	1.2	12	28	13: 20
(17)	12: 58: 49	2.3	0.9	13	29	13: 30

Table A2. Example 2 (A2) .

The number of rolling schedules	Order time	Restaurant preparation time is subject to distribution expectations	Restaurant preparation time follows the standard deviation of the distribution	Restaurant node number	Customer node number	The latest delivery time
(1)	11: 04: 08	2	0.9	1	13	11: 35

(2)	11: 20: 08	3	0.2	2	14	11: 50
(3)	11: 25: 58	5	1.7	3	15	12: 00
(4)	11: 28: 24	3.8	0.8	4	16	12: 00
(5)	11: 35: 19	2	0.9	5	17	12: 10
(6)	11: 46: 47	3	2	5	18	12: 15
(7)	11: 50: 07	5	0.4	6	19	12: 25
(8)	11: 58: 49	5.2	1.5	7	20	12: 30
(9)	12: 20: 32	2.4	1	8	21	12: 55
(10)	12: 31: 25	3	1.2	9	22	13: 00
(11)	12: 39: 36	2	0.9	9	23	13: 20
(12)	12: 40: 20	4	1.7	9	24	13: 25
(13)	12: 47: 14	5.5	1	10	25	13: 30
(14)	12: 51: 28	2	1	11	26	13: 30
(15)	12: 51: 54	6	1	12	26	13: 35

Table A3. Example 3 (A3) .

The number of rolling schedules	Order time	Restaurant preparation time is subject to distribution expectations	Restaurant preparation time follows the standard deviation of the distribution	Restaurant node number	Customer node number	The latest delivery time
(1)	11: 10: 42	6.2	1.2	1	13	11: 45
(2)	11: 20: 49	2.3	0.9	2	13	11: 50
(3)	11: 25: 08	2	1.7	3	14	11: 55
(4)	11: 28: 13	4.4	1.6	4	15	12: 00
(5)	11: 31: 25	4.6	0.4	5	16	12: 00
(6)	11: 40: 59	2	0.9	6	17	12: 15
(7)	11: 55: 24	3	1	7	18	12: 30
(8)	12: 10: 19	2.1	0.4	8	19	12: 40
(9)	12: 26: 17	3.7	0.2	8	20	12: 55
(10)	12: 29: 24	3	2	9	21	13: 00
(11)	12: 32: 06	4	1	10	21	13: 00
(12)	12: 45: 04	5.5	2.4	11	22	13: 25
(13)	12: 50: 43	5.5	1.9	12	23	13: 30

Table A4. Example 4 (A4) .

The number of rolling schedules	Order time	Restaurant preparation time is subject to distribution expectations	Restaurant preparation time follows the standard deviation of the distribution	Restaurant node number	Customer node number	The latest delivery time
(1)	11: 00: 36	5	1.5	1	12	11: 30
(2)	11: 05: 08	4.6	0.4	2	13	11: 35
(3)	11: 17: 47	2.4	1	3	14	11: 50
(4)	11: 29: 24	4	1	4	15	11: 55

(5)	11: 39: 20	2	0.6	4	16	12: 00
(6)	11: 48: 47	3.4	2	5	17	12: 15
(7)	11: 53: 07	5.5	2.4	6	18	12: 25
(8)	12: 19: 24	5	1.5	7	19	12: 40
(9)	12: 27: 32	4.6	0.4	8	19	12: 55
(10)	12: 31: 07	2	1	9	20	13: 00
(11)	12: 44: 39	5	2.4	10	21	13: 20
(12)	12: 54: 20	2	0.8	11	22	13: 30

References

1. <https://bg.qianzhan.com/report/detail/300/211112-19960858.html>

2. Giuffrida N, Fajardo-Calderin J, Masegosa A D, et al. Optimization and machine learning applied to last-mile logistics: A review[J]. Sustainability, 2022, 14(9): 5329.

3. Liu, L., Qu, T., Thurer, M., Ma, L., Zhang, Z., & Yuan, M. (2022). A new knowledge-guided multi-objective optimisation for the multi-AGV dispatching problem in dynamic production environments. International Journal of Production Research.

4. <https://tech.meituan.com/2020/02/20/meituan-delivery-operations-research.html> (美团)

5. Yan L , Guo B , Chao C , et al. FooDNet: Toward an Optimized Food Delivery Network based on Spatial Crowdsourcing[J]. IEEE Transactions on Mobile Computing, 2019, PP:1-1.

6. Xue G , Wang Z , Wang G . Optimization of Rider Scheduling for a Food Delivery Service in O2O Business[J]. Journal of Advanced Transportation, 2021, 2021(3):1-15.

7. Marlin W Ulmer,Barrett W Thomas,Ann Melissa Campbell,Nicholas Woyak. The Restaurant Meal Delivery Problem: Dynamic Pickup and Delivery with Deadlines and Random Ready Times[J]. Transportation Science,2021,55(1):

8. Xin, J.; Wei, L.; Wang, D.; Xuan, H. Receding horizon path planning of automated guided vehicles using a time-space network model. Optim. Control. Appl. Methods 2020, 41, 1889–1903.

9. Cheng W, Meng W. An efficient genetic algorithm for multi AGV scheduling problem about intelligent warehouse[J]. Robotic Intelligence and Automation, 2023, 43(4): 382-393.

10. Gao Y, Chen C H, Chang D. A Machine Learning-Based Approach for Multi-AGV Dispatching at Automated Container Terminals[J]. Journal of Marine Science and Engineering, 2023, 11(7): 1407.

11. Qiu L, Wang J, Chen W, et al. Heterogeneous AGV routing problem considering energy consumption[C].IEEE Int Conf on Robotics and Biomimetics, Zhuhai, China, 2015:1894-1899.

12. Chen J, Wu Y, Huang S, et al. Multi-objective optimization for AGV energy efficient scheduling problem with customer satisfaction[J]. AIMS Mathematics, 2023, 8(9): 20097-20124.

13. Xin, J.; Wei, L.; D’Ariano, A.; Zhang, F.; Negenborn, R. Flexible time-space network formulation and hybrid metaheuristic for conflict-free and energy-efficient path planning of automated guided vehicles. J. Clean. Prod. 2023, 398, 136472.

14. W. Zou, Q. Pan, T. Meng, L. Gao, Y. Wang, An effective discrete artificial bee colony algorithm for multi-AGVs dispatching problem in a matrix manufacturing workshop, Expert Syst. Appl., 161 (2020), 113675.

15. G. Li, B. Zeng, W. Liao, X. Li, L. Gao, A new AGV scheduling algorithm based on harmony search for material transfer in a real-world manufacturing system, Adv. Mech. Eng., 10 (2018),1–13.

16. Wang Y J, Liu X Q, Leng J Y, et al. Study on scheduling and path planning problems of multi-AGVs based on a heuristic algorithm in intelligent manufacturing workshop[J]. Advances in Production Engineering & Management, 2022, 17(4): 505-513.

17. Liu L, Qu T, Thüerer M, et al. A new knowledge-guided multi-objective optimisation for the multi-AGV dispatching problem in dynamic production environments[J]. International Journal of Production Research, 2023, 61(17): 6030-6051.

18. Reyes D , Erera A , Savelsbergh M , et al. The Meal Delivery Routing Problem. 2018.

19. Baris Yildiz,Martin Savelsbergh. Provably High-Quality Solutions for the Meal Delivery Routing Problem[J]. Transportation Science,2019,53(5):

20. Du J , Guo B , Liu Y , et al. CrowdNet: Enabling a Crowdsourced Object Delivery Network Based on Modern Portfolio Theory[J]. Internet of Things Journal, IEEE, 2019, 6(5):9030-9041.

21. Zachary Steever, Mark Karwan, Chase Murray. Dynamic courier routing for a food delivery service[J]. Computers and Operations Research, 2019, 107{5}:
22. Guofeng Sun, Zhiqiang Tian, Renhua Liu, Yun Jing, Yawen Ma, Purushothaman Damodaran. Research on Coordination and Optimization of Order Allocation and Delivery Route Planning in Take-Out System[J]. Mathematical Problems in Engineering, 2020, 2020{5}:
23. Tu W , Zhao T , Zhou B , et al. OCD: Online Crowdsourced Delivery for On-Demand Food[J]. IEEE Internet of Things Journal, 2019, PP(99).
24. Zheng J, Wang S, Wang L, et al. A two-stage algorithm for fuzzy online order dispatching problem[C]//2020 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2020: 1-8.
25. Liao W , Zhang L , Wei Z . Multi-objective green meal delivery routing problem based on a two-stage solution strategy[J]. Journal of Cleaner Production, 2020, 258:120627-.
26. Yu H, Luo X, Wu T. Online pickup and delivery problem with constrained capacity to minimize latency[J]. Journal of Combinatorial Optimization, 2020: 1-20.
27. Zheng, Jie, et al. "Modeling stochastic service time for complex on-demand food delivery." Complex & Intelligent Systems (2022): 1-15.
28. Jiang S, Li X, Zheng Q, et al. Approximate equal frequency discretization method[C]//2009 WRI global congress on intelligent systems. IEEE, 2009, 3: 514-518.
29. Department of Mathematics, Tongji University Advanced Mathematics [M] Beijing: Higher Education Press, 2014
30. Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints[J]. IEEE transactions on evolutionary computation, 2013, 18(4): 577-601.
31. Das, Indraneel; Dennis, J. E. (1998). Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. SIAM Journal on Optimization, 8(3), 631–657.
32. Sato M, Oyama A. Comparative Study of Crossovers for Decision Space Diversity of Non-Dominated Solutions[C]//2021 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2021: 01-08.
33. Park H, Son D, Koo B, et al. Waiting strategy for the vehicle routing problem with simultaneous pickup and delivery using genetic algorithm[J]. Expert Systems with Applications, 2021, 165: 113959.
34. SUN Y N , YEN G G , YI Z , et al, IGD Indicator-Based Evolutionary Algorithm for Many-Objective Optimization Problems. IEEE Transactions on Evolutionary Computation[J], 2019, 23 (2) : 173-187.
35. Steever Z, Karwan M, Murray C. Dynamic courier routing for a food delivery service[J]. Computers & Operations Research, 2019, 107: 173-188.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.