Research on Open Vehicle Routing Problem based on Multi-Objective and Multi-Constrain Genetic Algorithm

WANG Shi-yue
Donghua University, Shanghai, 201620, China
wsy19946291351@163.com

Abstract—With the advancement of urbanization, the efficiency of take-out delivery has become one of the focuses of urban residents. In order to comprehensively meet the needs of both the delivery system managers and the delivery customers, this paper designs a multi-objective and constraint linear programming model that takes into account the minimum total cost of delivery and is based on the timeliness of delivery to maximize customer satisfaction. At the same time, considering the high concurrency of customers' demands for delivery, this paper designs a set of multi-objective multi-constraint genetic algorithm based on the control evolution direction, which is carried out in feasible domain first, and then to the optimal solution, and elaborates the constraint control evolution operator and sorting operator. Finally, an example is given to verify the feasibility and effectiveness of the algorithm, and a new efficient solution method is proposed for the delivery route planning problem.

1. INTRODUCTION
With the development of science and technology, people have mastered more advanced data analysis and information management technology, and Vehicle Routing Problem (VRP) has been widely applied in many aspects of life, such as logistics distribution, take-out delivery, vehicle scheduling, garbage collection, site selection of building facilities. Up to now, the distribution scheme of VRP development still cannot satisfy the demand of customers and distribution company, and the constraints can not be well response in the operation of the problems in the real world, this article makes a take-out distribution situation of the campus surrounding abstract from the constrain of genetic algorithm based on multi-objective open vehicle routing planning research.

In this paper, the significance and contribution of the open vehicle routing problem based on multi-objective and multi-constraint genetic algorithm are as follows:

1. The influence of delivery waiting time on customer satisfaction is depicted through soft time windows to lower the penalty cost of platform and merchants and optimize the objective function;
2. Abstract the take-out delivery problem in hot periods around the campus into an open vehicle path planning problem. The existing model is used to increase the constraints in line with reality to obtain a near-optimal take-out delivery solution, reduce the risks and costs brought by take-out delivery in hot periods, and increase the total revenue of the platform, merchants and customers;
3. By adding constraints to the non-dominant sorting genetic algorithm with elite strategy, the heuristic algorithm can solve the open vehicle path problem with soft time windows quickly and effectively.
2. BACKGROUND

2.1 Motivation

Since it was proposed, the vehicle routing problem (VRP) has promoted the rapid development of related industries and derived a lot of new constraints to be considered and solved. Derive a new type of vehicle routing problem how for vehicle routing problem, multi-center vehicle routing problem, vehicle routing problem with time windows, vehicle routing problem with time limits, at the same time take delivery vehicle routing problem, the dynamic vehicle routing problem, green vehicle routing problem, heterogeneous problems such as the open vehicle routing problem. The emergence of these problems conforms to the development trend of the current society. Under the premise of meeting the distribution demand, green, sustainable development, resource saving, vehicle and energy efficiency improvement should be adopted to solve the problem of vehicle path. In the environment of big data, constraint simulation based on data will be more realistic so that the model can better solve practical problems and serve reality. Therefore, data can be used as a pheromone to better solve local planning problems without falling into local optimality, and data-driven can realize the minimization of the fitting difference between theory and reality.

2.2 Related Work

In 2007, Xiangyong Li[1] proposed a planning model to solve the open vehicle routing problem (OVRP) by introducing a fixed number of vehicles, heterogeneous vehicles, time windows and random travel time. Younghwan Park et al. [2] proposed in 2014 to focus on environmental issues and solve the green vehicle routing problem (GVRP) based on an eco-friendly strategy and algorithm. Wenjuan Gu[3] et al. improved the traditional planning model for solving the capacitated vehicle routing problem (CVRP) in 2019 aiming at the problem that multiple commodities of the same customer need to be visited multiple times, and proposed the model that merchants can be visited multiple times but given commodities need to be delivered in one delivery to solve constrained split delivery vehicle routing problem (C-SDVRP). In 2020, Roberto Baldacci et al. [4] proposed a new dual and proto-bound method to solve VRP of Fractional Objective Function.

Jonathan F. Bard et al. [5] proposed a model for solving the vehicle routing problem with satellite facilities (VRPSF) in 1997. By introducing satellite equipment, vehicle supplies could not necessarily return to the central warehouse, which solved the problem limited by vehicle capacity and path travel time. Ismail Yusuf et al. [6] proposed a genetic algorithm to solve rich-VRP (VRP with complex constraint) in 2014. They proposed a new method called "rank and select" and used two types of crossover and mutation. In 2017, Mazin Abed Mohammed et al. [7] used the improved genetic algorithm to solve CVP, and built the model to achieve the optimal solution of joint CVRP with accompanying target by reducing the time consumption and distance of all paths. Ahmed Kheiri et al. [8] proposed two meta-heuristic algorithms to solve VRP in 2019. The first algorithm introduces transformation probability to determine whether to improve the target value update probability and increase the chance of generating improved solutions; the second algorithm introduces a deterministic routing algorithm to build feasible routes and continuously improves them.

The above rich-VRP model and algorithm research mainly solved: 1. Vehicle distribution no need to return to the central warehouse to pick up the goods, save the cost of distribution path; 2. Introduction of elite strategy; 3. Introduce time window constraint. However, there are still the following disadvantages:

1. The objective function of the programming model is relatively simple, and multi-objective programming is also realized by linear weighting or hierarchical sequence, which does not meet the realistic situation;
2. The time complexity of the solving algorithm is too high, which is prone to exponential explosion when there are too many customers, and the solution cannot be obtained within an acceptable range. Moreover, the designed heuristic algorithm does not consider the constraint conditions.

In this paper, the OVRP model and Non-dominant sorting genetic algorithm (NSGA-II) multi-
objective genetic algorithm are introduced, and constraint operators are designed for the algorithm to solve the shortcomings in the above research work.

3. THE MODELING OF OPEN VEHICLE ROUTING PROBLEM WITH SOFT TIME WINDOWS

3.1 Problem Description
The route planning problem of take-out delivery is one of the hot spots in the research of short-distance vehicle route planning problem and the focus of the current society. In the university campus, for example, its surrounding is take-away food distribution produces the hottest spots, restricted by take-out delivery distance, can produce irregular area approximation of the campus as the center of take-out delivery range, in this takeaway business within the scope of quantity is limited, at the same time students in the school will produce a lot of orders at the period between eleven o’clock and thirteen o’clock.

Based on the example, have a certain number is inevitable in the small-time interval from a merchant order, this situation can be abstracted as a certain number of users and a business center need vehicle distribution task is as follows: delivering vehicles for starting businesses, providing distribution services to different location of the user, all delivery vehicle distribution is the distribution task. Under constraints such as the maximum vehicle load, the matching relationship between vehicles and users, and the time limit of platform estimation, etc., the goal of each distribution path planning, such as the minimum number of distributed vehicles, the minimum total cost of distribution and the maximum customer satisfaction, should be met as far as possible to maximize the benefits.

3.2 Problem Hypothesis
To simplify the above problems and make them easy to model abstractions, the following assumptions are made:

(1). The delivery vehicle visits each customer point only once and does not return to the merchant point;
(2). Distribution vehicles are only one type. In addition, the speed of the vehicle in the distribution process is always unchanged, ignoring the influence caused by non-human factors in the distribution process;
(3). The location coordinates of customer points and the demand for takeout have been obtained before delivery.

Symbol description:
- \( n \) represents the number of customers, \( V \) represents the node set, \( V = \{0,1,2,\cdots,n\} \), where 0 represents the merchant point; \( V' \) represents customers point set, \( V' = \{1,2,\cdots,n\} \); \( D \) represents the edge set, \( D = \{(i,j)|i,j \in V,i \neq j\} \); \( c_{ij} \) represents the transportation cost per unit distance between customer point \( i \) and customer point \( j \); \( d_{ij} \) represents the Manhattan distance between customer point \( i \) and customer point \( j \); \( m \) represents the maximum number of vehicles, \( R \) represents the vehicle set, \( R = \{k|k=1,2,\cdots,m\} \); \( c_r \) represents fixed distribution costs of vehicles; \( Q \) represents the maximum loading capacity of the vehicle; \( p_i \) represents the delivery service demand ordered by customer point \( i \); \( q_{ij} \) represents the delivery time from customer point \( i \) to customer point \( j \); \( t_i \) represents the time when the vehicle arrives at the customer point \( i \); \( e_i \) represents the expected delivery time of the customer in the customer point \( i \); \( e_i' \) represents the earliest delivery time allowed by the customer in the customer point \( i \); \( l_i \) represents the expected delivery time of the customer in the customer point \( i \); \( l_i' \) represents the expected delivery time set by the order platform of customer point \( i \); \( f_i(t_i) \) represents the customer satisfaction function of customer point \( i \); \( r_i \) represents the service time of customer point \( i \). \( x_{ki} \) is 0-1 variable, representing that the vehicle \( k \) goes from the customer point \( i \) to the
customer point \( j \) , \( x_{ij} = 1 \), otherwise \( x_{ij} = 0 \); \( y_{ik} \) is 0-1 variable, representing the customer point \( i \) serviced by the vehicle \( k \), otherwise \( y_{ik} = 0 \).

### 3.3 Modeling Process

Delivery and distribution for the route planning problem, from the merchants in the actual delivery and distribution after the meal to rooms and at the end of the room are not forced to return to the merchant for the next meal, at the same time platform will predict every order estimated time of delivery, so the take-away distribution problem can be abstracted as a multi-objective open vehicle routing problem with soft time windows. On this issue in the platform and merchants to consider as far as possible reduce the delivery cost of delivery, the user is considered food distribution in a relatively short period of time to complete, if within a certain period of time has not completed, it will inevitably bring punishment to the platform and business cost, so the multi-objective route planning model of objective function such as type (1) - (2) shown below:

\[
\begin{align*}
\min f_1 & = \sum_{i=0}^{n} \sum_{k=1}^{m} x_{ik} c_f + \sum_{i=0}^{n} \sum_{j=1}^{m} \sum_{k=1}^{m} x_{ik} c_y d_{ij} \\
\max f_2 & = \frac{1}{n} \sum_{i=1}^{n} f_i(t_i)
\end{align*}
\]

In formula (1): minimization goals for distribution vehicle number and distribution of minimum total cost, the goal of the first item for fixed vehicle distribution costs, the delivery platform in the distribution of vehicle allocation would produce the vehicle fixed distribution costs, the cost can be considered as the platform only need to start from merchants point 0 to connect with the merchants of the first point of the vehicle to pay the fees, so that part of the total cost is: \( \sum_{i=0}^{n} \sum_{k=1}^{m} x_{ik} c_f \), the second item is the cost incurred in vehicle distribution. In vehicle distribution, both energy consumption and vehicle loss will incur certain costs, which are linearly and positively correlated with the distance traveled by the vehicle. It may be assumed that the vehicle will incur certain costs \( c_y \) per unit distance. Since the model simulates the real suburban situation and the road network is relatively sparse, the Manhattan distance concept is introduced to calculate the actual distance between two points, and the cost is: \( \sum_{i=0}^{n} \sum_{j=1}^{m} \sum_{k=1}^{m} x_{ik} c_y d_{ij} \).

In formula (2): the maximization goal is the maximum mean value of customer satisfaction function. Traditional time window function can not be well meet the needs of users, users in a certain preference within the time window of time to get the service, but also for the actual situation of delivery and distribution, users want to delivery the sooner, the better service, so for this model should be endowed with soft time windows, the window borders blur, with membership function of fuzzy number to its representation, within the time period \( [e_i, l_i) \) for users expect service time, during this time, customer satisfaction was 100%. It’s willing to accept the service for the customer in the period of time \( [e'_i, l'_i) \cup [l_i, l'_i) \), but with time and delay in advance, customer satisfaction will drop dramatically. When user choose a take-out order platform will provide an estimated time \( l'_i \), according to the survey, more than 90% of the delivery orders will not exceed \( l'_i \), so when the delivery time exceeds \( l'_i \), it can be defined that customer satisfaction drops to 0, the user will make the refund, bad review operations with a high probability. Most of the customers on campus will want take-out delivery within a certain time, and customers can arrange for it by themselves. If take-out has been delivered in advance, customers cannot take food in time, it will happen that the food is cold or any other person take it away. Early or late delivery will bring penalty costs to the platform and merchants. Based on the above analysis and the introduction of the customer’s sensitivity factor \( \alpha, \beta \) to the event, the customer satisfaction function is defined by formula (3) as follows:\[9\):

\[
\begin{align*}
\end{align*}
\]
• Uniqueness constraint:
In take-out delivery, the uniqueness constraint is reflected in the one-to-one correspondence between delivery customers, vehicle volume and distribution center. The specific performance is as follows: (1) although a customer does not have limited distribution demand, the distribution task of a customer point is fixed before delivery, so each customer point only needs to be served once; (2) There is no single order in the distribution process, so all vehicles must start from the merchant point 0 to complete all tasks under the vehicle capacity limit; (3) In order to ensure the efficiency of vehicle distribution, the demand of a customer point only needs a single-vehicle distribution; (4) The customer point can only be served by one of all the vehicles from other customer points. Therefore, for the above four types of uniqueness constraints, they can be respectively expressed in formula (4) - (7):

\[
f(t_i) = \begin{cases} 
0, & t_i < e_i' \\
\left(\frac{t_i - e_i'}{e_i - e_i'}\right)^\alpha e_i', & t_i < e_i, \forall i \in V \\
1, & e_i \leq t_i < l_i \\
\left(\frac{l_i' - t_i}{l_i - l_i'}\right)^\alpha l_i', & l_i \leq t_i < l_i', \forall i \in V \\
0, & l_i' \leq t_i \end{cases} \tag{3}
\]

• Capacity constraints:
In take-out delivery, the delivery vehicle has a box specially designed to load the delivered goods. On the premise of ensuring the integrity of the delivered goods, the vehicle should load as many goods as possible. Therefore, the vehicle should first meet the total service demand without exceeding the vehicle's loading capacity, as shown in formula (8):

\[
\sum_{j=1}^{n} \sum_{i=1}^{m} x_{ijk} = 1, \forall j \in V' \tag{4}
\]

\[
\sum_{j=1}^{n} x_{ijk} = 1, \forall k \in R \tag{5}
\]

\[
\sum_{i=1}^{m} y_{ik} = 1, \forall i \in V' \tag{6}
\]

\[
y_{jk} = \sum_{i=1}^{m} x_{ijk}, \forall j \in V', k \in R \tag{7}
\]

• Time window constraints:
Due to the delivery vehicle arrives at a customer point and not directly to another customer point immediately. It needs to service customer, in the delivery and distribution, the service operation is waiting for customers to take orders. Therefore, we need to consider the relationship between the time of vehicle reach the next customer point and the time of arrival in the current customer, as shown in the following formula (9):

\[
\sum_{j=1}^{n} \sum_{i=1}^{m} x_{ijk} (t_i + r_i + q_{ij}) = t_j, \forall j \in V' \tag{9}
\]

• Time window violation penalty constraint:
In the process of delivery, if the actual delivery time is earlier than the delivery time acceptable to the customer or exceeds the estimated delivery time of the platform, the platform and the merchant will bear the excess penalty cost, so the customer point must be served within the time window \([e_i', l_i']\), as shown in formula (10) below:
From what has been discussed above, the open vehicle routing problem can be constructed into a 0-1 mixed integer planning model as follows:

$$
\begin{align*}
\text{min} & \quad \sum_{j=1}^{n} \sum_{k=1}^{m} x_{jk} c_{ij} + \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{m} y_{ik} c_{ij} d_{ij} \\
\text{subject to} & \quad \text{formula (3) - formula (10)}
\end{align*}
$$

$$
\text{max} \quad f_2 = \frac{1}{n} \sum_{i=1}^{n} t_i
$$

subject to formula (3) - formula (10)

4. **MULTI-OBJECTIVE AND MULTI-CONSTRAIN GENETIC ALGORITHM**

4.1 **Multi-Objective Genetic Algorithm NSGA-II**

In 2002, K. Deb et al. [10] proposed the non-dominant sorting genetic algorithm (NSGA-II) with elite strategy, which solved the main problems existing in NSGA and realized accurate search performance while greatly improving sorting speed. However, the algorithm does not consider constraint conditions, so it is not applicable to this model, and it needs to be improved on this basis. The principle of this algorithm can be summarized as follows:

- **Basic concepts:**
  - Pareto non-dominant: For the multi-objective minimization model, there is a vector $\alpha \in A$, s.t. $\forall \beta \in A$, $\alpha < \beta$;
  - Pareto frontier: Under the constraint of multi-objective optimization model, the surface formed on space by the set of all Pareto non-dominant solutions is Pareto frontier.

- **Define the symbols in the algorithm:**
  1. Feasible solution set is a set of solutions containing all feasible solutions; non-feasible solution set is a set of solutions containing all non-feasible solutions;
  2. _pareto_front_ is the ordinal number obtained by fast non-dominant sorting of feasible solutions; _pareto_front_lastlayer_ is the ordinal number of the last row after the fast non-dominant sorting of feasible solutions; _non_pareto_front_ is the ordinal number obtained by fast non-dominant sorting of non-feasible solutions;

4.2 **Design of the Algorithm**

![Figure 1 Logic diagram for the Constrained- NSGA II algorithm](image-url)
Based on the Constrained -NSGA II algorithm built in Part III and the NSGA-II algorithm in Part IV of this paper, a Constrained -NSGA II algorithm considering complex constraints is proposed. The algorithm has several operators, such as rapid non-dominant sorting and individual congestion sorting. The overall logic is shown in Figure 1.

(a). Establishing and coding of initial population:

This model deals with the open vehicle routing problem with soft time windows. A non-repeating natural number is used to encode merchant points and customer points. The corresponding gene of merchant points is: \( \{0\} \), and the corresponding gene of customer points is: \( \{1, 2, \cdots, n\} \). The initial population is generated by randomly generating codes for customer points and merchant points as the corresponding distribution sequence.

Customer points were assigned to different paths according to the constraints and decided to form a delivery path for the customer points meeting the constraints at the same time. Finally, customer point 0 was added at the head of the gene to form a complete single path for delivery.

(b). Crossover operator:

This model adopts the analog binary crossover method, which is suitable for floating-point encoding. Definition: The parent chromosome of the \( k \) gene is expressed as \( p_{1,k}, p_{2,k} \), and the progeny chromosome of the \( k \) gene is expressed as \( c_{1,k}, c_{2,k} \), then:

\[
\begin{align*}
    c_{1,k} &= \frac{(1-\beta_k)p_{1,k} + (1+\beta_k)p_{2,k}}{2} \\
    c_{2,k} &= \frac{(1+\beta_k)p_{1,k} + (1-\beta_k)p_{2,k}}{2}
\end{align*}
\]

where,

\[
    \beta_k = \begin{cases} 
        \frac{(2u_k)^{\eta}}{\eta}, u_k \in (0, 0.5] \\
        \frac{1}{2(1-u_k)^{\eta}}, u_k \in (0.5, 1)
    \end{cases}
\]

and \( u_k \in U(0,1) \), \( \eta \) represents the distribution function of the crossover operator, suppose \( \eta = 2 \).

(c). Mutation operator:

The optimal variation interval \([a,b]\) is defined, and the variation probability is \( g \). When \( g \leq a \), the algorithm searches blindly in the global state, and the time complexity increases exponentially. When \( g \geq b \), the algorithm falls into the local optimal. Therefore, this model adopts the method of reverse variation. The specific steps are as follows:

Step1: The starting and ending positions of several genes in a parent chromosome were randomly selected;

Step2: The genes in the selected gene fragment are treated in reverse order.

(d). Select operator:

The selection operator simulates the process of biological competition. The use of the roulette gambling selection operator will accelerate the convergence and lead to local optimization. Therefore, the selection operator of tournament competition is chosen as the way to select the parent generation. Through multiple iterations, until the number of selected individuals reaches the population size.

(e). Constrained evolutionary control operator:

Constrained evolutionary control operator is a process of dealing with constraints and optimizing algorithms. According to the actual model, the global optimal solution is generated at the boundary of the feasible region, which can be interpreted as the feasible solution reaches the optimal solution in the process of approaching the non-feasible solution. It is known that the non-feasible solution plays a great role in maintaining population diversity in genetics. Therefore, the dominant principle is improved, Pareto non-feasible solution ordering based on two-level virtual fitness function model is established, the utilization rate of non-feasible solution is increased, and a complete Pareto frontier satisfying the constraints is obtained.
The principle of the dominant and non-dominant relationship between feasible solutions and non-feasible solutions is as follows:

1. For a feasible solution, the closer the solution is to the Pareto front, the better;
2. For the non-feasible solution, firstly judge the degree of violation of constraint, and the less the violation degree is, the better the solution is. Secondly, if the violation degree is the same, the closer the solution is to the front of Pareto, the better the solution is.
3. Feasible solutions are superior to non-feasible solutions.

Due to the effect of the non-feasible solution, the above principle 3 is improved: the feasible solution is not strictly superior to the non-feasible solution, that is, the solution with the least degree of violation in the non-feasible solution has the same level as the solution in the feasible solution group. Suppose that the plane where feasible solution and non-feasible solution intersect in space is \( f(x) \), \( f(x+a) \) is the dynamic function that \( f(x) \) tends to the non-feasible domain, and \( f(x+a) \) is the first set of solutions that are searched from \( f(x) \) to non-feasible domain. Combined with the idea of simulated annealing, the solution with the least degree of violation in the non-feasible solution can enter into the feasible solution set with a certain probability of \( P = e^{\frac{f(x+a)-f(x)}{T}} \).

Determine the degree of constraint violation of chromosomes:

The violation degree function of the (in) equality constraint defined in the programming problem is the distance measure of the chromosome map in the feasible region of the solution space. Suppose that the multi-objective programming model has \( k \) equality constraints, \( j \) inequality constraints, \( h_i(x) = 0(i=1,2,\cdots,k) \) is expressed as the standard form of equality constraints, \( g_j(x) \leq 0(i=1,2,\cdots,j) \) is expressed as the standard form of inequality constraints, the equality constraints are composed of an infinite number of points, for points in the space, satisfy. For inequality constraints, \( \xi \) may be assumed as a buffer parameter, so that points in the space that are centered on all equality constraints and within the circle with radius \( \xi \) still satisfy the equality constraints. If the solution in the solution set satisfies the constraint, the violation degree function is 0; otherwise, the violation degree function is \( g_i(x) \); For the equality constraint, the violation degree function of the solution satisfying the constraint condition is 0, otherwise, the violation degree function is \( h_i(x)-\xi \). The violation degree function is shown as follows:

\[
G_i(x) = \begin{cases} 
\max \{0, g_i(x)\}, & 1 \leq i \leq j \\
\max \{0, |h_i(x)|-\xi\}, & j+1 \leq i \leq j+k 
\end{cases} \quad (13)
\]

For any solution in the solution set, the sum of the degrees of violation of all constraints is defined as:

\[
\phi(x) = \sum_{i=1}^{m} \left[ \max \{0, g_i(x)\} \right]^2 + \sum_{j=1}^{n} \left[ \max \{0, |h_j(x)|-\xi\} \right]^2 
\]

(14)

In the algorithm to solve the open vehicle routing problem with soft time windows, there are 9 constraints, including 5 equality constraints, 2 inequality constraints and two 0-1 variable constraints. Among them, 5 equality constraints and two 0-1 variable constraints have made the solution to satisfy this part of constraints through gene coding and data type conversion. Therefore, only inequality constraints are considered, that is, the sum of the degree of violation of all constraints can be reduced to:

\[
\phi(x) = \sum_{i=1}^{i} \left[ \max \{0, g_i(x)\} \right]^2 
\]

(15)

Among them, when \( \phi_1(x) = 0 \), feasible solution is put into a feasible solution set; otherwise, feasible solution is put into a non-feasible solution set. For two inequality constraints, to reduce the number of single constraint dimensional effect on the total violation degree, first of all, in violation of degree function normalized processing, get equation (16), the second after standardization function will be mapped to the range 0 to 1, due to focusing all solutions in violation of the number of constraints is not
all the same, so the violation degree after normalization function to a large extent depends on the number $N_u$ of the solutions of violating the constraints, by defining "solution and the distance from the feasible region" remove this effect, as shown in formula (17):

$$
\varphi_i(x) = \sum_{i=1}^{\max(0, g_i(x))} \frac{\max_j g_j(x)}{g_i(x)}, \ l = 1, 2, \ldots, j
$$

$$
\varphi(x) = \frac{\varphi_i(x)}{N_u}
$$

At least two virtual fitness functions are required in the fast non-dominant ordering of non-feasible solutions, so the mean value of inequality constraint violation degree function is defined, as shown below:

$$
\varphi_i(x) = \sum_{i=1}^{\max(0, g_i(x))} \frac{\max_j g_j(x)}{N_u}, \ l = 1, 2, \ldots, j
$$

(f). Individual congestion distance operator:
The individual congestion distance operator enables selective sorting of individuals with the same non-dominant order. The congestion distance of individual $i$ is the distance between the adjacent individuals $i-1$ and $i+1$ in the objective function space. In this model, individuals with large congestion distance are selected preferentially, which can make the target more evenly distributed in space and maintain the population diversity. The specific steps are as follows:

**Step1:** Initialize the crowding distance of individuals in the same layer, that is, $L[i]_L = 0$, where $L[i]_L$ represents the crowding distance of individual $i$;

**Step2:** Sort individuals at the same level in ascending order;

**Step3:** So that the sorting edge of the individual has a selection advantage. Given A large number M, let $L[0]_L = L[i]_L = 0$;

**Step4:** To calculate the congestion distance of the intermediate individuals:

$$
L[i]_L = L[i]_L + \frac{L[i - 1]_m - L[i - 1]_m}{f_{\max} - f_{\min}}
$$

Where $L[i]_m$ is the function value of the $i$th individual in the $m$ objective function, $f_{\max}$ and $f_{\min}$ are the maximum and minimum values of the $m$ objective function respectively.

**Step5:** For all objective functions, repeat **Step2-4** until the crowding distance of all individuals is obtained.

(g). Fast non-dominant sorting operator:
Fast non-dominant sorting stratifies the population according to the level of the non-inferior solution of the individual, aiming to guide the search to Pareto optimal solution. Fast non-dominant sort can be divided into fast non-dominant sort operators with feasible solutions and fast non-dominant sort operators with non-feasible solutions, which will be described in detail below:

- Fast non-dominant sorting operator for feasible solutions
  In the multi-objective programming problem, a hierarchical adaptive value model is adopted for the fast non-dominant ranking of feasible solutions, that is, a virtual fitness function is constructed according to the objective function, which is generally divided into two levels:
  (1) Fitness value of Pareto dominance relation based on feasible solution;
  (2) The crowding distance operator of the feasible solution (when the first order is used at the same time);

Let's define the $i$th chromosome to be $i_{front}$ at the first level and $i_{distance}$ at the second level. First, the first parameter $i_{front}$ is used to quickly sort the solutions in the feasible solution set. The steps are as follows:

**Step1:** Find all Pareto non-dominant individuals in the population, store them in set one, take set one
as the first level non-dominant individual set, and give the individuals in the set the same non-dominant order;

**Step2:** Define the individuals dominated by Pareto non-dominant individuals outside the set as \( n_i - 1 \) (\( n_i \) is the number of individuals dominated by the population individual);

**Step3:** The individuals of \( n_i - 1 = 0 \) are stored in set two, and the individuals in set two are assigned a corresponding non-dominant order according to the hierarchical operation in **Step1. Step2−3** is repeated until all the individuals are assigned a non-dominant order.

- Fast non-dominant sorting operator for non-feasible solutions

Considering the equality constraint \( h_j (x) = 0 \), this has been solved in the coding process. It is necessary to establish a hierarchical adaptive value model for non-feasible solutions by analogy with the fast non-dominant sorting operator of feasible solutions. In the Pareto sorting method of non-feasible solutions, two problems are proposed:

Because the population in the algorithm will converge to Pareto frontier eventually, we give priority to the problem of measuring the degree of violation of constraints, and work out the following sorting method:

1. Based on the virtual fitness value of Pareto dominance relation with non-feasible solution violating constraint degree, the basic concept mentioned above is selected here as the virtual fitness function: \( \varphi(x) \phi_i (x) \);

2. The virtual fitness value based on the objective function value of the infeasible solution is regarded as the second level evaluation standard, namely \( \varphi_m (x) \):

\[
\varphi_m (x) = \sqrt{\sum_{n=1}^{M} \left( \frac{f_n (x)}{\max f_n (x)} \right)^2}, M = 2, m = 1, 2
\]

In formula (20). Max f(x) represents the maximum value of M target functions. Is a two-objective combinatorial optimization problem, so M=2, m=1,2. Therefore, the smaller the value of formula (20) is, the better the performance of the solution on the objective function will be. Finally, formula (20) is considered as the distance between any non-feasible solution and Pareto non-dominant frontier.

The steps for quick non-dominant sorting of feasible and non-feasible solutions are as follows:

**Step1** Pareto sequence number is obtained, and the last layer is defined as `pareto_front_lastlayer`;

**Step2** The sequence number `non_pareto_front` of each layer is obtained by Pareto ordering of the non-feasible solutions according to the fast non-dominant sorting operator of the non-feasible solution;

**Step3** The individual with Pareto sequence number `non_pareto_front = 1` in the unfeasible solution set is taken out and stored in the feasible solution set. Then the Pareto sequence number of the feasible solution set is `pareto_front_lastlayer`, which is merged with the solution in the last layer of the feasible solution;

**Step4** According to the method of congestion sorting operator and combined with the fast non-dominant sorting operator of feasible solutions, the congestion sorting of feasible solutions in each layer of feasible solution set is completed and arranged in descending order;

**Step5** According to the method of congestion sorting operator and combined with the fast non-dominant sorting operator of non-feasible solutions, the congestion sorting of non-feasible solutions in each layer of non-feasible solution set is completed and arranged in descending order;

**Step6** In order to integrate the non-feasible solution and feasible solution into the same index system, the sequence number of non-feasible solution set Pareto is updated, and `pareto_front_new = non_pareto_front + pareto_front_lastlayer`;

**Step7** Sort the remaining non-feasible solutions in the non-feasible solution set;
The sorting results are shown in Figure 2:

![Figure 2 Fast non-dominant sort result graph](image)

The algorithm flow chart of the sorting is shown in Figure 3 below:

![Figure 3 Flow chart of fast non-dominant sorting algorithm](image)

(h). Elite strategy selection operator:
The individual with the highest fitness in the current population does not participate in the
hybridization and gene variation of the population, and the individual with the lowest fitness after participating in the genetic operation is replaced by it. The purpose of this operation is to ensure that the individuals with the best fitness can be retained intact in the offspring population.

And when the number of iterations is reached, the algorithm process is ended.

### 4.3 Example Analysis

| The vehicle number | 0  | 1  | 2  | 3  | 4  | 5  |
|--------------------|----|----|----|----|----|----|
| X-axis             | 4.39| 1.42| 0.96| 3.51| 2.77| 3.41|
| Y-axis             | 2.41| 3.37| 2.53| 2.52| 3.78| 2.73|
| Demand             | 1.01| 0.73| 0.59| 0.71| 0.80| 0.80|

| The vehicle number | 5  | 6  | 7  | 8  | 9  | 10 |
|--------------------|----|----|----|----|----|----|
| X-axis             | 3.41| 3.34| 2.19| 2.44| 1.11| 1.58|
| Y-axis             | 2.73| 1.55| 1.61| 3.71| 2.01| 4.54|
| Demand             | 0.80| 0.63| 1.36| 1.80| 1.99| 0.31|

Table 1 Basic Information of distribution

Table footnote: Keep two decimal places.

This paper randomly generated point location and customer demand information on the model simulation, randomly generated in the two-dimensional space two 0 to 5 digital simulation in the process of actual customer point of latitude and longitude, randomly generated at the same time 0 to 2 the digital simulation of real cases, single customer points take-out delivery requirements, use the Manhattan distance to get customer points with the customer, the distance between customers and merchants matrix. The following shows the distribution results simulated by one of the randomly generated Numbers.

The relevant parameters are set as follows: there are 10 customers, the unit transportation cost between customer points is $1/km, the fixed vehicle freight is $10, the maximum vehicle load is 5 kg, the speed is 10 km/h, the customer service time is 5 minutes, the earliest delivery time allowed by the customer is 60% of the lowest delivery time, the customer's expected early delivery time is 70% of the lowest delivery time. The expected delivery time of customers is 90% of the longest delivery time, and the expected delivery time set by the platform is the longest, and the sensitivity coefficient is 0.8. After several experiments and simulations, the number of selected populations is 50 and the number of iterations is 100.

Programming based on Matlab12.0 for the customer point location information, demand information, as shown in table 1, distribution information of each car, with the total distribution, total distribution, load factors, customer satisfaction, the total cost, delivery vehicles distribution route information as shown in table 2, the optimal path diagram as shown in figure 4 (the blue line that connects merchants points with the customer, the pink line that connects the client with the customer points):

### Table 2 Optimal distribution scheme implementation results

| Serial number | Delivery distance | Delivery time | Load factor | Distribution path |
|---------------|-------------------|---------------|-------------|-------------------|
| 4             | 12.88             | 97.26         | 99.02       | 0,7,1,3,9         |
| 7             | 13.51             | 111.04        | 99.68       | 0,8,2,10,5,4,6   |

| Indicators | Number |
|-----------|--------|
| TDD       | 26.3834|
| TDT       | 111.0395|
| LF        | 99.45  |
| CS        | 1      |
| TCDV      | 46.4   |

Table footnote: TDD: Total Delivery Distance/km; TDT: Total Delivery Time/ min; LF: Load Factor/%; CS: Customer Satisfaction; TCDV: Total Cost of Distribution Vehicle / USD
5. CONCLUSIONS
Considering the actual situation of take-out delivery around campus, this paper constructs a multi-objective model of open vehicle routing problem with soft time windows, and uses constrained NSGA-II with constraints to solve it. The simulation experiment is carried out by generating random numbers, and the results of the example analysis show that the optimal vehicle distribution path is obtained. This model can be applied to the take-out delivery industry. It is mentioned in the model that many parameters can be changed according to different actual conditions to meet the real distribution environment. The future development direction of the model is that the sensitivity coefficient of customers to the soft time window in customer satisfaction is generated according to the customer order and more uncertain factors in the take-out distribution process are considered. The ideal future state of the delivery model is the combination of OVRP with the robot obstacle avoidance model and the delivery demand generation prediction model, so as to produce a more realistic model and serve the reality better.

REFERENCE
[1] Li Xiangyong. Research on the model and algorithm for Vehicle Routing Problem [D]. Shanghai Jiao Tong University, 2007.
[2] Park Y, Chae J. A review of the solution approaches used in recent G-VRP (Green Vehicle Routing Problem)[J]. International Journal of Advanced Logistics, 2014, 3(1-2):27-37.
[3] Gu W, Cattaruzza D, Ogier M, et al. Adaptive large neighborhood search for the commodity constrained split delivery VRP[J]. Computers & Operations Research, 2019, 112.
[4] Baldacci R, Lim A, Traversi E, et al. Optimal Solution of Vehicle Routing Problems with Fractional Objective Function[J]. Transportation encc, 2020, 54.
[5] Bard J F, Huang L, Dror M, et al. A branch and cut algorithm for the VRP with satellite facilities[J]. IIE Transactions, 1998, 30(9):821-834.
[6] Yusuf, Baba, Iksan. Applied Genetic Algorithm for Solving Rich VRP[J]. Applied Artificial Intelligence, 2014, 28(10):957-991.
[7] Mohammed M A, Abd Gani M K, Hamed R I, et al. Solving vehicle routing problem by using improved genetic algorithm for optimal solution[J]. Journal of Computational encc, 2017:255–262.
[8] Kheiri A, Dragomir A G, Mueller D, et al. Tackling a VRP challenge to redistribute scarce equipment within time windows using metaheuristic algorithms[J]. EURO Journal on Transportation and Logs, 2019, 8(2).
[9] Hu Zuoan, Jia Yez, Li Bowei, et al. An optimization of the vehicle routing problem based on customer satisfaction [J]. Industrial Engineering Journal, 2019, 22(1): 100-107.

[10] Deb K, Pratap A, Agarwal S and Meyarivan T, "A fast and elitist multiobjective genetic algorithm: NSGA-II," in IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197, April 2002, doi: 10.1109/4235.996017.