

Research Article

Data-Driven Robust Optimization of the Vehicle Routing Problem with Uncertain Customers

Jingling Zhang ¹, Yusu Sun ¹, Qinbing Feng, ¹ Yanwei Zhao ¹ and Zheng Wang²

¹Key Laboratory of Special Equipment Manufacturing and Advanced Processing Technology, Ministry of Education, Zhejiang University of Technology, Hangzhou 310023, China

²The State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan, China

Correspondence should be addressed to Jingling Zhang; jlzhang@zjut.edu.cn

Received 18 May 2022; Revised 19 August 2022; Accepted 19 September 2022; Published 5 October 2022

Academic Editor: Qingling Wang

Copyright © 2022 Jingling Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the increasing proportion of the logistics industry in the economy, the study of the vehicle routing problem has practical significance for economic development. Based on the vehicle routing problem (VRP), the customer presence probability data are introduced as an uncertain random parameter, and the VRP model of uncertain customers is established. By optimizing the robust uncertainty model, combined with a data-driven kernel density estimation method, the distribution feature set of historical data samples can then be fitted, and finally, a distributed robust vehicle routing model for uncertain customers is established. The Q-learning algorithm in reinforcement learning is introduced into the high-level selection strategy using the hyper-heuristic algorithm, and a hyper-heuristic algorithm based on the Q-learning algorithm is designed to solve the problem. Compared with the certain method, the distributed robust model can effectively reduce the total cost and the robust conservatism while ensuring customer satisfaction. The improved algorithm also has good performance.

1. Introduction

To reduce transportation costs in planning vehicle transportation and distribution links, the vehicle routing problem (VRP) has always been a key problem in the field of logistics scheduling. The VRP was first proposed by Dantzig and Ramser in 1959 [1]; the basic problem is a capacity vehicle routing problem (CVRP). Under the constraints of meeting the known vehicle load and customer demand, the problem aims to minimize the vehicle transportation distance to serve all customers, thereby reducing the logistics distribution cost.

In the classical vehicle routing problem, it is generally assumed that all the information is complete and fixed; that is, the path planner grasps all the information before planning the path, including customer information and road network information, and then finds the optimal solution or satisfactory solution satisfying the constraint conditions. However, in the actual distribution process, the above

information is not invariable, and it is also difficult for the path planner to fully grasp the information of all distribution nodes in advance, resulting in the uncertainty of information, which will make the original optimization problem nonoptimal or even impossible. Studying the VRP with uncertain parameters is called the vehicle routing problem with uncertainty (UVRP).

At present, most of the research on UVRP is conducted for the uncertainty of customer demand or travel time. In real-world distribution, ignoring the impact of customer absence will bring many additional transportation costs and efficiency losses. It is of great significance to study the problem of uncertain customers. Lei et al. [2] first studied the uncertain customer and its zoning problem and proposed a two-stage solution. In the research of VRPUC, most problems are optimized by assuming that the probability distribution of customer existence is a deterministic value [3] or formulating the strategy of real-time re-optimizing the path [4]. However, the probability distribution of customer

presence cannot be accurately measured and can only be estimated based on historical data. At the same time, the re-optimization will lead to frequent return of vehicles to the warehouse to pick up goods, which will lead to higher transportation costs and lower efficiency. Therefore, in our study, we consider the uncertainty of customer presence probability and predict the probability distribution of customer presence based on the historical demand information of customers.

For UVRP, the most common method is to build an uncertainty model and then convert it into a deterministic model to solve them. Researchers have adopted many approaches to transform uncertainty models into deterministic models: Markovic et al. [5] regarded the urban garbage collection problem as a VRP with random demand and travel time and expressed the problem as a chance-constrained programming model with a normal distribution. Hou et al. [6] established a stochastic programming model of the VRPUC with a time window and solved it by using an adaptive genetic algorithm. Ge et al. [7] studied the electric VRP with random demand and active remedies and established a model with probability constraints. Souyris et al. [8] proposed a robust model for VRP with soft time window and service time uncertainty and verified it based on real-world data by branch and price method. Solano-Charris et al. [9] proposed a robust optimization method based on discrete scenes to deal with uncertain travel costs for VRP with uncertain travel costs. For uncertainty problems, stochastic programming requires exact probability distribution information of random variables, which cannot be obtained from observation. Classical robust optimization can solve this problem, but the solution is too conservative. Several heuristic algorithms have been proposed for UVRP, and these include variable neighborhood search [10], tabu search [11], and particle swarm optimization [12].

In contrast, distributed robust optimization theory has developed rapidly in recent years, which can make up for the shortcomings of classical robust optimization and stochastic optimization. It combines statistical learning and optimization theory and obtains a good enough solution by assuming that the parameters obey some possible distributions. At present, distributed robust optimization has been applied to energy scheduling [13, 14], outpatient scheduling [15, 16], fault detection [17], and other fields, and has achieved good results. However, the application of distributed robust optimization in UVRP is relatively rare, especially in solving VRPUC. It is also meaningful to use distributed robust optimization theory to solve UVRP.

Reinforcement learning, as a powerful decision-making tool, has attracted extensive attention. Naturally, we think of applying reinforcement learning to the high-level strategy of using a hyper-heuristic algorithm to select the low-level strategy. Zhang et al. [18] took the reinforcement learning algorithm based on the deep Q-network as the high-level strategy of the hyper-heuristic algorithm, used it to evaluate the performance of the underlying operator, and learned which operator to use in a specific situation to obtain the maximum reward through interaction with the environment. Qin et al. [19] introduced several metaheuristic

algorithms with different characteristics as the low-level heuristic strategies and introduced policy-based reinforcement learning as the high-level selection strategy. In addition, deep learning is used to extract hidden patterns in collected data to better combine the advantages of underlying heuristics. However, their proposed algorithm is complex and has poor performance. At the same time, we can see that reinforcement learning is increasingly considered in the design of hyper-heuristic algorithms, and great progress has been made in recent years.

For VRPUC, we proposed a data-driven distributed robust optimization method to solve this problem. By considering the uncertainty of customer existence probability, we established a distributed robust optimization vehicle routing model and the data-driven kernel density estimation method to fit the historical data and then optimize the robust model. A reinforcement learning algorithm based on Q-learning algorithm is designed to solve the above model. In summary, the main contributions of this paper are as follows:

- (1) Compared with the previous studies, which generally assumed that the presence of customers obeyed a certain probability distribution, we considered the uncertainty of the probability of the presence of customers.
- (2) We introduced the distributed robust optimization theory to establish a mathematical model for solving the problem and used the historical data of customer points to optimize the model. The experiments showed that the proposed method could solve this problem well.
- (3) We proposed a hyper-heuristic algorithm based on reinforcement learning to solve the model. We used Q-learning algorithm as a high-level selection strategy to select the low-level operators. Through experiments, we can find that our algorithm has positive effects.

2. VRPUC Model

2.1. Problem Description. The VRPUC is described as follows: there is a distribution center with known locations, and there is a group of customers with known locations. The demand of each customer is known, and the model of the vehicles is fixed. It is known that some customers have fixed demand and must have vehicles to serve them every time; the remaining customers present according to a deterministic probability, they can produce situations that do not need services, and the probability of presence is uncertain. A certain number of vehicles are assigned to deliver services to customers. Each customer can be served by only one vehicle, and the vehicle must meet the demand of each customer. The vehicle must return to the distribution center after serving all customers that should be served. Because of the expected demand, the vehicle will not meet the demand of all customers. At this time, the vehicle will return to the distribution center for replenishment and then return to continue

to serve the remaining customers. The goal is to solve the total distance cost, including replenishment distance, on the premise of meeting all customer demand.

2.2. Mathematical Model. The formulation for the VRPUC is as follows: let $i=0$ be the distribution center, the set of customers be represented by i ($i=1, 2, 3, \dots, L$), the set of vehicles be represented by k ($k=1, 2, 3, \dots, K$), and Q be the standard capacity of the vehicle. Each customer demand is represented by d_i ($i=1, 2, 3, \dots, L$). The uncertain demand probability of each customer is represented by p_i , and the p_i values are independent of each other; if the customer demand is determined, then $p_i=1$. The overall event set is represented by R , and a customer presence event is represented by ζ . R/ζ represents the remaining events after removing ζ events, that is, events for which the customer does not require services. The set of points for which the customer does not meet the demand is O , the penalty coefficient is ε , the distance cost is DT , the penalty cost is S , the total cost is $Cost(x_{ijk}, y'_{ik})$, and c_{ij} represents the distance from customer i to customer j . The following variables are defined:

$$y_{ik} = \begin{cases} 1, & \text{customer } i \text{ is delivered by vehicle } k, \\ 0, & \text{others,} \end{cases} \quad (1)$$

$$x_{ijk} = \begin{cases} 1, & \text{vehicle } k \text{ moves from customer points } i \text{ to } j, \\ 0, & \text{others.} \end{cases}$$

The mathematical model is as follows:

Objective:

$$\min \sum_{\zeta \subseteq R} \Pr(\zeta) \cdot Cost(x_{ijk}, y'_{ik}). \quad (2)$$

Subject to:

$$\sum_{i=1}^L d_i y_{ik} \leq Q \quad \forall k, \quad (3)$$

$$\Pr(\zeta) = \prod_{i \in \zeta} p_i \cdot \prod_{i \in R/\zeta} (1 - p_i), \quad (4)$$

$$Cost(x_{ijk}, y'_{ik}) = DT(x_{ijk}) + S(y'_{ik})$$

$$= \sum_{k=1}^K \sum_{i=0}^L \sum_{j=0}^L c_{ij} x_{ijk} + 2\varepsilon \cdot \sum_{k=1}^K \sum_{i \in O} c_{io} y'_{ik}, \quad (5)$$

$$\sum_{k=1}^K y_{ik} = 1 \quad \forall i, \quad (6)$$

$$\sum_{i \in L} \sum_{k \in K} x_{ijk} = 1 \quad \forall j = 1, 2, 3, \dots, L, \quad (7)$$

$$\sum_{i, j \in S \times S} x_{ijk} \leq |S| - 1, \quad S \subset \{1, 2, \dots, L\}, \quad (8)$$

$$S \neq \Phi \quad \forall k,$$

$$0 \leq p_i \leq 1, \quad (9)$$

$$y'_{ik} = \{0, 1\} \quad \forall k, i. \quad (10)$$

Equation (2) represents the target expected value, that is, the minimum expected cost. Constraint (3) indicates that the total demand of customer points served by each vehicle is not greater than the standard load capacity of the vehicle. Constraint (4) represents the probability product of the event. The former is the probability product of the occurrence of the demand event, and the latter is the probability product of the nonoccurrence of the demand event. Constraint (5) represents the total cost. The former is the calculation formula of distance cost, and the latter is the calculation formula of penalty cost. Constraint (6) ensures that each customer point is served. Constraint (7) ensures that each customer has only one vehicle for service. Constraint (8) eliminates the subloop. Constraint (9) indicates that the probability of customer demand is $[0, 1]$; that is, customers can have demand or probability demand in a certain distribution. Equation (10) judges whether the customer's demand is met. If so, the value is 0; otherwise, it is 1.

3. Robust Optimization of the VRPUC Formulation

3.1. Robust Model Optimization. Distributed robust optimization is generally used for the approximate distribution of known uncertain parameters. The actual distribution is derived from the combination of the approximate distribution and historical data. The worst case is selected in the actual distribution, and then, an improved solution is obtained from the worst case, which better optimizes the inconsistency between the approximate distribution and the actual distribution, resulting in overly conservative determination of the method. For the model with uncertain probability of customer presence, distributed robust optimization is adopted because the actual distribution of the probability of each customer presence is unknown, but the first-order and second-order moments can be obtained from historical data.

According to Reference [20], the minimum value is obtained when there is a probability distribution, that is, the expected minimum value. Therefore, (2) can also be written as

$$\min_{x, y} E_P \left(\prod_{i \in P} p_i \cdot \prod_{i \in R/P} (1 - p_i) \cdot Cost(x_{ijk}, y'_{ik}) \right), \quad (11)$$

where P represents the probability distribution of compliance, and $E_P(\bullet)$ represents the expected value of solution. The distributed robust optimization method is introduced to optimize it, and P' is designed as a set of probability measures, which includes all possible distributions of the demand probability p_i of each customer with a real situation, and its range is larger than the P set. It can be imagined that, if the worst case (maximum value and upper bound) is selected in the P' set and meets the required conditions, the situation in the P' set must also meet the required conditions. Therefore, (11) can be rewritten as

$$\min_{x,y} \sup_{P \in P'} E_{P'} \left(\prod_{i \in P'} p_i \cdot \prod_{i \in R \setminus P'} (1 - p_i) \cdot \text{Cost}(x_{ijk}, y'_{ik}) \right). \quad (12)$$

Because the latter part of (12) is only related to the result of path planning and is not directly related to the probability of an event at a certain point, constraint (12) can be written as

$$\min_{x,y} \sup_{P \in P'} E_{P'} \left(\prod_{i \in P'} p_i \cdot \prod_{i \in R \setminus P'} (1 - p_i) \cdot \text{DT}(x_{ijk}) + S(y'_{ik}) \right), \quad (13)$$

which implies

$$\begin{aligned} \min_{x,y} \sup_{P \in P'} E_{P'} \left(\sum_{k=1}^K \sum_{i=0}^L \sum_{j=0}^L c_{ij} x_{ijk} \cdot \prod_{i \in P'} p_i \cdot \prod_{i \in R \setminus P'} (1 - p_i) \right) \\ + 2\varepsilon \cdot \sum_{k=1}^K \sum_{i \in O} c_{io} y'_{ik}. \end{aligned} \quad (14)$$

According to (14), the key to the solution is the first half without considering the penalty cost. Because the penalty cost can only be determined by path planning, that is, according to the actual situation, it is impossible to obtain its maximum value through a functional expression. Therefore, (14) can be solved by using

$$\sup_{P \in P'} E_{P'} \left(\sum_{k=1}^K \sum_{i=0}^L \sum_{j=0}^L c_{ij} x_{ijk} \cdot \prod_{i \in P'} p_i \cdot \prod_{i \in R \setminus P'} (1 - p_i) \right). \quad (15)$$

The solution is divided into two stages. First, we solve the first half, that is,

$$\max_{P \in P'} E_{P'} \left(\sum_{k=1}^K \sum_{i=0}^L \sum_{j=0}^L c_{ij} x_{ijk} \cdot \prod_{i \in P'} p_i \cdot \prod_{i \in R \setminus P'} (1 - p_i) \right). \quad (16)$$

Expanding its expected formula gives

$$\max_{P \in P'} \int_{\varphi} \sum_{k=1}^K \sum_{i=0}^L \sum_{j=0}^L c_{ij} x_{ijk} \cdot \prod_{i \in P'} p_i \cdot \prod_{i \in R \setminus P'} (1 - p_i) dp. \quad (17)$$

Unifying p_i and $(1 - p_i)$ in (17) and rewriting them into corresponding piecewise functions yield

$$\max_{P \in P'} \int_{\varphi} \sum_{k=1}^K \sum_{i=0}^L \sum_{j=0}^L c_{ij} x_{ijk} \cdot \prod p_i dp, \quad (18)$$

$$\text{s.t.} \sum_{i \in L} \sum_{k \in K} x_{ijk} = 1 \quad \forall j \in L, \quad (19)$$

$$p_{\min} \leq p_i \leq p_{\max}, \quad (20)$$

$$p_i = \begin{cases} p_i, & i \in P', \\ 1 - p_i, & i \in \frac{R}{P'}. \end{cases} \quad (21)$$

Let μ be the average value obtained from historical data and σ^2 be the variance. Then, the probability distribution of set P has the following formulas:

$$\int_{P'} dp = 1, \quad (22)$$

$$\int_{P'} p_i dp = u_i, \quad i \in P', \quad (23)$$

$$\int_{P'} p_i^2 dp = \mu_i^2 + \sigma_i^2, \quad i \in P'. \quad (24)$$

Because there are only two results (whether the customer needs services), it can be seen that the P probability should roughly meet the binomial distribution; that is, the variable is a discrete variable, so (24) can also be expressed as

$$\begin{aligned} nb \cdot p_i = u_i, \quad nb = 1, 2, \dots, nb, i \in P', \\ nb \cdot p_i \cdot (1 - p_i) = \sigma_i^2, \quad nb = 1, 2, \dots, nb, i \in P'. \end{aligned} \quad (25)$$

Equations (18)–(21) can be transformed by using duality theory. Firstly, we transform the following equation:

$$\min \sum_{k=1}^K \sum_{i=0}^L \sum_{j=0}^L c_{ij} x_{ijk} \cdot \prod p_i, \quad (26)$$

and constraint condition (20) to

$$\max \sum_{i=0}^L (\beta_i p_{\min} - \alpha_i p_{\max}), \quad (27)$$

$$\begin{aligned} \text{s.t. } \beta_i - \alpha_i \leq c_{ij} x_{ijk}, \\ i, j = 0, 1, \dots, L, k = 1, 2, \dots, K, x_{ijk} = \{1, 0\}, \end{aligned} \quad (28)$$

where α_i and β_i are Lagrange multipliers. Therefore, (15) can be expressed as

$$\sup_{P \in P'} E_{P'} \left(\max_{\beta_i - \alpha_i \leq c_{ij} x_{ijk}} \sum_{i=0}^L \beta_i p_{\min} - \sum_{i=0}^L \alpha_i p_{\max} \right). \quad (29)$$

Similarly, (29) can be expressed as

$$\max_{\beta_i - \alpha_i \leq c_{ij} x_{ijk}} \int_{P'} \sum_{i=0}^L \beta_i p_{\min} - \sum_{i=0}^L \alpha_i p_{\max} dp. \quad (30)$$

Again, (31) and (22)–(24) can be transformed using the cone duality theory, and because $c_{ij} x_{ijk}$ is a constant with an undetermined value, the conditions can be omitted. Therefore,

$$\min \sum_{i=0}^L \alpha'_i u_i + \sum_{i=0}^L \beta'_i (u_i^2 + \sigma_i^2) + v, \quad (31)$$

$$\text{s.t. } v + \sum_{i=0}^L \alpha'_i p_i + \sum_{i=0}^L \beta'_i p_i^2 \geq \beta p_{\min} - \alpha p_{\max} \quad \forall p_i \in P', \quad (32)$$

where α'_i , β'_i , and v are also Lagrange multipliers. Because the conditions before transformation in the above relationship are equal sign relationships, the values of the three variables are $\alpha_i \in \mathbb{R}^{n \times n}$, $\beta_i \in \mathbb{R}^n$, and $v \in \mathbb{R}$. The variable after removing the subscript can be expressed as the corresponding matrix vector, and because this condition reflects the strong dual relationship, it can be written as follows by using constraint condition (32):

$$\beta p_{\min} - \alpha p_{\max} - \sum_{i=0}^L \alpha'_i p_i - \sum_{i=0}^L \beta'_i p_i^2 \leq v \quad \forall p_i \in P'. \quad (33)$$

$$\min_{\alpha, \beta \in \mathbb{R}^n} \left\{ \sum_{i=0}^L (\alpha'_i u_i + \beta'_i (u_i^2 + \sigma_i^2)) + \max_{p_i \in P'} \left\{ \beta p_{\min} - \alpha p_{\max} - \sum_{i=0}^L \alpha'_i p_i - \sum_{i=0}^L \beta'_i p_i^2 \right\} \right\}. \quad (36)$$

The final formula is

$$\min_{\alpha, \beta \in \mathbb{R}^n} \left\{ \sum_{i=0}^L (\alpha'_i u_i + \beta'_i (u_i^2 + \sigma_i^2)) + \max_{p_i \in P'} \left\{ \beta p_{\min} - \alpha p_{\max} - \sum_{i=0}^L \alpha'_i p_i - \sum_{i=0}^L \beta'_i p_i^2 \right\} \right\} + 2\varepsilon \cdot \sum_{k=1}^K \sum_{i \in O} c_{io} y_{ik}'. \quad (37)$$

In (36), the maximum part is a linear programming function when p_i is used as the variable value and other parameters are used as fixed values. It can be seen that it is a convex function with a maximum value. The maximum value occurs when

$$p_i = -\frac{\alpha'}{2\beta'}, \quad (38)$$

from which it can be obtained that

$$F_{\max}(p_i) = \frac{4\beta'(\alpha p_{\max} - \beta p_{\min}) - \alpha'^2}{4\beta'}. \quad (39)$$

According to the constraints (28), (34), and (35) and the attributes of whether the probability p_i itself is ≥ 0 or ≤ 0 , the optimal arrangement solution with variables $(\alpha_i, \beta_i, \alpha, \beta, y_{ik})$ can be solved, the minimum value can be determined, the fixed solution of p_i can be obtained, and the final deterministic robust optimization model with parameters can be obtained.

Therefore, if (33) is constant, the following condition must be met:

$$\max \beta p_{\min} - \alpha p_{\max} - \sum_{i=0}^L \alpha'_i p_i - \sum_{i=0}^L \beta'_i p_i^2 \leq v \quad \forall p_i \in P'. \quad (34)$$

According to semidefinite optimization theory, constraint (32) can also be written as

$$\begin{bmatrix} \alpha' & \frac{\beta'}{2} \\ \frac{\beta'}{2} & v - \beta^n p_{\min} + \alpha^n p_{\max} \end{bmatrix} \geq 0, \quad n = 1, 2, \dots, n. \quad (35)$$

Finally, (15) can be simplified to the following dual formula:

3.2. Optimizing Robust Models through Data Driving. Solving the VRPUC model based on distributed robust optimization requires knowing the distribution of uncertain customer probability variables and the distribution eigenvalues such as mean and variance. The samples obtained from simple data collection cannot represent the overall data, which is one-sided. To obtain the distribution characteristics of uncertain parameters more accurately, the data-driven non-parametric kernel density estimation method is used. The uncertain variable to be estimated is the probability that the customer point needs service, which is set as p . Suppose P_1, P_2, \dots, P_n are random samples of n discrete variables, which have a probability density function of $f(x)$. The specific form and parameters of the function are unknown. The empirical distribution function can be expressed as

$$F_n(p) = \frac{1}{n} (\text{the number of } P_1, P_2, \dots, P_n < p). \quad (40)$$

Suppose the kernel function is $K_O(p)$. Then, its density estimation function is

$$\begin{aligned}\hat{f}_n(p) &= \frac{1}{2h_n} \int_{p-h_n}^{p+h_n} dF_n(t) = \int_{p-h_n}^{p+h_n} \frac{1}{h_n} K_0\left(\frac{t-p}{h_n}\right) dF_n(t) \\ &= \frac{1}{nh_n} \sum_{i=1}^n K_0\left(\frac{P_i-p}{h_n}\right).\end{aligned}\quad (41)$$

Selecting the most widely used Gaussian function as the kernel function transforms the above formula to

$$\begin{aligned}\hat{f}_n(p) &= \frac{1}{nh_n} \sum_{i=1}^n \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(p-p_i/h_n)^2}{2}\right) \\ &= \frac{1}{\sqrt{2\pi}h_n} \sum_{i=1}^n \exp\left(-\frac{(p-p_i)^2}{2h_n^2}\right).\end{aligned}\quad (42)$$

Bandwidth h has a great influence on the $\hat{f}_n(p)$ distribution. An empirical method proposed in Reference [21] can be used to estimate bandwidth: for sample data close to a normal distribution or a bimodal probability function, we select

$$\begin{aligned}h_0 &= 1.06\sigma n^{(-1/5)} \text{ or} \\ h_0 &= \min\left\{\sigma, \frac{R}{1.34}\right\} 1.06n^{(-1/5)},\end{aligned}\quad (43)$$

where σ is the standard deviation of the sample data and R is the quartile distance obtained from the sample data. We next design a historical data sample of a customer point with 1000 historical data samples about a customer presence probability, and the customer point has a maximum probability that the demand probability is $>55\%$. We can use the above method to estimate its data distribution, as shown in Figure 1.

4. Hyper-Heuristic Algorithm

In the field of reinforcement learning, the Q-learning algorithm is widely used because of its classicality and easy to understand principle [22]. A hyper-heuristic algorithm based on the Q-learning algorithm (HH-QL) was designed. The design points of the Q-learning algorithm are mainly state, action, reward, punishment values, Q value, discount rate (γ), and learning rate (α).

4.1. Design of the State Value. $state = [s_1, s_2, \dots, s_n]$ represents the performance of the fitness value of the algorithm at this stage. For some problems where the final solution is determined, it is easy to judge the gap between this time and the final solution. For the VRP, the state of the algorithm cannot be clearly expressed. At this time, the fitness value is poor, or it may just jump out of the local optimal solution and go to another peak. When it finally reaches the peak, the fitness value may be better than this value. Therefore, the following $state$ value design method in [19] is adopted:

$$state = \frac{-(fit - fit')}{fit'} + Ck, \quad (44)$$

where fit represents the current fitness value, fit' represents the fitness value of the previous generation, Ck represents the cardinality of different types of operators, and the cardinality of mutation type operators is different from that of non-mutation-type operators, with values of 20 and 40, respectively, so as to distinguish the impact of different types of operators on the current state.

4.2. Design of the Action and Reward Values. $Action = [a_1, a_2, \dots, a_n]$ represents the next action that the algorithm will perform. Only low-level heuristics (LLH) underlying operators are selected, so $action$ represents the sequence number of the operator. For instance, a^1 is the designed No. 1 operator, that is, a single path operator. The operator proposed in Reference [23] is used as the underlying operator. It mainly includes three categories: local research (LLH-L), mutation (LLH-M), and location-based radial ruin (LLH-LR), for a total of 11 LLH underlying operators.

$reward = [r_1, r_2, \dots, r_n]$ indicates the evaluation of the impact of $action$ on the algorithm in the historical stage. The evaluation can be good or bad. A good evaluation will guide the algorithm to increase the probability of selecting the action when it encounters the same stage of $state$ in the future. A bad evaluation will reduce the possibility of selecting the action. The design of $reward$ has different effects because of the different types of operators. For instance, the impact of the 2-OPT operator is immediate, while the impact of destruction and reorganization operators is delayed. Therefore, $reward$ consists of immediate reward and delayed reward, which are, respectively, aimed at the operators that produce immediate influence and the operators that produce delayed influence. The design is as follows:

$$reward = I_re + F_re, \quad (45)$$

where I_re represents immediate reward value, given by

$$I_re = \begin{cases} I_re + 1, & fit < fit', \\ I_re, & fit = fit', \\ I_re - 1, & fit > fit'. \end{cases} \quad (46)$$

And F_re represents the delayed reward value.

If the fitness value is improved, the immediate reward is +1; otherwise, it is -1; the delayed reward for the immediate reward operator is always set to 0. For the delayed reward operator, its value is set to 0 in the current calculation. At this time, the serial number of the delayed reward operator and relevant information, such as the Q value, are saved. If the fitness value has been improved before the next delayed operator appears, the value of the corresponding operator reward is +1, and its Q value and other information will be updated at the same time.

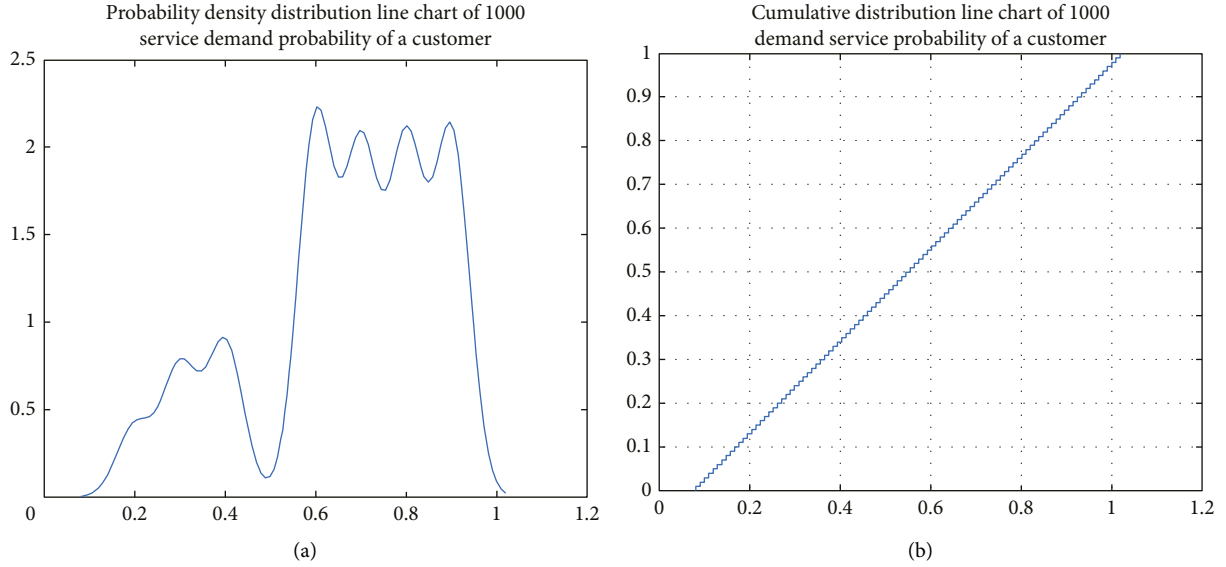


FIGURE 1: Customer data distribution charts. (a) Probability density distribution line chart and (b) cumulative distribution line chart.

4.3. Design of the Q Value. The Q value is the function value calculated by according to the *reward* value and the Q history value. It is used to prevent the evaluation from being too one-sided and affecting the trend of the algorithm based on using *reward* only. The calculation formula is as follows:

$$Q_{t+1}(s_t, a_t) = Q(s_t, a_t) + \alpha \cdot [r_{t+1} + \gamma \cdot \max_a Q(s_{t+1}, a) - Q(s_t, a_t)], \quad (47)$$

where s represents *state*, a represents *action*, r represents *reward*, t represents the corresponding value of the previous generation, $t+1$ represents the updated value of this generation, and \max_a represents the value with the largest Q value among all *action* values.

4.4. Design of γ and α Values. γ indicates the impact on future reward and punishment values. If the γ value is larger, more emphasis is placed on the future prediction results; that is, it will have a more positive view of the current behavior. In contrast, if the γ value is smaller, more emphasis is placed on the current income and there are stricter requirements for the current behavior. The value of γ has been studied in detail in the literature [23], and the recommended value is 0.8. In most cases, it has a good effect on the algorithm.

α represents the learning degree of the Q value function. Larger α values indicate that the new value replaces the old value to a greater extent. In contrast, smaller α values indicate that the old value still occupies an absolute position in the data, and the algorithm pays more attention to the historical effect of the action. In the VRP, the algorithm may converge to a local optimal solution, so the α value needs to be increased when the fitness value has not been improved

for a long time. By using the characteristics of reciprocating fluctuation of the cosine function, the α value can be determined as

$$\alpha_t = \cos\left(\frac{\pi}{2t_{\max}} + n \cdot 2\pi\right), \quad (48)$$

$$\text{mod}\left(\text{SN}, \frac{t_{\max}}{200}\right) = 0, \quad n = 0, 1, \dots, N.$$

The SN value is designed as the record of the number of iterations for which the fitness value is not improved.

4.5. Other Considerations. To avoid the situation that, when the probability of the customer presence is very high, the service path has been arranged in the path, but the final customer point does not generate the required service, and the bundling customer point method was adopted before the population generation. The method can avoid a sharp increase in cost resulting from the long path between the two customer points before and after the unserved customer points to generate services. The bundling principle is based on bundling nearby according to the distance between customer points.

The algorithm flow is as follows (see Figure 2):

Step 1: Bundle the customer points. First, uncertain customers are bundled with other fixed customers. Let the uncertain customer point be UC_i and judge the corresponding j fixed customer point according to the original distance matrix. If the distance c_{ij} is the minimum value compared with other fixed customer points, the customer point j is bundled with i to form a new customer point $C[i, j]$.

Step 2: Initialize the population. Randomly generate N_{pop} group individuals, generate feasible solution

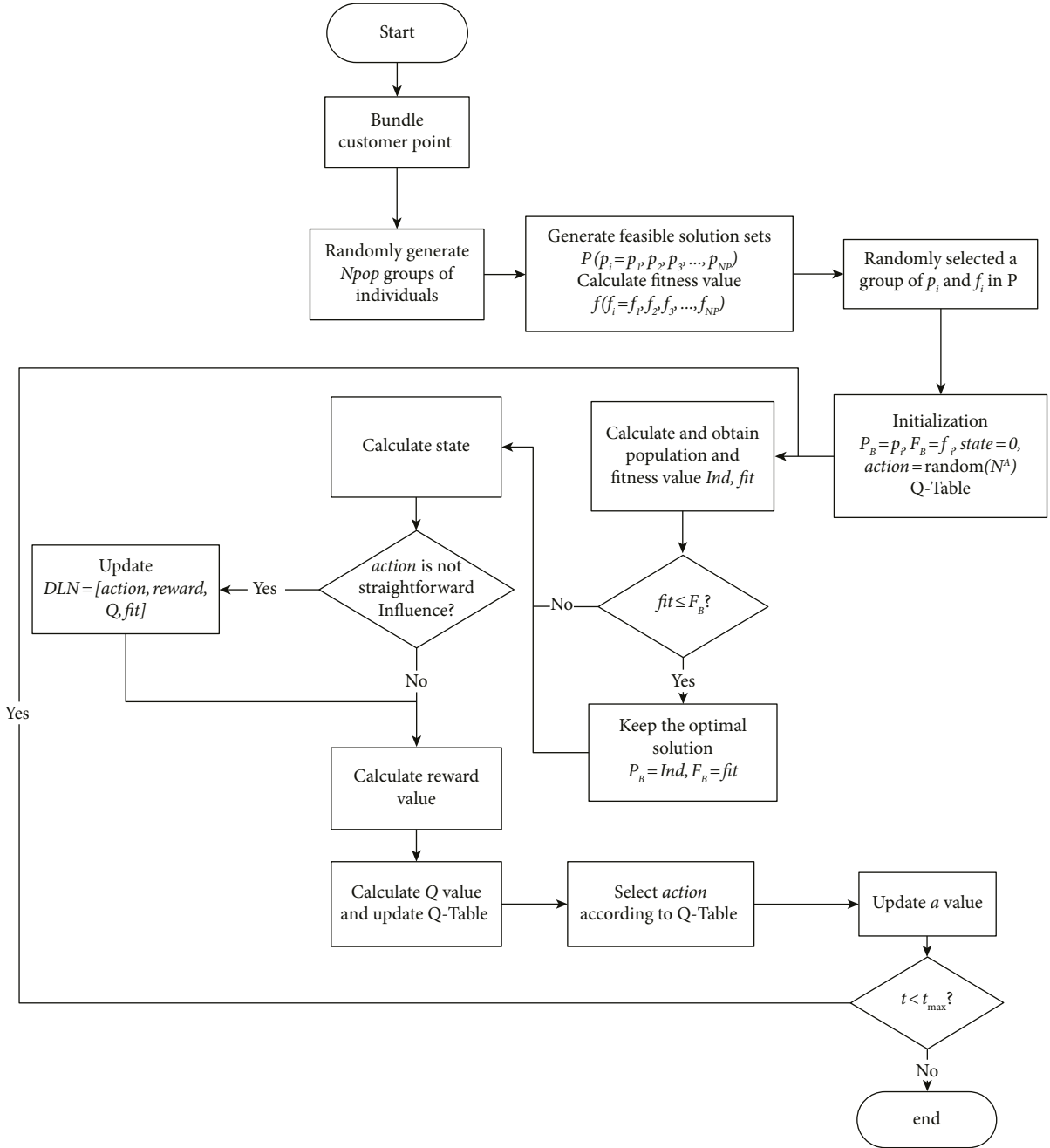


FIGURE 2: Algorithm flowchart.

group $P(p_i = p_1, p_2, \dots, p_{NP})$, calculate the fitness value as $f(f_i = f_1, f_2, f_3, \dots, f_{NP})$, and randomly select a group of P . Initialize the global optimal population P_B and the optimal solution F_B , initialize the $state$ value to 0, initialize the $action$ to any 11 operators, and initialize the Q-Table to 0.

Step 3: Calculate and update. The initialized $action$ operator is used to search the solution to obtain a new individual Ind and fitness value fit . At the same time, if $fit \leq F_B$, this proves that the solution at this time is better, then update the global optimal population P_B and F_B so that $P_B = Ind$ and $F_B = fit$.

Step 4: Calculate $state$. First, we judge whether to accept the solution according to the simulated annealing. If it is accepted, the $state$ value is calculated. Otherwise, the value remains unchanged.

Step 5: Evaluate $action$. We determine the operator corresponding to the $action$ currently. If it is an operator that has a delayed impact on the algorithm and if DLN is the empty set, we record the corresponding $DLN = [action, reward, Q, fit]$. If it is not the empty set, the history value in the corresponding Q-Table will be updated again.

Step 6: Calculate *reward* and *Q* values. We calculate *reward* and *Q* values according to equations (45)–(47) and update the *Q*-Table.

Step 7: Select action. According to the *Q*-Table, we select the *action* with the largest *Q* value corresponding to the *state* as the next *action* of the algorithm.

Step 8: Update learning rate α . We update learning rate α according to equation (48).

Step 9: Determine the iteration. If the current iteration number $t \leq t_{\max}$, we return to Step 4 for reiteration; otherwise, we exit the algorithm.

5. Computational Experiments

5.1. Implementation Aspects, Configuration of Parameters, and Instances. The algorithm was coded in MATLAB, and all experiments were conducted on a computer with an Intel Core i5-3230M and 12 GB of RAM. After repeated testing, the parameters used in the algorithm are set to the following: discount rate $\gamma = 0.8$ in the *Q* value function, the initial value of $\varepsilon = 0.5$, iterative maximum iteration $t_{\max} = 10^6$, empirical pool $N^E = 800$, and the number of samples selected for learning $N^S = 600$.

All calculation instances (<https://www.bernabe.dorrnsoro.es/vrp/>) in set A in the CVRP were selected for experiment. In each calculation instance, 25% of the total customer point scale was randomly selected as the uncertain customer. The random probability was designed at 0.3–0.9, and different interval conditions such as [0.3, 0.5] and [0.6, 0.9] were used to generate 1000 random {0,1} samples that are not uniformly distributed, where 0 represents no demand and 1 represents demand. The demand of the next customer point can only be known when the vehicle arrives at the previous customer point. The demand probability was counted every 10 times in 1000 random samples. Through the method described above, a random demand quantity based on the demand quantity of the original customer point was generated, and 1000 historical samples were generated for each customer point (and, at a customer point with defined demand, the demand remains constant). Because there are many uncertain customer points in each instance, they were not listed one by one. Only the statistical mean μ and variance σ^2 of some customer points in the randomly selected instance were displayed, as listed in Table 1.

5.2. Comparison with Other Algorithms. To test the performance of the algorithm, the HH-QL algorithm is compared with the literature [24–26]. The results are shown in Table 2. Each calculation instance is calculated 20 times, the shortest distance of the total traveling path is the objective function, and the smaller the distance is, the better the performance of the algorithm is. Column BKS represents the optimal distance solution for the instance, column Min represents the optimal distance solution obtained by the algorithm, column Avg represents the average of the optimal distance solution, and column DEV represents the gap between its costs and

BKS. The DEV is calculated is shown in (49). Also, the values in bold refer to the optimal solutions.

$$\text{DEV} = \frac{\text{Min} - \text{BKS}}{\text{BKS}}. \quad (49)$$

It can be seen from the results that the proportion of the optimal solution obtained by LNS-ACO, HVNSOS, and OHGA in the all instances is 62.96%, 74.00%, and 51.85%, respectively, and the proportion of HH-QL is 81.48%. Compared with other algorithms, HH-QL algorithm has some advantages in the overall optimal rate. The average deviation of each algorithm for the optimal solution of 27 instances is 0.60%, 0.13%, 0.37%, and 0.07%. HH-QL is 0.53%, 0.09%, and 0.06% lower than the other three algorithms. In all instances with more than 60 customers, the algorithm proposed in this paper is relatively more accurate than other algorithms. It can be concluded that the algorithm in this paper has a better search effect on the set a standard example compared with the other three algorithms of multigroup search.

5.3. Comparison with Certain Parameters Problem. To test the effectiveness of the distributed robust model and algorithm mentioned above, the proportion of nonsatisfaction customers and the additional cost caused by nonsatisfaction customers are evaluated.

Table 2 lists the path distance cost and customer satisfaction ratio of each calculation example after robust optimization obtained from the calculation example of set A. The greater the percentage of cost reduction and the larger the proportion of customer satisfaction after optimization, the better the optimization effect. The instance column lists the serial number of the problem, Certain in the Type column represents certain parameters problem, Robust represents robust optimization problem of uncertain parameters, the Cost column lists the distance cost, the Decrease column lists the percentage of the expected cost lower than the original planned cost, and the Total column lists the proportion of path planning with minimum cost that can be successfully met in 1000 historical samples with uncertain probability of customer presence. E1 represents the proportion of samples that can be satisfied by all customer points in the total sample, E2 represents the proportion that two customer points are not satisfied, and E3 represents the proportion that three or more customer points are not satisfied.

After planning the route, if the demand of the customer point is not met, the vehicle immediately returns to the origin for replenishment and then returns to the customer point that does not meet the demand to continue the distribution service according to the original route. Table 3 shows the set examples of measuring the additional distance cost incurred by returning to the warehouse to pick up the goods and continue delivery because the customer demand cannot be satisfied. The shorter the cost of increased distance, the better the robust optimization effect. The increased cost is 0, which means that the vehicle does not need to return to the warehouse for delivery. If the penalty

TABLE 1: Mean and variance of some customer points.

Instance	μ	σ^2	Instance	μ	σ^2
A-n44-k6.2	0.78	0.21	A-n80-k10.16	0.84	0.13
A-n44-k6.6	0.65	0.31	A-n80-k10.26	0.77	0.21
A-n44-k6.22	0.64	0.26	A-n80-k10.32	0.93	0.08
A-n44-k6.26	0.58	0.25	A-n80-k10.33	0.55	0.21
A-n44-k6.27	0.80	0.10	A-n80-k10.39	0.53	0.25
A-n44-k6.28	0.50	0.26	A-n80-k10.45	0.40	0.07
A-n44-k6.29	0.60	0.27	A-n80-k10.52	0.84	0.12
A-n44-k6.30	0.62	0.20	A-n80-k10.60	0.65	0.31
A-n44-k6.35	0.78	0.11	A-n80-k10.71	0.77	0.25
A-n44-k6.36	0.41	0.06	A-n80-k10.73	0.67	0.21
A-n44-k6.41	0.62	0.25	A-n80-k10.74	0.74	0.25
A-n80-k10.2	0.59	0.22	A-n80-k10.76	0.85	0.10
A-n80-k10.3	0.60	0.27	A-n80-k10.77	0.65	0.31
A-n80-k10.7	0.52	0.20	A-n80-k10.78	0.65	0.21
A-n80-k10.13	0.86	0.10	A-n80-k10.80	0.35	0.05
A-n80-k10.15	0.85	0.13			

TABLE 2: CVRP set A standard instance test results.

Instance	BKS	LNS-ACO [24]		HVNSOS [25]		OHGA [26]		HH-QL		
		Min	DEV%	Min	DEV%	Min	DEV%	Min	Avg	DEV%
A-n32-k5	784	784	0	748	0	748	0	784	784	0
A-n33-k5	661	661	0	661	0	661	0	661	661	0
A-n33-k6	742	742	0	742	0	742	0	742	742	0
A-n34-k5	778	778	0	778	0	778	0	778	778	0
A-n36-k5	799	799	0	799	0	799	0	799	799	0
A-n37-k5	669	669	0	669	0	669	0	669	669	0
A-n37-k6	949	949	0	949	0	949	0	949	950	0
A-n38-k5	730	730	0	730	0	730	0	730	730	0
A-n39-k5	822	822	0	822	0	822	0	822	824	0
A-n39-k6	831	831	0	831	0	833	0.24	831	833	0
A-n44-k6	937	937	0	937	0	937	0	937	939	0
A-n45-k6	944	958	1.48	944	0	953	0.95	944	960	0
A-n45-k7	1146	1146	0	1146	0	1146	0	1146	1149	0
A-n46-k7	914	914	0	914	0	914	0	914	914	0
A-n48-k7	1073	1084	1.03	1073	0	1073	0	1073	1074	0
A-n53-k7	1010	1010	0	1010	0	1017	0.69	1010	1016	0
A-n54-k7	1167	1167	0	1167	0	1167	0	1167	1171	0
A-n55-k9	1073	1073	0	1073	0	1074	0.09	1073	1074	0
A-n60-k9	1354	1354	0	1354	0	1355	0.07	1354	1358	0
A-n61-k9	1034	1067	3.19	1035	0.09	1035	0.10	1035	1061	0.10
A-n62-k8	1288	1308	1.55	1291	0.23	1308	1.55	1291	1305	0.23
A-n63-k9	1616	1649	2.04	1628	0.74	1630	0.87	1616	1629	0
A-n63-k10	1314	1329	1.14	1319	0.38	1329	1.14	1318	1321	0.03
A-n64-k9	1401	1415	1.00	1414	0.93	1416	1.07	1412	1418	0.79
A-n65-k9	1174	1185	0.94	1177	0.26	1184	0.85	1174	1179	0
A-n69-k9	1159	1170	0.95	1159	0	1170	0.95	1159	1168	0
A-n80-k10	1763	1815	2.95	1779	0.91	1790	1.53	1776	1789	0.74
Average			0.6		0.13		0.37			0.07

coefficient is set to 1, the penalty cost is the distance cost of the driving path. The Customer number column lists the maximum number of customers who do not meet the demand of customers in 1000 historical samples. The More cost column lists the increased cost of returning to the origin by not meeting the demand of customer points in one of 1000

historical samples. The Maximum and Minimum columns list the maximum and minimum costs, respectively. The Fact cost column lists the distance cost obtained by driving 1000 historical samples according to the actual customer points. The symbol — indicates that there is no dissatisfaction with customers for certain solutions (Table 4).

TABLE 3: VRPUC set A standard instance test results.

Instance	Type	Cost	Decrease (%)	Total		
				E1 (%)	E2 (%)	E3 (%)
A-n44-k6	Certain	937	—	—	—	—
	Robust	737	21.34	100	0	0
A-n45-k6	Certain	944	—	—	—	—
	Robust	707	25.10	99.8	0.2	0
A-n45-k7	Certain	1146	—	—	—	—
	Robust	897	21.73	100	0	0
A-n46-k7	Certain	914	—	—	—	—
	Robust	715	21.77	99.8	0.2	0
A-n48-k7	Certain	1073	—	—	—	—
	Robust	829	22.74	99.8	0.2	0
A-n53-k7	Certain	1010	—	—	—	—
	Robust	823	18.51	99.4	0.6	0
A-n54-k7	Certain	1167	—	—	—	—
	Robust	860	26.31	99.8	0.2	0
A-n55-k9	Certain	1073	—	—	—	—
	Robust	873	18.64	99.6	0.3	0.1
A-n60-k9	Certain	1354	—	—	—	—
	Robust	1011	25.33	99.9	0.1	0
A-n61-k9	Certain	1034	—	—	—	—
	Robust	854	17.41	99.9	0.1	0
A-n62-k8	Certain	1288	—	—	—	—
	Robust	998	22.52	99.6	0.3	0.1
A-n63-k9	Certain	1616	—	—	—	—
	Robust	1127	30.26	99.8	0.2	0
A-n63-k10	Certain	1314	—	—	—	—
	Robust	944	28.16	99.9	0.1	0
A-n64-k9	Certain	1401	—	—	—	—
	Robust	1035	26.12	99.8	0.1	0.1
A-n69-k9	Certain	1159	—	—	—	—
	Robust	949	18.12	99.6	0.2	0.2
A-n80-k10	Certain	1763	—	—	—	—
	Robust	1316	25.35	99.8	0.1	0.1

Table 2 lists the test experiments of the set A instance, along with the certain and robust optimization methods in terms of expected value and customer point satisfaction. Because the paths obtained by the certain method account for the presence of all customer points, the Total column does not exist. The path obtained by the robust optimization method accounts for the expectation of the probability of customer presence. The demand of the customer point is appropriately reduced according to the probability of customer point presence. Therefore, when the customer point actually exists, the vehicle load is insufficient, resulting in the failure to meet the demand of a certain point. If replenishment is returned, it will affect this point and subsequent points. Combining this with the comparison diagram of cost value and expected value in Figure 3, we can obtain that the overall distance value after robust optimization is less than that obtained using the certain method, and the proportion range of reduction is 17%–30%, which is obvious, reflecting the adjustment of distance cost. Moreover, in the customer satisfaction diagram, it can be seen that the height of the E1

column almost accounts for the absolute value of the whole column diagram of an instance. Meanwhile, with the increase of the scale of customer points, there is a high probability that three or more customer points are not satisfied; that is, the more customers there are, the greater possibility of uncertain parameter disturbance, the greater the difficulty in determining the degree, and the higher the requirements for the planned path. It can be concluded that the satisfaction of the robust optimized path to the customers in the actual sample is still high, and there is a low probability that one or two customers will not be satisfied.

Table 3 lists the solution obtained after calculating the actual transportation distance cost according to the customer presence data of 1000 historical data samples in the set A test experiment and the solution of the distance cost generated by replenishment in case of replenishment. As shown in Figure 4, regardless of the maximum value, minimum value, or average value, the line after robust optimization is almost below the line of certain results; that

TABLE 4: VRPUC set A cost results.

Instance	Type	Customer number	Fact cost			More cost		
			Maximum	Minimum	Average	Maximum	Minimum	Average
A-n44-k6	Certain	—	937	898	920.68	—	—	—
A-n44-k6	Robust	0	942	894	919.42	0	0	0.00
A-n45-k6	Certain	—	944	895	914.51	—	—	—
A-n45-k6	Robust	2	937	877	901.79	168	0	0.25
A-n45-k7	Certain	—	1146	458	665.94	—	—	—
A-n45-k7	Robust	0	1127	504	709.54	0	0	0.00
A-n46-k7	Certain	—	914	409	723.10	—	—	—
A-n46-k7	Robust	1	929	415	738.80	94	0	0.19
A-n48-k7	Certain	—	1073	1054	1065.01	—	—	—
A-n48-k7	Robust	2	1090	1041	1070.83	144	0	0.29
A-n53-k7	Certain	—	1010	969	995.00	—	—	—
A-n53-k7	Robust	2	992	913	965.07	92	0	0.33
A-n54-k7	Certain	—	1167	1143	1154.22	—	—	—
A-n54-k7	Robust	2	1178	1098	1138.61	196	0	0.35
A-n55-k9	Certain	—	1073	1026	1055.33	—	—	—
A-n55-k9	Robust	3	1061	992	1032.20	252	0	0.34
A-n60-k9	Certain	—	1355	1323	1344.10	—	—	—
A-n60-k9	Robust	2	1390	1281	1333.11	258	0	0.26
A-n61-k9	Certain	—	1037	1000	1022.09	—	—	—
A-n61-k9	Robust	1	1007	959	985.07	110	0	0.11
A-n62-k8	Certain	—	1298	1251	1280.25	—	—	—
A-n62-k8	Robust	3	1233	1174	1213.00	332	0	0.91
A-n63-k9	Certain	—	1616	1595	1606.07	—	—	—
A-n63-k9	Robust	1	1558	1522	1540.38	102	0	0.20
A-n63-k10	Certain	—	1318	429	943.18	—	—	—
A-n63-k10	Robust	2	1258	398	913.75	104	0	0.10
A-n64-k9	Certain	—	1415	1357	1391.67	—	—	—
A-n64-k9	Robust	3	1394	1265	1324.91	296	0	0.34
A-n69-k9	Certain	—	1166	1121	1152.51	—	—	—
A-n69-k9	Robust	2	1148	1099	1127.72	188	0	0.41
A-n80-k10	Certain	—	1779	671	993.14	—	—	—
A-n80-k10	Robust	4	1728	562	939.46	358	0	0.52

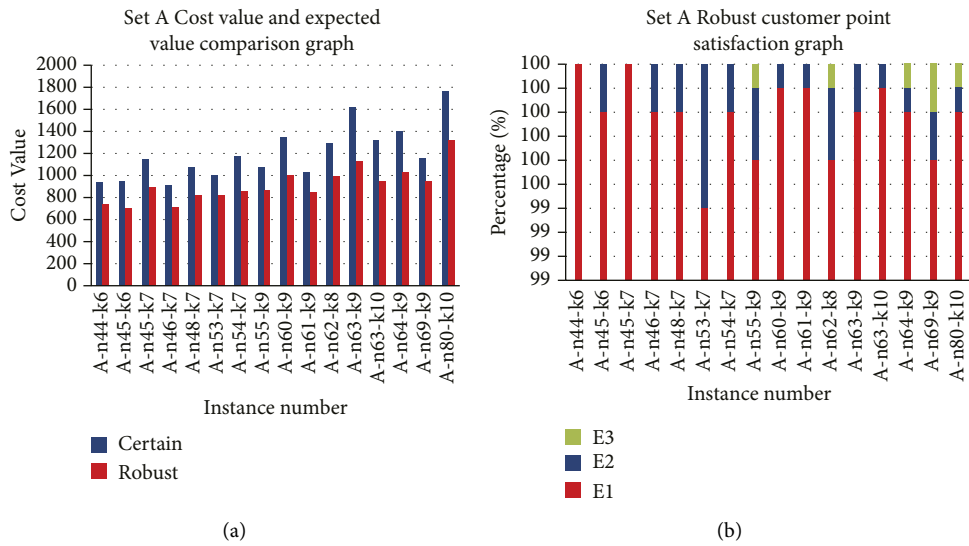


FIGURE 3: Set A results charts. (a) Set A cost and expected value comparison graph. (b) Set A robust customer point satisfaction graph.

is, the prior path after robust optimization can reduce the transportation distance cost and prevent unnecessary cost waste compared with the path obtained by using the certain

method. In the maximum figure, when the customer scale is large, the degree of savings is higher, and when the customer scale is small, the result is not obvious. In the graphs of



FIGURE 4: Set A cost comparison graphs. (a) Set A fact cost maximum comparison graph, (b) set A fact cost minimum comparison graph, (c) set A fact cost average comparison graph, and (d) set A robust average sum difference graph.

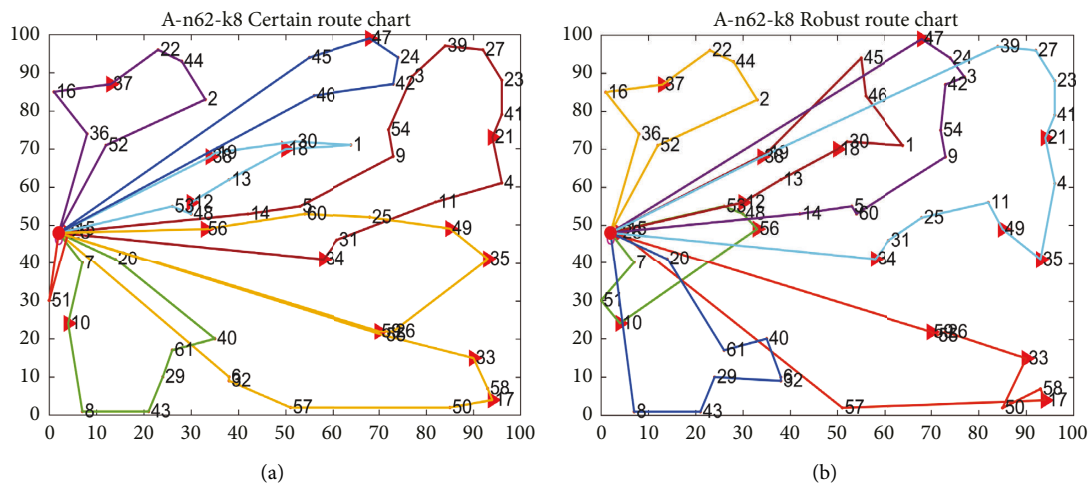


FIGURE 5: A-n62-k8 certain (a) and robust (b) route comparison charts.

minimum value and average value, the effect is quite obvious after the customer point scale reaches 48.

As shown in Figure 4(d), if the penalty coefficient is assumed to be 1 and only the impact of replenishment

distance cost is considered, the average value of the actual distance cost after robust optimization of 1000 historical data samples about uncertain customers is added to the average value of the penalty cost caused by replenishment;

that is, the total cost, which is the difference graph compared with the average value of the actual distance cost obtained by the certain method, is certain cost average – (robust cost average + more cost average). It can be seen from the figure that the total cost of most of the two instance sets after robust optimization is less than that of the certain method. There are only three exceptions in set A, accounting for 18.75% of the total. When the line chart is almost approximate, owing to the low replenishment probability and small replenishment distance of the path obtained after robust optimization, the total cost maintains a downward trend, which fully reflects the advantage of the robust model in cost saving.

Figure 5 shows the path diagram of the certain and robust optimization methods with a large total cost reduction in the set. The points of the triangle represent uncertain customer points. From the figure, one can see that the path distributions of the two methods are roughly the same. Because the certain method looks for the shortest path graph with customers whose need for services is certain, there is less intersection between paths, while the robust optimization method pursues the reasonable distribution of uncertain customers. Because the bundling point method is used in the algorithm, in robust optimization, uncertain customer points tend to be on the same path as the nearest customer points, such as points 49, 35, and 4 in A-n62. On the same path as the nearest point, when the uncertain customer point does not have a demand or there is a demand, the change of the total path is small. Of course, from the distribution of paths, because of the many path intersections, the proposed algorithm may design more iterations in the calculation, so the result is not optimal. However, if it is the optimal solution, the distance cost will only be shorter, better reflecting the effect of robust optimization.

6. Conclusions and Future Work

A distributed robust optimization method is proposed to solve the probability uncertainty problem of customer demand service. The method uses the distributed robust optimization theory to transform the uncertain model into a deterministic robust optimization model with parameters. At the same time, the model is optimized by using the data-driven kernel density estimation method. We also develop a hyper-heuristic algorithm for solving the problem.

Compared with other algorithms, our hyper-heuristic algorithm based on reinforcement learning has good performance. In solving UVRP, the data-driven distributed robust optimization model produces much better results than the deterministic model, and our method can effectively solve the problem and reduce the additional cost.

This paper studies VRPUC, but in real world, there are more uncertain factors, such as vehicle travel time and customer demand. It is necessary to consider the joint influence of multiple uncertain factors in future work. The hyper-heuristic underlying operators in this paper are common and relatively general. It is also meaningful to design some specific underlying algorithms to solve such problems in the future.

Data Availability

All calculation instances (<https://www.bernabe.dorrnsoro.es/vrp/>) in set A in the CVRP are selected for the experiment. All the data used to support the findings of this study are also available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The study was supported by the National Natural Science Foundation, China (no. 61402409), the Natural Science Foundation of Zhejiang Province China (no. LY19F030017), and also supported by the Foundation of State Key Laboratory of Digital Manufacturing Equipment and Technology (Grant no. DMETKF2022024).

References

- [1] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.
- [2] H. Lei, G. Laporte, and B. Guo, "Districting for routing with stochastic customers," *EURO Journal on Transportation and Logistics*, vol. 1, no. 1–2, pp. 67–85, 2012.
- [3] S. S. Özarık, L. P. Veelenturf, T. V. Woensel, and G. Laporte, "Optimizing e-commerce last-mile vehicle routing and scheduling under uncertain customer presence," *Transportation Research Part E: Logistics and Transportation Review*, vol. 148, Article ID 102263, 2021.
- [4] M. W. Ulmer, "Anticipation versus reactive re-optimization for dynamic vehicle routing with stochastic requests," *Networks*, vol. 73, no. 3, pp. 277–291, 2019.
- [5] D. Markovic, G. Petrovic, Z. Cojbasic, and D. Marinkovic, "A metaheuristic approach to the waste collection vehicle routing problem with stochastic demands and travel times," *Acta Polytech Hung*, vol. 16, pp. 45–60, 2019.
- [6] L. Hou, H. Zhou, and C. Liang, "Vehicle routing problem with uncertain demand and travel time," *Computer Integrated Manufacturing Systems*, vol. 17, no. 1, pp. 101–108, 2011.
- [7] X. Ge, Z. Zhu, and Y. Jin, "Electric vehicle routing problems with stochastic demands and dynamic remedial measures," *Mathematical Problems in Engineering*, pp. 1–15, 2020.
- [8] S. Souyris, C. E. Cortés, F. Ordóñez, and A. Weintraub, "A robust optimization approach to dispatching technicians under stochastic service times," *Optics Letters*, vol. 7, no. 7, pp. 1549–1568, 2013.
- [9] E. Solano-Charris, C. Prins, and A. C. Santos, "Local search based metaheuristics for the robust vehicle routing problem with discrete scenarios," *Applied Soft Computing*, vol. 32, pp. 518–531, 2015.
- [10] Z. H. Hu, J. B. Sheu, L. Zhao, and C. C. Lu, "A dynamic closed-loop vehicle routing problem with uncertainty and incompatible goods," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 273–297, 2015.
- [11] X. Ge, Z. Zhu, and Y. Jin, "Electric vehicle routing problems with stochastic demands and dynamic remedial measures," *Mathematical Problems in Engineering*, vol. 2020, Article ID 8795284, 15 pages, 2020.
- [12] B. F. Moghaddam, R. Ruiz, and S. J. Sadjadi, "Vehicle routing problem with uncertain demands: an advanced particle

- swarm algorithm,” *Computers & Industrial Engineering*, vol. 62, no. 1, pp. 306–317, 2012.
- [13] Y. Ye, Y. Liu, and X. Chen, “Energy management optimization model of island microgrid with uncertain generation and load,” *Water Resources and Power*, vol. 38, no. 1, pp. 191–195, 2020.
- [14] Y. Zhou, M. Shahidehpour, Z. Wei, Z. Li, G. Sun, and S. Chen, “Distributionally robust unit commitment in coordinated electricity and district heating networks,” *IEEE Transactions on Power Systems*, vol. 35, no. 3, pp. 2155–2166, 2020.
- [15] H. Chu, “Distributionally robust optimization with applications in healthcare service operations management,” *Beijing Institute of Technology*, vol. 2016, pp. 53–76, 2016.
- [16] K. S. Shehadeh, A. E. M. Cohn, and R. Jiang, “A distributionally robust optimization approach for outpatient colonoscopy scheduling,” *European Journal of Operational Research*, vol. 283, no. 2, pp. 549–561, 2020.
- [17] C. Shang, S. X. Ding, and H. Ye, “Distributionally robust fault detection design and assessment for dynamical systems,” *Automatica*, vol. 125, Article ID 109434, 2021.
- [18] J. Zhang, Q. Feng, Y. Zhao, J. Liu, and L. Leng, “Hyper-heuristic for cvrp with reinforcement learning,” *Computer Integrated Manufacturing Systems*, vol. 26, no. 4, pp. 1118–1129, 2020.
- [19] W. Qin, Z. Zhuang, Z. Huang, and H. Huang, “A novel reinforcement learning-based hyper-heuristic for heterogeneous vehicle routing problem,” *Computers & Industrial Engineering*, vol. 156, Article ID 107252, 2021.
- [20] P. Mohajerin Esfahani and D. Kuhn, “Data-driven distributionally robust optimization using the Wasserstein metric: performance guarantees and tractable reformulations,” *Mathematical Programming*, vol. 171, no. 1–2, pp. 115–166, 2018.
- [21] Y. Song, L. Wan, and Y. Liu, “Coarse-to-fine detection for nests on pylon,” *Power Syst Clean Energy*, vol. 32, no. 6, pp. 85–88, 2016.
- [22] R. S. Sutton and A. G. Barto, “Reinforcement learning: an introduction,” *IEEE Transactions on Neural Networks*, vol. 9, no. 5, p. 1054, 1998.
- [23] P. Rakshit, A. Konar, P. Bhowmik et al., “Realization of an adaptive memetic algorithm using differential evolution and Q-learning: a case study in multirobot path planning,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 4, pp. 814–831, 2013.
- [24] S. Akpinar, “Hybrid large neighbourhood search algorithm for capacitated vehicle routing problem,” *Expert Systems with Applications*, vol. 61, pp. 28–38, 2016.
- [25] Y. Li and H. Fan, “Hybrid variable neighborhood biotic search algorithm for vehicle routing problem with capacity constraints,” *Control and Decision*, vol. 33, no. 7, pp. 1190–1198, 2018.
- [26] N. Lin, Y. Shi, T. Zhang, and X. Wang, “An effective order-aware hybrid genetic algorithm for capacitated vehicle routing problems in internet of things,” *IEEE Access*, vol. 7, Article ID 86114, 2019.