Research Article

A Biobjective Vehicle Routing Problem with Stochastic Demand and Split Deliveries

Yachao Wu,1,2 Min Zhou,1 Dezhi Zhang3,4 and Shuangyan Li4

1School of Economics and Management, China University of Mining and Technology, Xuzhou 221116, China
2School of Management Engineering, Huaiyin Institute of Technology, Huai’an 223001, China
3School of Traffic & Transportation Engineering, Central South University, Changsha, Hunan 410075, China
4College of Logistics and Transportation, Central South University of Forestry and Technology, Changsha, Hunan 410004, China

Correspondence should be addressed to Dezhi Zhang; dzzhang@csu.edu.cn

Received 30 November 2021; Revised 16 May 2022; Accepted 31 May 2022; Published 13 July 2022

1. Introduction

The vehicle routing problem (VRP) is one of the most important elements of many logistic systems, which was introduced by Dantzig and Ramser [1] more than 50 years ago. The VRP is rooted in a wide variety of real-world applications and has attracted a lot of attention in both academia and industry in the last decades. In a classic VRP, the objective is to seek the optimal set of routes from the depot to customers with different locations for delivering goods or services, where the demands of customers are known and cannot be split. However, this unsplit distribution strategy requires more vehicles and often results in route failure when demand exceeds the residual capacity of the vehicle. Such route failures occur particularly in cases where the customer demand is stochastic and close to the vehicle capacity, which increases the total travel cost. In many real-life situations, customers have stochastic demands that may result in significant route failure [2–4]. Hence, it is important to employ a split strategy to reduce route failure with stochastic demand.

Moreover, if we consider only the minimization of travel costs as an optimization objective, there will be substantial differences on the workloads of different routes, which can impact drivers’ income or benefits. This can be seen as unfair by the drivers. Such unfairness will affect driver’s enthusiasm and satisfaction, reduce the level of distribution services, and reduce employee’s loyalty and distribution efficiency [5, 6].

Currently, employees are a vital resource to companies. Driver’s welfare is a significant issue since drivers exist in an environment of competition among logistics companies [7].

Therefore, it is necessary to investigate the vehicle routing problem with stochastic demand, split deliveries, and route workload balance in real-life situations. In this study, we address a biobjective vehicle routing problem, which simultaneously considers the workload balance of all routes and the total travel costs. The biobjective VRP is a variation of VRP with stochastic demand and split deliveries; therefore, it is also NP-Hard [8].

Stochastic vehicle routing problems have been studied for almost 50 years [9]. Generally, the following three
stochastic situations are included there: stochastic demand, stochastic customers, and stochastic travel time [2]. Bertsimas et al. [10] addressed a priori optimization solution to solve the vehicle routing problem with stochastic demand, where the first-stage solution is constructed for a set of vehicle routes, and the routes are then adjusted randomly. A priori optimization method has been applied in VRPs by Laporte and Louveaux [3], Gendreau et al. [11], and Lei et al. [6].

To deal with the stochastic VRPs, there exist some other possible policies in the literature. For example, Dror et al. [12] proposed a policy in which the vehicle returns to the depot to reload demands after a route failure and then resumes the route at the failure customer. Laporte et al. [13] investigated the problem of multiple route failures with normal and Poisson’s distribution of demands. Bertsimas et al. [14] addressed a preventive recourse strategy, in which the vehicle may preventively return to reload in advance if there exists a route failure. Aykagan and Erera [15] developed a paired-vehicle recourse policy in which some vehicle routes are paired a priori for the purpose of pooling the capacity of two vehicles to reduce total travel costs. They solved the problem using a Tabu search algorithm. Goodson et al. [16] proposed a set of rollout policies based on fixed routes, which uses predecision and postdecision situations to distinguish two additional rollout variants. Zhu et al. [17] formulated a paired cooperative reoptimization (PCR) strategy, which can realize cooperation between a pair of vehicles and can be applied in multivehicle situations. A heuristic was developed to dynamically alter the visiting sequence and the vehicle assignment based on the updated information. Biesinger et al. [18] addressed an integer L-shaped method based on decomposition and branch-and-cut approach to solve the VRP with stochastic demands. Other related studies were conducted by Adulyasak and Jaillet [19], Gounaris et al. [20], and Goodson et al. [21]. For a comprehensive review of the stochastic VRPs, interested readers can refer to Jorge et al. [22].

There exist a number of studies on VRP with split delivery. Dror and Trudeau [23] introduced an integer programming formulation for the VRP [12]. Frizzell and Giffin [24] studied the vehicle routing problem with split deliveries, while considering time window constraints and grid network distances. Dror et al. [25] investigated that more savings can be obtained from split deliveries in the situation where the average customer demand exceeds 10% of the vehicle capacity. McNabb et al. [26] have found that the optimal solution of the VRP with split deliveries is much better than that of VRP with the condition of a limited number of vehicles. Jin et al. [27] investigated the split delivery vehicle routing problem, presenting a two-stage algorithm with valid inequalities. Prins et al. [28] addressed the vehicle routing problem with split delivery and proposed a tour splitting algorithm. Simulation results show that the greedy randomized adaptive search procedure and an iterative local search are competitive. Archetti et al. [29] proposed two exact branch-and-cut algorithms to solve the SDVRP and provided optimal bounds for the problem by relaxing some formulations. Vidal [30] presented several technical notes on split algorithms and provided a simpler alternative algorithm by highlighting a stronger property of the directed acyclic graph. Moshref-Javadi and Lee [31] studied the split delivery vehicle routing problem while considering the customer-centric and multicommodity. Chen et al. [32] addressed the effect of split deliveries on the solution of the open vehicle routing problem, given that each customer’s demand is allowed to be split into two parts at most.

In recent years, vehicle routing problems with route workload balance have gained significant attention from researchers. Workload balance becomes increasingly an important issue for VRP in real-world logistics systems (e.g., milk distribution, express package delivery, waste collection, etc.). Lehuédé et al. [7] proposed an integer programming model on the vehicle routing problem with load balancing, in which the load balancing is the balance of all routes’ lengths. Jozefowiez et al. [33] presented a metaheuristic method involving classical multiobjective operators, and an elitist diversification mechanism is designed to improve its efficiency. Chen et al. [34] developed an improved model of vehicle routing problem with time windows and load balancing (VRPTL), which aims to meet four objectives: the minimum number of vehicles, the minimum total travel cost, a balance of route length, and the customers’ time windows. They also designed a two-stage heuristic algorithm. Mendoza et al. [35] addressed the problem that considered the duration constraint. Two strategies were used to address route-duration constraints in the VRP: one included handling the duration constraints as chance constraints, and the other included penalizing the objective function if there were violations of the duration constraint. Szymon and Dominik [36] applied the parallel Tabu search algorithm with graphic processing units (GPUs) to solve the multicriteria vehicle routing problem. Smilowitz et al. [37] presented an optimization model with workforce management in periodic delivery operations, aiming to achieve a balance between workforce management and total travel. Oyola and Løkketangen [38] addressed the VRP with work balancing to minimize the difference between individual route lengths as well as the total travel cost. Santiago et al. [39] mapped a biobjective mixed-integer linear model to the problem of how to arrange daily routes to minimize total travel cost and balance the workload of drivers within the required time windows. A real case study was conducted based on a company in Tenerife. Koulaeian et al. [40] investigated the multidepot VRP with simultaneous pickup and delivery, and the balance in the allocation of routes to drivers is considered. Meanwhile, a metaheuristic algorithm embedded with a genetic algorithm and an imperialist competitive algorithm is proposed. Other examples of studies addressing problems concerning the balance of routes can be found in Carlsson et al. [41] and Oyola and Løkketangen [38]. Recently, Haniyeh and Rahmati [42] addressed the algorithms for the load-balanced capacitated vehicle routing problem. Mancini et al. [43] examined the collaborative vehicle routing problem with time and service consistency and workload balance. They found that workload balancing will benefit in collaborating among carriers.
almost without additional cost. Matl et al. [44] investigated the workload equity in vehicle routing and formulated a balance criterion. They revealed the importance of selecting the right workload resources. Matl et al. [45] also published recent surveys of vehicle routing problems with workload balance and identified future directions.

Based on the above analysis, we can see that the split delivery strategy has been proved to be an effective method to save transportation costs by improving vehicle loading ratio [23–25]. This method can also reduce route failure when considering stochastic demand [2, 46]. Moreover, considering the workload balance of routes can make scheduling more fair and secure, which benefits drivers. On the other hand, when jointly considering stochastic demand and route balance, special resource strategies and split delivery methods should be applied to reduce route failure and to allocate demands to different vehicles to balance the workload of routes. These requirements become more challenging in optimization modelling and the corresponding solutions. However, to the best of our knowledge, studies on the VRP that consider stochastic demand, split deliveries, and workload balance of routes remain scarce. Therefore, this study aims to fill this gap by focusing on a biobjective vehicle routing problem with stochastic demand and split deliveries, which aims to minimize the total travel cost as well as the balance of route workload (i.e., the difference among the workloads of the routes).

The main contributions of this study are summarized as follows.

(1) An optimization model for biobjective vehicle routing problems is presented. It considers stochastic demand, split deliveries, and the balance of the workloads of all routes.

(2) An improved adaptive large neighborhood search algorithm embedded with the improved weighted sum method is designed to solve the proposed problem, which is verified by some modified Solomon’s instances.

(3) Some managerial insights on the arrangement of distribution routes are revealed, which are based on the analysis of the experimental results.

The remainder of this paper is organized as follows: In Section 2, we provide the formulation of the problem, some definitions and notation, and some algorithm principles. An adaptive large neighborhood search heuristic based on a set of insertion and removal operators is described in Section 3, whereas experimental examples based on modified standard Solomon’s instances are given in Section 4. Finally, conclusions are discussed in Section 5, together with recommendations for further studies.

2. Problem Formulation and Definitions

2.1. Problem Description. In this study, we consider a vehicle routing problem with stochastic demand and split deliveries, which aims to minimize total travel costs and standard deviation of all routes’ workloads. Here, the workload of a route is its duration, including the tour driving time and service time occurred at each customer node. To solve the proposed problem, we designed a recourse policy for paired vehicles. In this policy, we suppose that one customer can be split into at most two paired routes [6, 12, 47].

The policy allows the distribution service to fail once and meet part of the demand first. Each customer is assigned to a vehicle or two for service by a paired route strategy, which considers the balance of the lengths of routes simultaneously. Different vehicles can be paired, while each vehicle can be matched once at most. In the paired route, customers can be served by one or two vehicles. According to the number of vehicles, customers can be classified into two groups. The first group is the general customers served by only one vehicle, while the other is the split customers served by two vehicles.

For the above two different group customers, the corresponding resource policies are adopted as follows. When a failure takes place at the nodes of general customers, we apply the traditional recourse policy. We define two types of failure at the customer nodes as follows: (1) Class I failure, the failure that occurs when the remaining load of the vehicle is less than the demands of the customer. In this case, the vehicle delivers its load, returns to the depot to reload, and then returns to the customer to meet the remaining demands. (2) Class II failure, the failure that occurs when the remaining load of the vehicle is equal to the demands of the customer. In this case, the vehicle delivers its load, returns to the depot to reload, and then returns to the next customer.

When a failure occurs among the split customers, the planned percentage of demands assigned to two vehicles is preallocated in the first stage. Thus, the split customer can be seen as two general customers and use the same recourse policy.

2.2. Model Assumptions. To facilitate the presentation of the essential ideas without loss of generality, several underlying assumptions are given as follows.

(1) The demand of customer \( i \), \( d_i \), is an independent discrete random variable whose distribution is known.

(2) Each customer’s demand \( d_i \) never exceeds the capacity of the vehicle (i.e., \( P[d_i \leq C] = 1 (i \in V/\{0\}) \)).

(3) The distance between any two customers is fixed and known, and \( S_{ij} = S_{ji} \).

(4) The distance between the various customers is consistent with the triangular inequality (i.e., \( S_{ik} + S_{kj} > S_{ij} \)).

(5) All distribution vehicles start from the depot, complete the distribution task, and then return to the same depots.

(6) The demands of all customers must be met. However, the demand of a customer can be split into at most two paired routes. And any route can have only one split customer [2, 6, 24].
(7) No planned routes can fail more than once. That is to say, the cumulative customer demand on any route is less than twice the vehicle capacity,
\[ P\{\sum_{i} d_i \leq 2C\} \equiv 1. \]
When the route is a paired route, the demands of the split customers should be the planned percentage of the vertexes on the corresponding paired routes, denoted as \( a_{ij}^m d_i \), while the vehicle in its paired route services the rest of the customers becomes \((1 - a_{ij}^m) d_i\), where \(0 < a_{ij}^m < 1\) [47].

2.3. Mathematical Model

Notation:

\( V = \{0, 1, 2 \ldots n\} \): Set of nodes, and 0 represents the depot; 1, 2, 3, \ldots, \( n \) is the nodes of customers

\( V_1 \): Set of the unsplit customers

\( V_2 \): Set of the split customers

\( S \subseteq V \setminus \{0\} \): Set of all customers

\( M = \{1, 2, 3, \ldots, m\} \): Set of vehicles

\( k = |M| \): Total number of servicing vehicles

\( S_{ij} \): Distance between customers \( i \) and \( j \); this was used to replace the cost between \( i \) and \( j \)

\( d_i \): Demand of customer \( i \), random variables

\( C_m \): Capacity of vehicle \( m \)

\( r_m \): Routes of vehicle \( m \)

\( \mu \): The probability threshold that ensures that each route fails at most once

\( y_{im} \): The amount of the demand served by vehicle \( m \) for customer \( i \)

\( x_{ijm} = \begin{cases} 1 & \text{if edge } (v_i, v_j) \text{ is traveled by vehicle } m \\ 0 & \text{otherwise}, \end{cases} \)

\( \phi(x) \): The solution vector

\( D_{ij}^m \): The cumulative demand when vehicle \( m \) gets to the customer vertex \( i \)

\( P_D(t) = P(D = t) \): The probability function of the discrete random variable

\( w_m \): The workload of vehicle \( m \)

\( \bar{w} \): The average workload of all routes

\( \sigma \): The variance of the workload of all routes

\( t_{mi} \): The time serviced customer \( i \) by vehicle \( m \)

\( v_m \): The average speed of vehicle \( m \)

\( f_m \): The unit fixed cost of vehicle \( m \) each day

\( c_m \): The unit transport cost of vehicle \( m \)

\( \alpha_1, \alpha_2 \): The weights of the optimal objective 1 and objective 2, respectively

\( \varphi(R) \): The deterministic travel cost of solution \( R \)

\( \psi(m) \): The recourse cost of vehicle \( m \)

\( o \): Operator, including removal operator and insertion operator.

\[
\begin{aligned}
\phi (1) &= \min \{ \varphi (R) + E (\psi (m)) \} \\
&= \min \left\{ \sum_{m \in M} \sum_{i \in V} c_m S_{ij} x_{ijm} + f_m K + c_m E (\psi (m)) \right\}.
\end{aligned}
\]

Equation (1) is the objective function, minimizing the total expected costs, which includes two parts: the deterministic cost of the planned routes and the expected recourse cost. The former includes the traveling cost and fixed cost of vehicles.

\[
\phi (2) = \min \sigma.
\]

Equation (2) means minimizing the variance of the workloads of all routes. We use the variance to measure the balancing of the workload of all routes in this study, subject to the following constraints.

\[
\begin{aligned}
1 &\leq \sum_{m \in M} \sum_{i \in V} x_{ijm} \leq 2, (\forall j \in V), \\
\sum_{j \in V} x_{ijm} &= \sum_{j \in V} x_{jim}, (\forall i \in V, \forall m \in M), \\
\sum_{j \in V} x_{ijm} &= \sum_{i \in V} x_{jim}, (\forall m \in M), \\
\sum_{j \in S} x_{ijm} &\leq |S| - 1, (\forall m \in M), \\
P \left\{ \sum_{i \in V} y_{im} \sum_{j \in V} x_{ijm} \leq 2C_m \right\} &< \mu, (\forall m \in M), \\
y_{im} &\leq d_i^m x_{ijm}, (\forall i \in V, \forall m \in M), \\
\sum_{m \in M} y_{im} &\leq d_i^m, (\forall i \in V), \\
x_{ijm} &\in \{0, 1\}, (\forall i \in V, \forall j \in V, \forall m \in M), \\
R &\left( x_{ijm} \right), (\forall i, j \in V, \forall m \in M), \\
w_m &= \frac{\sum_{i \in V} \sum_{j \in V} S_{ij} x_{ijm} + E (\psi (m))}{v_m} \\
&+ \sum_{i \in V} \sum_{j \in V} t_{mi} x_{ijm}, (\forall m \in M), \\
\bar{w} &= \frac{1}{K} \sum_{i=1}^{K} w_i, \\
\sigma &= \sqrt{\frac{1}{K - 1} \sum_{i=1}^{K} (w_i - \bar{w})^2}.
\end{aligned}
\]
Equations (3)–(4) ensure that each customer is visited by at most two vehicles. Equation (5) indicates that the number of vehicles departing the depot is equal to the number of vehicles returning to the depot, and that all vehicles need to start from the depot and return to the depot. Equation (6) states that there are no subcircuits in the routes. Equation (7) represents a probability threshold constraint that ensures that each route fails at most once based on the probability threshold \( \mu \). Equation (8) implies that a vehicle can service a customer only when the vehicle goes through it. In addition, the demand for services does not exceed the total demands of the customer. Equation (9) ensures the demand of each customer is met. Equation (10) indicates the decision variables are binary variables. Equation (11) represents the decision variable, the probability that the path fails at most once based on the probability threshold \( \mu \). Equation (12) represents the decision variable, the probability that the path fails at the point of the customer is calculated as follows.

\[
\varphi_i = \left\{ \begin{array}{ll}
\sum_{l=1}^{C} P_{d_{m}} (l) P_{D^m_{l+1}} (C - 1), & (i \in V_1) \\
\sum_{l=1}^{C} P_{d_{m}} (l) P_{D^m_{l+1}} (C - 1), & (i \in V_2)
\end{array} \right.
\]

(19)

(3) Using \( r_i^m \) to represent the first type of failure and \( \varphi_i^m \) to represent the second type of failure, the probability of failure is calculated as follows:

\[
\varphi_i^m = \left\{ \begin{array}{ll}
\sum_{l=1}^{C} P_{d_{m}} (l) P_{D^m_{l+1}} (c - 1), & (i \in V_1) \\
\sum_{l=1}^{C} P_{d_{m}} (l) P_{D^m_{l+1}} (c - 1), & (i \in V_2)
\end{array} \right.
\]

(20)

\[
\tau_i^m = \sum_{l=0}^{C-1} P_{D^m_{l+1}} (t) - \sum_{l=0}^{C-1} P_{D^m_{l+1}} (t) - \varphi_i^m.
\]

(21)

Using (15) to (21), \( E[\psi (r_m)] \) can be easily computed as follows:

\[
E[\psi (r_m)] = \sum_{i=2}^{n} (\tau_i^m S_{i} + \varphi_i^m S_{i+1}).
\]

(22)

The total expected cost of a solution \( R \) is then

\[
E[\psi (R)] = \sum_{k=1}^{K} E[\psi (r_k)],
\]

(23)

where \( K \) is the total number of all routes.

2.4. The Expectation of Recourse Cost. We divided failures into two classes, which are defined in Section 2.1. If a failure occurs when vehicle \( m \) visits a customer \( i \), the cost of the recourse routes can be calculated according to the two failure types as follows.

When the failure belongs to a class I:

\[
S_{i}^{m} = 2S_{i0m}.
\]

(15)

When the failure belongs to a class II:

\[
S_{i0m}^{m} = S_{i0m} + S_{i+10m} - S_{i+10m}.
\]

(16)

The calculation of service failure probability is shown as follows.

We only consider the situation where the customer demands are discrete random variables. We define it as the probability function of the discrete random variable \( D \) that is equal to \( t \); then \( P_{d_{m}} \) is a known value.

(1) When the vehicle \( m \) visits customer \( i \), the probability of the cumulative demand that reaches \( D_{i}^{m} \) is \( P_{D_{i}^{m}} (t) \); it can be calculated as follows:

\[
P_{D_{i}^{m}} (t) = \left\{ \begin{array}{ll}
\sum_{l=0}^{t} P_{D_{i+1}^{m}} (t - l) P_{d_{m}} (l), & (i \in V_1) \\
\sum_{l=0}^{t} P_{D_{i+1}^{m}} (t - l) P_{d_{m}} (l), & (i \in V_2)
\end{array} \right.
\]

(17)

The boundary condition is shown as follows.

\[
P_{D_{i}^{m}} (t) = \left\{ \begin{array}{ll}
P_{d_{m}} (l), & (i \in V_1) \\
P_{d_{m}} (l), & (i \in V_2)
\end{array} \right.
\]

(18)

Here, \( a_{i}^{m} \) is the split percentage. It is in the range of 0 to 1.
total expected costs of \( R_1 \) and \( R_2 \), while let \( f^{1\,f}_{(1)}(R_1) \) and \( f^{1\,f}_{(1)}(R_1) \) be the corresponding variance of the lengths of all routes.

(1) For the total expected costs, if \( f^{1\,f}_{(1)}(R_1) > f^{1\,f}_{(1)}(R_2) \), we regard solution \( R_2 \) as better than \( R_1 \) and we define \( g_1 \) as follows.

\[
g_1 = \begin{cases} 
\frac{f^{1\,f}_{(1)}(R_1) - f^{1\,f}_{(2)}(R_2)}{f^{1\,f}_{(1)}(R_1)} & f^{1\,f}_{(1)}(R_1) \neq 0 \\
0 & f^{1\,f}_{(1)}(R_1) = 0 \end{cases} \tag{24}
\]

If \( g_1 < 0 \) (i.e., \( f^{1\,f}_{(1)}(R_1) > f^{1\,f}_{(1)}(R_2) \)), we regard the solution \( R_2 \) as better than \( R_1 \).

If \( g_1 > 0 \) (i.e., \( f^{1\,f}_{(1)}(R_1) < f^{1\,f}_{(1)}(R_2) \)), we regard the solution \( R_1 \) as better than \( R_2 \).

(2) For the second objective (the variance of the workloads of all routes), when \( f^{1\,f}_{(2)}(R_1) > f^{1\,f}_{(2)}(R_2) \), we regard the solution \( R_2 \) as better than \( R_1 \), defining \( g_2 \) as follows.

\[
g_2 = \begin{cases} 
\frac{f^{1\,f}_{(2)}(R_1) - f^{1\,f}_{(2)}(R_2)}{f^{1\,f}_{(2)}(R_1)} & f^{1\,f}_{(2)}(R_1) \neq 0 \\
0 & f^{1\,f}_{(2)}(R_1) = 0 \end{cases} \tag{25}
\]

If \( g_2 < 0 \), that is to say, \( f^{1\,f}_{(2)}(R_1) > f^{1\,f}_{(2)}(R_2) \), we regard the solution \( R_2 \) as better than \( R_1 \).

If \( g_2 > 0 \), that is to say \( f^{1\,f}_{(2)}(R_1) < f^{1\,f}_{(2)}(R_2) \), we regard the solution \( R_1 \) as better than \( R_2 \).

(3) We define the heuristic function as follows.

\[
h = \alpha_1 \cdot g_1 + \alpha_2 \cdot g_2, \tag{26}
\]

where \( \alpha_1, \alpha_2 \) are the weights of the two objectives. The difference in weight represents the degree of emphasis of the two objectives. Note that \( \alpha_1 \) and \( \alpha_2 \) fall into the interval [0, 1] and there exists \( \alpha_1 + \alpha_2 = 1 \).

In the iterative process of the algorithm, if \( g_1 < 0, g_2 < 0 \), the new solution is clearly better than the previous solution, so accept the new solution; if the new solution is not accepted; if \( g_1 < 0, g_2 > 0 \), or \( g_1 > 0, g_2 < 0 \), whether to accept the new solution is based on the acceptance criteria of the heuristic and the heuristic function.

To enrich the searching space of the solution algorithm, the algorithm designed in this study accepts the infeasible solutions by a certain probability. If the value of \( \alpha_1 \) is small, the first optimization objective, namely, the total expected cost, will be prior to the second one; while if the value of \( \alpha_1 \) is large, the second objective, the variance of the workloads of all routes, will be prior to the first one.

### 3. An Adaptive Large Neighborhood Search Heuristic Algorithm

As a variant of the VRP, the proposed problem is also a NP-hard, which poses great computational challenge for large-size instances by exact solution algorithms [6, 8, 15, 42]. In this regard, a heuristic method is proved to be an effective method [22, 42]. Therefore, we employed an adaptive large neighborhood search (ALNS) heuristic algorithm to deal with the above proposed optimization problem. The ALNS heuristic was first introduced by Ropke and Pisinger [48]. Demir et al. [49] modified some of the existing removal operators and insertion algorithms, as well as designed some new removal and insertion operators to solve their specific problems. In addition, they also introduced a scoring mechanism for self-adjustment of some removal/insertion mechanisms and produced multiple initial solutions during the search. The heuristic algorithm was recently applied by Grangier et al. [50] to solve the two-echelon multiple-trip VRP with satellite synchronization and by Luo et al. [46] to the VRP with stochastic demand and weight-related cost.

The ALNS heuristic is currently widely used, which can jump out of a local optimal solution to a certain extent and thus accelerate the process of finding a global optimal solution. In the iterative process of our algorithm, customer vertices are removed and reinserted by means of selected removal operators and insertion operators [13]. The removal operator and insertion operator are selected by a probabilistic mechanism. Among these operators, the total worst cost travel removal, route balance removal, and insertion are new. Other operators are inspired by previous studies [49]. In view of the requirements of this paper, we redesigned those operators. At the end of the iteration, a feasible temporary solution is obtained that can be discarded or make the new feasible solution.

Our improved ALNS heuristic algorithm is described as follows and the framework is shown in Figure 1.

#### Step 1.
Set the initial parameters and construct the initial solution. Then, define the initial solution as the current solution and the global optimal solution.

#### Step 2.
Select the remove operators and the insertion operator through the operator selection mechanism to construct a new solution where the new solution satisfies the threshold requirement in equation (6).

#### Step 3.
Based on the acceptance criteria of the algorithm, update the current solution and the global optimal solution.

#### Step 4.
Update the scores and the usage frequency of the operators. Every 50 iterations, according to the scoring criteria and solution performance of operators, update the operator score and frequency.

#### Step 5.
Output the global optimal solution if the stopping criterion is satisfied; otherwise, return to Step 2.
For the convenience of the following description, the following notations are defined:

Iter: the current iteration step number
Max-iter: the largest iteration step number
Cons-iter: the continuous iteration step number that the current best solution maintains unchanged
Max-cons-iter: the largest iteration step number that the current best solution maintains unchanged

Stopping criterion: The number of iterations on the current global optimal solution remains constant, reaches the given max-cons-iter, or the total number of iterations reaches max-iter. Then, the search procedure stops.

3.1. The Construction of the Initial Solution. We were inspired by the means of the construction heuristic, which includes split deliveries and route pairing as shown by Shaw [51],
when considering constructing an initial solution. We devised this construction heuristic to generate an initial solution in this paper. The heuristic first sorts the expected demands in ascending order. When the vertices are sorted, starting from the left of the sequence, sequentially select the cheapest insertion overall route \( m \). We use \( I_h(i,r_m) \) to denote the cost of vertex \( h \) in the position \( i \) of route \( m \). To determine the initial solution efficiently, the insertion cost includes only the deterministic part of the route and excludes the recourse cost. Thus, the insertion cost can be calculated as follows:

\[
I_h(i,r_m) = s(v_{i-1}^m,v_h^m) + s(v_h^m,v_i^m) - s(v_{i-1}^m,v_i^m).
\]  

(27)

The insertion process must always meet the requirements of the probability threshold in constraint (6). If there is no feasible insertion for the current vertex, we split customer demand and search the routes in the current solution for the first suitable single route among the unpaired routes, where a fraction of the customer demand can be inserted into the position with the smallest insertion cost. The inserted fraction is the largest value that can be inserted into the route with the premise of meeting constraints (6). To simplify the problem, we consider a limited number of values of this percentage with an initial value of 0.9 with a stepwise reduction of 0.1. If a first feasible single route exists, the remaining part of the customer demand will be inserted into a second unpaired route, which will then be paired with the first route, while a new path is generated if the second route does not exist. We can determine a feasible initial solution quickly by the above method.

3.2. Removal Operators. To efficiently explore the solution space, four removal operators were developed in our proposed ALNS framework.

3.2.1. Workload Balance Worst Removal. The workload balance worst removal operator concentrates on the vertex customers that have a relatively great impact on route workload balance. During this operation of the algorithm, we calculate the variance change of the workload of all routes after the removal of a customer vertex, and select the customer vertices with the largest variance change to remove from current solution routes until the number of removed vertices reaches the expected number of removal. The variance change of the route can be calculated as follows:

\[
\Delta \sigma = \sigma(R) - \sigma_{-q}(R).
\]  

(28)

3.2.2. Random Removal. The random removal operators adopted from Pisinger and Ropke [52], Ropke and Pisinger [48], and Demir et al. [49] are used in our proposed ALNS framework. This operator randomly selects customer vertices and removes them from the corresponding routes. Although the random removal operator has no direct relationship with the path cost and line equalization, it can diversify the search space and make the algorithm jump out of the local search area in some situations.

3.2.3. Deterministic Cost Worst Removal. The purpose of the deterministic cost worst removal operator is to remove customer vertices that appear to be inserted in the wrong position in the current solution. It was inspired by Ropke and Pisinger [48], Laporte, et al. [13], and Lei, et al. [6]. In this study, we consider only the deterministic cost part of the solution. Denote \( \phi(r_m) \) as the deterministic cost of route \( r_m \) and \( \phi_{-i}(r_m) \) as the deterministic cost of route \( m \) without customer \( i \). The cost change is calculated as follows:

\[
\Delta \phi(i,m) = \phi(r_m) - \phi_{-i}(r_m).
\]  

(29)

The algorithm first computes the cost changes associated with the removal of a customer from each route and finds the customer vertices with the largest cost changes in each route. The next step is to sort the routes in decreasing order of their largest cost changes. The following two cases should be considered. One case is where the number of routes is larger than \( q \), and the algorithm sorts the routes in descending order based on the largest customer vertex change. It then removes the customer vertex from the first \( q \) routes. The other case is where the number of routes is smaller than \( q \), and the algorithm iteratively removes the vertex with the largest cost change in each route until the number removed reaches \( q \).

3.2.4. The Recourse Cost Worst Removal. The recourse cost worst removal operator is similar to the deterministic cost worst removal operator. It focuses on removing vertices of the routes in the current solution that appear at the wrong location. However, it concentrates on only the expected recourse of the customer vertices in each route. The algorithm first computes the recourse cost changes associated with the removal of one customer vertex from each route by equation (30) and then sorts the routes in decreasing order of their largest expected recourse cost changes. It then considers two cases: (1) If \( q \) is larger than the number of routes, the operator removes the customer vertex with the largest expected recourse cost changes of the current solution until the number of removed vertex reaches \( q \). (2) Otherwise if \( q \) is smaller than the number of routes, the operator removes the vertex with the largest expected recourse cost change from the first \( q \) routes.

\[
\Delta E[\psi(R)] = E[\psi(R)] - E[\psi_{-q}(R)].
\]  

(30)

3.3. Insert Operators

3.3.1. Workload Balance Best Insertion. The workload balance best insertion operator is designed for only route workload balance. The algorithm loosely inserts each removed vertex into the route, until all the removed customer vertices are inserted into the routes. During the insertion process, constraint (7) is always given priority. For each insertion, the algorithm first computes the variation in possible insertion points of each route based on equation (31) and then selects the position, in which the change of the
variance of workload is maximum. This insertion procedure is repeated until all the vertices are inserted into the route.

\[
\Delta \sigma = \sigma(R) - \sigma_{e_1}(R).
\]  

3.3.2. Greedy Insertion. This operator is a simple construction heuristic that inserts each removed customer vertex into the best feasible position within the routes. At each step, the greedy insertion operator selects a removed customer vertex that has the lowest incremental insertion cost among all those evaluated for insertion at all possible positions in all the routes. Here, for simplicity, we consider only the deterministic travel cost when we calculate the incremental insertion cost. This insertion procedure is repeated until all removed customer vertices have been inserted.

3.3.3. Split Insertion. The split insertion operator splits the customer’s demands and allocates them among two unpaired routes. The algorithm first enumerates all possible pairs of unpaired routes. In the paired routes, one route serves a certain percentage of the demand, and the other route serves the remaining demand without violating constraint (7). Note that we define the percentage of service requirements as follows [47].

\[
\beta_1 = \frac{n_1}{\sum_{i=1}^{n_1} E(d_i^1) + \sum_{i=1}^{n_2} E(d_i^2)}
\]

\[
\beta_2 = 1 - \beta_1,
\]

where \(n_1\) is the number of customer vertices on the first route and \(n_2\) is the number of the customer vertices on the other route. What we should pay special attention to is that the split in Section 3.1 is not considered.

To determine the best insertion position for the splitting customer in each possible route, the operator computes the increased cost by reinserting the customer vertex into the entire possible paired route. After that, the operator inserts the percentage of the demand into the position that has the smallest increase in the total travel cost and marks the two vehicle routes as paired routes. If no two such routes exist, a new route is generated for the customer vertex. We then insert it into the route without a split.

3.3.4. Demand and Failure Sorting Insertion. This operator first sorts the removed customer vertices in order of the expected demand from large to small. The routes in the solution are then sorted in the order of their service failure probability from small to large as shown by Laporte et al. [13]. With iteration, the algorithm selects the customer vertex with the maximum expected demand from the removed vertices collection, inserting it into the first route without violating assumption 8. The optimal insertion position in the route is the location where the minimum total cost increment is generated. A new route is created when a vertex cannot be inserted into an existing route. After each insertion, the service failure probability for each path is recalculated.

3.4. Operators’ Selection Mechanism. There are four removal operators and four insertion operators in this paper. We divide the entire search procedure of \(N\) iterations into \(N\) “segments” and define \(N_s = \lceil N/A \rceil\) to represent the number of sequential iterations in a segment. Every operator \(o\) is associated with a score \(y_o\), which shows how well an operator has performed during this segment. The score \(y_o\) is reset to zero at the beginning of each segment. The score \(y_o\) is calculated as follows:

\[
y_o = y_o + \begin{cases} 6 & \text{A feasible solution,} \\ 10 & \text{A better solution,} \\ 30 & \text{Global optimal solution.} \end{cases}
\]

We denote the adaptive weight of operator \(o\) by \(\rho_{os}\) during the segment \(s\) and use \(\beta_{os}\) to represent the number of times operator \(o\), which has been selected during segment \(s\). When \(s = 1\), then \(\rho_{os} = 1\); that is to say, all the operators have the same weight initially. After \(N_s\) iterations, that is, at the beginning of segments \(s + 1\), the value of \(\rho_{s+1}\) is updated for each operator \(o\) according to their performance in the previous segments as follows:

\[
\rho_{o(s+1)} = (1 - \kappa)\rho_{os} + \frac{\kappa y_{ij}}{\beta_{os}}
\]

where \(\kappa \in [0, 1]\) is a parameter to control the speed of the adaptive weight adjustment mechanism as it reacts to changes in the effectiveness of the algorithms. In our simulation experiments, the value of \(\kappa\) is set to 0.1 as in Emeç et al.’s work [53].

3.5. Solution Acceptance. The acceptance of a new solution is based on Dueck’s record-to-record travel (RRT) [54]. Inspired by the RRT criterion, we improved the acceptance and designed the solution acceptance criterion as follows.

After several iterations (suppose \(i\) times), use \(R^*\) to denote the current global optimal solution. The corresponding goal values are \(f^*_1\) and \(f^*_2\). Using \(R^*_i\) to denote the current solution, the corresponding goal value are \(f^*_1\) and \(f^*_2\). In the next iteration \((i + 1)\) times, the solution is denoted as \(R^*_{i+1}\), with corresponding goal values \(f^*_{i+1}\) and \(f^*_{i+1}\).

If \(\alpha_1 f^*_{i+1} - f^*_1/f^*_1 + \alpha_2 f^*_{i+1} - f^*_2/f^*_2 < 0.1\), the solution \(R^*_{i+1}\) produced in the \((i + 1)\)th iteration is an acceptable solution, so we update the current solution. It accepts a certain degree of bad solutions, which serves to enrich the source of the solution.

If \(\alpha_1 f^*_{i+1} - f^*_1/f^*_1 + \alpha_2 f^*_{i+1} - f^*_2/f^*_2 < 0\), the solution \(R^*_{i+1}\) produced in the \((i + 1)\)th iteration is better than the current solution, so we update the current solution. Continue to judge whether the solution \(R^*_{i+1}\) is better than the current global optimal solution \(R^*\). If \(\alpha_1 f^*_{i+1} - f^*_1/f^*_1 + \alpha_2 f^*_{i+1} - f^*_2/f^*_2 < 0\), then
solution $R_{i+1}$ produced in the $i + 1$ times’ iteration is better than the current global optimal solution $R^*$, so we update the current global optimal solution $R^*$. Note that $\alpha_1 + \alpha_2 = 1$.

3.6. Stopping Criterion. The ALNS algorithm is a heuristic algorithm, so it is difficult to find a clear termination condition. To take into account the effectiveness and speed of the algorithm, we use the following stopping criterion. The algorithm stops when the quality of the solution does not improve within 300 consecutive iterations, or the total number of iterations of the algorithm exceeds 2000.

4. Computational Experiments

The algorithm described in Section 4 was coded using MATLAB and run on a laptop computer with a 4 GHz dual processor and 2 GB RAM. We first describe our experimental design and the corresponding value of the input parameters. Moreover, we have conducted some analysis on the results of simulation experiments. The following computational experiments performed are modified by Solomon’s instance [55].

4.1. Description of the Experiments and Input Parameters. In the following experiments, we assume that all customer demands are discrete distributions that comply with a Poisson distribution. That is to say, the cumulative probability distribution can be computed directly. For instance, if a customer vertex $i$ on the route $m$ then the cumulative demand $D_i^m = \sum_{j=1}^{m-1}d_{ij}^m \sim \text{Poisson} (\mu_i^m)$. For the split customer vertex $i$ on the route $m$, if $d_i^m \sim \text{Poisson}(\mu_i^m)$, then $\alpha_i^m d_i^m \sim \text{Poisson}(\alpha_i^m \mu_i^m)$, and we can deduce that $D_i^m = \sum_{j=1}^{m-1}d_{ij}^m + \alpha_i^m d_i^m \sim \text{Poisson} (\sum_{j=1}^{m-1}d_{ij}^m + \alpha_i^m d_i^m)$. In the modified Solomon instances, the coordinates of the customer vertices keep the same as the Solomon instances, and the demand of each customer is generated based on the Poisson distribution, whose mean value is equal to the corresponding determinate value of vertex requirement in the Solomon instances. Since the capacity of the vehicle in the original instance is too large for our problem, we set the capacity of the vehicle in our case study to 70. There are six sets of problems where the geographical data are randomly generated: R1, R2, RC1, RC2, C1, and C2. As we know, R1 and R2 have uniform distributions, RC1 and RC2 are semiclustered, and C1 and C2 are clustered in groups. Since the time window is not considered in our study, the customer’s service time information and time window are removed from the corresponding cases. Therefore, the instances can be merged into four categories, R, RC, C1, and C2. Every Solomon’s instance has 100 customers, and we consider only the first 25 and 50 customer vertices in our test experiments. The location information of customers distribution is shown as Figure 2.

If the demand of each customer is relatively small compared with the capacity of the vehicle, the splitting strategy shows less impact on the total expected cost. Inspired by Ho and Haugland’s research [56], we have transformed customer demands into the interval $[l_i, u_i]$, and redefined customers’ demands as follows.

$$d_i^m = l + \frac{u - l}{E(d_i) - E(d)} (E(d_i) - E(d))$$

where $E(d_i) = \min\{E(d_i), i \in V\}$ and $E(d) = \max\{E(d_i), i \in V\}$. The values of $l$ and $u$ are set to 0.5 and 0.7, respectively, according to Lei et al. [47].

In addition, Table 1 shows the corresponding value of the initial input parameters of the proposed ALNS, which are
adopted from the existing literature [13, 46, 55] and determined by repeated simulation experiments.

### 4.2. Computational Results and Analysis

We used the proposed ALNS algorithm to solve the above optimal model. To justify the validity of our model and algorithm, several simulation experiments that were adapted from Solomon's instances were conducted. Each instance is replicated 10 times, and the computational results are shown in Figures 3–6.

#### 4.2.1. Computational Results Analysis

As seen from Figures 3–6, with the increase in the weight of the first objective function, there is an increase of the first goal and a decrease of the second goal.

### Table 1: The initial input parameters of the proposed ALNS.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Initial value</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Capacity of each vehicle (unit)</td>
<td>70</td>
</tr>
<tr>
<td>( f_m )</td>
<td>Fixed dispatch cost of vehicle ( m ) (yuan/vehicle)</td>
<td>150</td>
</tr>
<tr>
<td>( c_{mv} )</td>
<td>Driving cost per unit distance of vehicle ( m ) (yuan/km)</td>
<td>2.25</td>
</tr>
<tr>
<td>( v_m )</td>
<td>Speed of vehicle ( m ) during delivery (km/h)</td>
<td>40</td>
</tr>
<tr>
<td>( t_{mi} )</td>
<td>Time required for vehicle ( m ) to serve customer ( i ) (h)</td>
<td>0.25</td>
</tr>
<tr>
<td>Max-iter</td>
<td>Maximum number of iteration steps</td>
<td>2000</td>
</tr>
<tr>
<td>Max-cons-iter</td>
<td>The maximum number of consecutive iteration steps for which the current best solution remains unchanged</td>
<td>50</td>
</tr>
<tr>
<td>( \alpha_i )</td>
<td>Initial weight of neighborhood searching operator ( i )</td>
<td>1</td>
</tr>
</tbody>
</table>
decrease of the second one. In other words, the first goal of total the expected cost decreases, while the second goal of the variance of the workloads of routes increases. Therefore, we can obtain that there exists a negative correlation between the total expected cost and the workload balances of routes.

Figures 3 and 4 show that the total expected cost decreases steadily with the increase of the value of the objective function weight parameter \( \alpha_1 \), while the variance of the workloads of all routes keeps the growth trend. The possible reason is that the customers with R type or RC type are uniformly distributed or semiclustered distribution. When the customer distribution is similar to R or RC, the enterprise can give different weights to the two goals according to its own preference. Moreover, we found that there exists a critical point when \( \alpha_1 \) is around 0.6. When \( \alpha_1 > 0.6 \), the above two goals are sensitive to the change of the value of the parameter \( \alpha_1 \).

Figure 5 indicates that changes about the first goal and the second one with different weights when the customers are clustered with group distribution. It can be found that the value of the first goal, namely, the total expected cost, decreases slowly with the increase of the value of \( \alpha_1 \) when \( \alpha_1 \leq 0.7 \), while that of the second goal maintains a slow growth trend. However, when \( \alpha_1 > 0.7 \), the above two goals are sensitive to the change of the value of the parameter \( \alpha_1 \), and with the increase of \( \alpha_1 \), those changes become more violent. This implies that the total expected cost will decrease sharply, while the standard variance of the workloads of all routes becomes larger sharply.

Figure 6 shows the total expected cost and the variance of the workloads of all routes with different weights when the customer is a C2 type that is clustered in group distribution. As Figure 6 shows, with the increase of \( \alpha_1 \), the value of first goal of the total expected cost is reduced, although it changes only slightly when \( \alpha_1 \leq 0.7 \). The value of the second goal and the variance of the workloads of all routes increase with the increase of the value of parameter \( \alpha_1 \).

4.2.2. The Effects of the Weight Parameter Vehicle \( \alpha_1 \) on Vehicles’ Routing Schemes. We examine the effects of the weight parameter vehicle \( \alpha_1 \) on vehicles routing schemes. Figures 7–9 show the different vehicle routing schemes for the instance with 25 customers classified RC group (i.e., semiclustered distribution), which considers the following three different weight scenarios: (1) vehicle routing scheme only considering the first goal (i.e., \( \alpha_1 = 1 \)), (2) vehicle routing scheme only considering the second goal (i.e., \( \alpha_1 = 0 \)), and (3) vehicle routing scheme considering both goals with \( \alpha_1 = 0.6 \).

Figure 7 shows the vehicle routing when \( \alpha_1 = 1 \) and \( \alpha_2 = 0 \). In this scenario, the optimization objective is to minimize the total expected costs only and completely disregards the balance of the workload of all routes. We can see from Figure 7 that the vehicle is basically distributed in the region. The total expected cost under this situation is the smallest. Balancing of the workload of all service routes is unsatisfactory in conjunction with consideration of Figure 4.

Figure 8 shows the vehicle routing when \( \alpha_1 = 0 \) and \( \alpha_2 = 1 \). In this scenario, the model proposed only considers the
optimization objective on the balancing of the workload of all service routes. We can see that a portion of the vehicles will carry out cross-district distribution services in this situation, which will obviously increase the total expected costs.

Figure 9 means that the decision-makers consider the two different goals simultaneously: the total expected travel cost and the balancing of the length of all service routes. We can see that several cross-regional distribution routes are generated, but the number of such cross-regional distribution routes is less than that of the second scenario. In this situation, the first goal and the second goal are trade-offs.

Next, we further made a comparative analysis about the workload balancing under two different routing schemes (i.e., one scheme with $\alpha_1 = 1$ and the other one with $\alpha_1 = 0.6$). As shown in Figure 10, the workloads balance of different vehicles in Figure 10(b) is better than that shown in Figure 10(a) obviously. That implies that it is significant to design a reasonable parameter $\alpha_1$ to achieve a better trade-off between the total expected cost and workload balances of all vehicle routes.

### 4.2.3. Validity Analysis on the ALNS Algorithm

Since there are no comparative data and no competing heuristics for our proposed biobjective problem, comparisons with alternative heuristics and best-known solutions are not possible. Therefore, the value of $\alpha_1$ is set to 1. Thus, the biobjective VRP is converted to a single-objective VRP, which only considers the total travel cost. At the same time, we remove the possibility of split operators to obtain a good solution for this problem and make a comparative analysis. The computational results are shown in Table 2.

Table 2 shows the results of using the various algorithms. In contrast to the results for the unsplit VRPSD (VRP with stochastic demand) algorithm and our algorithm, splitting deliveries can reduce the total travel cost. Compared with Lei et al. [47], we can see that our proposed algorithm can achieve a satisfactory solution when solving a single-goal VRP problem. Table 2 also shows that the seconds of our heuristic algorithm under C1 is significantly less than the other types (R, RC, and C2). Through analyzing the characteristics of the instances and the operators, it is found that the customers of instances of class C1 are distributed in clusters, which means a route servicing customer in one cluster will have fewer detours and significantly lower costs.

Figure 11 shows the convergences of ALNS algorithm to solve the instances with 25 customers under four different customer distribution situations (i.e., R, RC, C1, and C2). The value of the weight parameter $\alpha_1$ is set to 1. And the maximum

![Figure 10: Workloads of all service vehicles under two different routing schemes.](image)

**Table 2: Comparison of the modified Solomon instances.**

| Type | $|V|$ | Unsplit VRPSD | Lei [47] | Our heuristic algorithm |
|------|-----|--------------|----------|------------------------|
|      |     | #Route | Cost    | #Routes | Cost    | #Route | Cost    | Seconds |
| R    | 25  | 13     | 1231.68 | 16      | 1155.12 | 9      | 1103.15 | 988.89  |
| RC   | 25  | 13     | 1845.98 | 13      | 1658.39 | 9      | 1315.62 | 839.94  |
| C1   | 25  | 13     | 1092.43 | 14      | 969.98  | 9      | 984.64  | 876.38  |
| C2   | 25  | 13     | 1246.96 | 14      | 1159.72 | 9      | 1112.51 | 809.67  |
| R    | 50  | 25     | 2559.67 | 24      | 2492.75 | 25      | 2395.57 | 36451.00|
| RC   | 50  | 25     | 3917.40 | 25      | 3686.30 | 23      | 3260.72 | 33029.05|
| C1   | 50  | 25     | 2341.48 | 25      | 2265.76 | 27      | 2063.90 | 2866.91 |
| C2   | 50  | 25     | 2631.18 | 24      | 2547.59 | 28      | 2341.33 | 33102.07|

*Note.* It should be pointed out the cost in columns 4, 6, 8 in Table 2 means only the total length of routes, which does not include the fixed costs of vehicles.
number of iterations is equal to 2000. It can be seen from Figure 11 that the algorithm converges quickly. The current solution tends to the global optimal solution, which shows that the ANLS algorithm is of high quality in obtaining high-quality solutions for the VRP with stochastic and split deliveries.

From the above analysis, we can obtain the following conclusions or managerial insights. (1) There exists a negative correlation between the total expected cost and the workload balances of all routes. Specifically, the total expected cost will increase when we consider the workload balances of all routes. (2) It is observed that when the decision-maker gives the objective function with different weights, different vehicle routing plans are determined. Customers with different distributions are sensitive to different weight ranges. Therefore, it is necessary to set a reasonable value of the weight parameter based on corresponding customer distribution types to achieve a better trade-off between total expected cost and route workload balance. (3) The proposed ALNS algorithm embedded with the improved weighted sum method is effective to solve the proposed biobjective vehicle routing problem.

5. Conclusion

In this study, we focus on the fact that the driver’s pay and benefits are greatly affected by the corresponding travel lengths and workloads of routes. By considering the driver fairness and practical constraints, we addressed a new biobjective vehicle routing problem with stochastic demand and split deliveries, which considers the total expected cost and the balance of the workload of all routes. To solve the proposed problem, we developed an improved ALNS heuristic algorithm embedded with an improved optimization method. This improved optimization method is effective to normalize two different objectives and benefits us to find near-optimal solutions under different situations. We introduced new removal and insertion operators and modified some of the existing operators, which enable us to achieve high-quality solutions. Some instances modified from Solomon’s instances are used to validate the effectiveness of the proposed ALNS algorithm. Meanwhile, we designed an improved optimization method to normalize two different objectives, which benefits us to find near-optimal solutions under different situations. It is observed that when the decision-maker gives two different goals with different weights, different vehicle routing schemes are determined. Moreover, customers with different distributions are sensitive to different weight ranges. Therefore, it is necessary to set a reasonable value of the weight parameter based on corresponding customer distribution types to achieve a better trade-off between total expected cost and route workload balance.

This work can be extended by the following four research directions. First, multiperiod vehicle routing problem with stochastic demand, split deliveries, and route workload balance should be investigated. Second, to enhance its applicability in real-life scenarios, the relationship between the route workload balance and the weight coefficients can be explored in the future. Finally, further research on the solution algorithm is necessary, and new removal operators,
insertion operators, and adaptive scoring mechanisms may enhance the solution quality.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

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

Acknowledgments

The work that is described in this paper was jointly supported by the Natural Science Foundation of Hunan Province, China (nos. 2021JJS0857 and 2021JJS1167) and the High-End Think Tank Project of Central South University (no. 2021znzk08).

References


