Hybrid Discrete Mushroom Reproduction Algorithm for Open Location-Routing Problem with Fuzzy Demands (Postprint)

Authors: Liu Fan, Zhang Huizhen, Zhou Xun

Date: 2020-09-28T00:00:00+00:00

Abstract

For the research on the open location-routing problem with constraints, a mathematical model is established under fuzzy demand conditions with the objective of minimizing the sum of warehouse location costs, vehicle traveling distance costs, opportunity loss costs, and extra distances. Through modifications to the mushroom reproduction algorithm, partial mapping crossover and path relinking algorithm are employed to replace the parent update mechanism in the original algorithm; in the neighborhood search component, a probabilistic method is utilized for neighborhood selection; a stochastic simulation program is used to simulate the designed routes and calculate the extra traveling distance and opportunity loss costs resulting from service failure. While preserving the original characteristics of the algorithm, it is successfully applied to combinatorial optimization problems. Finally, through a series of numerical experiments and comparisons, the correctness and validity of the model, as well as the computational efficiency and optimization capability of the hybrid discrete mushroom reproduction algorithm, are verified.

Full Text

Preamble

Title: Hybrid Discrete Mushroom Reproduction Algorithm for Solving Open Location-Routing Problem with Fuzzy Demands

Authors: Liu Fan, Zhang Huizhen†, Zhou Xun
Affiliation: Business School, University of Shanghai for Science & Technology, Shanghai 200093, China

Abstract: This paper proposes a mathematical model for the open location routing problem with fuzzy demands, considering constraints on warehouse location costs, vehicle travel distance costs, opportunity loss costs, and additional distances, with the objective of minimizing their sum. By modifying the mushroom reproduction algorithm, we replace the parent update mechanism with partial mapping crossover and path relinking algorithms; use a probabilistic method for neighborhood selection in the local search component; and employ a stochastic simulation program to simulate designed routes and calculate additional travel distances and opportunity loss costs resulting from service failures. While retaining the original characteristics of the algorithm, we successfully adapt it to combinatorial optimization problems. Finally, through a series of computational tests and comparisons, we verify the correctness and effectiveness of the model and the computational efficiency and optimization capability of the hybrid discrete mushroom reproduction algorithm.

Keywords: open location-routing problem; fuzzy demand; mushroom reproduction algorithm; path relinking algorithm; stochastic simulation

0 Introduction

Efficient and precise vehicle routing planning and rational warehouse location have significant impacts on modern logistics management, not only reducing distribution costs but also improving customer satisfaction and enterprise competitiveness. Many scholars have noted that when the vehicle routing problem (VRP) and facility location problem (FLP) are considered separately, managers cannot make optimal decisions due to insufficient consideration of influencing factors. Today, their integrated problem—the location-routing problem (LRP) —has become a research hotspot in logistics management, supply chain management, and operations optimization.

By adding different constraints to the standard LRP according to practical requirements, various LRP extensions can be formed. For instance, considering uncertainties such as customer demands and service times in logistics distribution, various fuzzy constraints can be incorporated into LRP models, forming location-routing models with fuzzy demands [7], location-routing models with time windows under uncertainty [12], multimodal transportation network location-routing models with fuzzy demands [8], multi-objective fuzzy location-routing problem models in emergency logistics systems [4], and dynamic location-routing problem models for emergency supplies considering post-earthquake road network changes with multiple vehicle types [16,17]. Additionally, due to the rise of third-party logistics in recent years, many enterprises undergoing strategic restructuring have focused more on core competencies by outsourcing logistics services to third-party companies. This has led to the proposal of open location-routing problems (OLRP) for third-party logistics distribution. Although OLRP also faces various fuzzy constraints in practice, as OLRP is a relatively new problem, existing literature contains limited research on OLRP with fuzzy constraints. Only in reference [15] did Wang Haijun et al. establish a multi-objective mixed-integer programming model for OLRP, using probability to select safer and more reliable routes for distribution, and solved an instance regarding rescue material distribution in the Wenchuan earthquake using NSGA-II (Non-dominated Sorting Genetic Algorithm) and NSDE (Non-dominated Sorting Differential Evolution Algorithm).

LRP belongs to the class of NP-hard problems. Exact algorithms (such as branch-and-bound, column generation, etc.) can only solve small-scale LRP problems. However, LRP problems abstracted from actual logistics distribution are often large-scale, making it difficult for exact algorithms to provide solutions within acceptable computational time. Consequently, intelligent optimization algorithms capable of delivering near-optimal solutions within limited computational time have become the primary solution method for LRP problems. For example, ant colony algorithms [10–13], genetic algorithms [4,8,15,17–19], and simulated annealing algorithms [5,12] have all achieved satisfactory results in solving LRP problems.

This paper combines fuzzy demands with OLRP, proposing the Open Location-Routing Problem with Fuzzy Demands (OLRP-FD), which effectively addresses the uncertainty of customer demands currently faced in third-party logistics distribution and is of great significance for reducing enterprise distribution costs. Furthermore, this paper introduces new neighborhood search methods into the basic mushroom reproduction algorithm [14] and designs a discrete mushroom reproduction algorithm for solving OLRP-FD problems. This not only provides a new solution approach for OLRP problems but also broadens the application field of the mushroom reproduction algorithm. The neighborhood search in the algorithm generates offspring spores to refine these regions based on evaluating the search of parent mushroom neighborhoods to find better solutions. Numerical experiments demonstrate that the algorithm can provide relatively good solutions for OLRP-FD within limited computational time.

---

1 Model

The difference between OLRP-FD and standard OLRP lies in that customer demands are fuzzy rather than deterministic. Only when the delivery vehicle arrives at a customer can the actual demand be known. Therefore, in OLRP-FD, when a delivery vehicle arrives at a customer, it may find that the current customer cannot be served due to insufficient remaining vehicle capacity. In this case, the vehicle must return to the warehouse for reloading and then come back to serve the unserved customer. Fuzzy customer demands in OLRP-FD not only lead to additional travel distances from vehicles returning to warehouses for reloading but also result in opportunity loss costs when some customers cannot be served due to insufficient pre-allocation of vehicle capacity.

This paper uses credibility measure theory to construct the mathematical model for OLRP-FD. The model minimizes total distribution costs (including vehicle fixed activation costs, planned total travel distance costs, and opportunity loss costs for unserved customers) and minimizes the total additional travel distance caused by service "failures," subject to vehicle capacity and maximum travel distance constraints.

1.1 Credibility Theory

Fuzzy set theory was first proposed by Zadeh [21] through membership functions. Subsequently, Kaufmann [23] and Zadeh [22] perfected the possibility measure theory for fuzzy events, but it lacked self-duality. Liu [9] proposed credibility measure theory for fuzzy variables to define its self-duality property.

Let $(\Theta, \mathcal{P}(\Theta), \mathrm{Pos})$ be a possibility space, where $A \in \mathcal{P}(\Theta)$ is an event and $\mathrm{Pos}\{A\}$ denotes the possibility of event A occurring. The necessity of event A (i.e., the necessity of event A occurring) is then defined as $\mathrm{Nec}\{A\} = 1 - \mathrm{Pos}\{A^c\}$, where A^c denotes the complement of event A. Consequently, the credibility of event A can be obtained as $\mathrm{Cr}\{A\} = \frac{1}{2}(\mathrm{Nec}\{A\} + \mathrm{Pos}\{A\})$.

In this paper, customer demand is represented using triangular fuzzy variables $\tilde{d} = (d_1, d_2, d_3)$, where d_1 and d_3 are the lower and upper bounds of customer demand, respectively, within which the actual demand must fall; d_2 represents the customer demand corresponding to a membership function value of 1. These three parameters are generally obtained from relevant data or subjective estimates based on experience. Assuming a customer' s actual demand is r, the membership function is shown in Figure 1 [Figure 1: see original paper], and its possibility, necessity, and credibility can be calculated using equations (1)-(3), respectively.

1.2 Problem Description

Based on credibility theory, this paper constructs a mathematical model for the open location-routing problem with fuzzy demands. Selected warehouses dispatch vehicles to provide distribution services to customers under the following assumptions: (a) each vehicle can serve at most one route; (b) warehouses and vehicles have fixed capacity limits; (c) vehicles have maximum travel distance limits; (d) after serving the last customer, vehicles do not return to the warehouse but instead return to the third-party logistics company' s vehicle distribution center; (e) due to the fuzziness of each customer' s demand, some customers may not be served due to insufficient vehicle availability, resulting in corresponding opportunity loss costs.

For the case of fuzzy customer demands, this paper uses credibility theory to determine whether the current vehicle and warehouse can serve a new customer. Assume the capacities of vehicles and warehouses are Q and P, respectively. When a vehicle or warehouse has served the e-th customer in its planned route, the current remaining capacity of the vehicle is $Q_e = Q - \sum_{j=1}^{e} q_j$; the current remaining capacity of the warehouse is $P_e = P - \sum_{j=1}^{e} p_j$. Then, the credibility that the next customer' s demand does not exceed the vehicle capacity and warehouse capacity can be expressed as equations (4) and (5), respectively.

Furthermore, based on equation (3), $\mathrm{Cr}\{d_{e+1} \le Q_e\}$ and $\mathrm{Cr}\{d_{e+1} \le P_e\}$ can be expressed as equations (6) and (7), respectively. Cr serves as a parameter for evaluating whether a distribution vehicle or warehouse can serve a particular customer. When $\mathrm{Cr} = 0$, the customer will definitely not be served by the current vehicle or warehouse; when $\mathrm{Cr} = 1$, the customer will definitely be served.

This paper sets two thresholds, DPI and API ($\mathrm{DPI}, \mathrm{API} \in [0, 1]$). By comparing the credibility of vehicles and warehouses with the set thresholds, when Cr is greater than the threshold, the customer will be served by the current vehicle or warehouse; otherwise, it will not. Obviously, the values of DPI and API significantly impact planned travel distances and additional travel distances. When DPI is low, vehicle capacity is utilized as much as possible, making it highly probable that customers later in the route will fail to be served, causing vehicles to return to the warehouse for reloading and generating additional travel distances. When DPI is high, more vehicles are dispatched from the warehouse, increasing planned travel distances but reducing the number of service failures and consequently reducing additional distances. When API is low, the warehouse consumes its capacity to serve customers as much as possible, which may not only lead to capacity overflow but also result in insufficient reserved vehicles at the warehouse, leaving some customers assigned to this warehouse unserved and generating substantial opportunity loss costs. When API is high, the warehouse is more cautious in allocating its capacity to customer demands, resulting in fewer capacity overflow situations. This paper uses a stochastic simulation program to solve for the corresponding additional travel distances and conducts sensitivity analysis on DPI to obtain optimal parameter settings.

1.3 Mathematical Model

The main parameters of the model are set as follows:

- I: Set of warehouses
- J: Set of customers
- J_K: Third-party logistics company vehicle distribution center point
- K: Set of vehicles
- $\tilde{d}_j$: Demand of the j-th customer, $\tilde{d}_j = (d_{j1}, d_{j2}, d_{j3})$
- Q_k: Capacity of the k-th vehicle, $Q_k \in K$
- P_i: Capacity of the i-th warehouse, $P_i \in I$
- O_i: Opening cost of the i-th warehouse, $O_i \in I$
- F_k: Activation cost of the k-th vehicle, $F_k \in K$
- AD_k: Additional distance traveled by the k-th vehicle
- C_{ij}: Unit travel cost from point i to point j, $i, j \in V$
- TD_k: Maximum travel distance of vehicle k, $TD_k \in K$
- D_{ij}: Euclidean distance from point i to point j, $i, j \in V$

- B_j: Loss cost when customer j is not served, $j \in J$

Decision variables are set as follows:

- $X_{ij}^k = \begin{cases} 1, & \text{if vehicle } k \text{ travels from point } i \text{ to point } j \\ 0, & \text{otherwise} \end{cases}, \forall i, j \in V, \forall k \in K$
- $Y_{ij} = \begin{cases} 1, & \text{if warehouse } i \text{ serves customer } j \\ 0, & \text{otherwise} \end{cases}, \forall i \in I, \forall j \in J$
- $Z_i = \begin{cases} 1, & \text{if warehouse } i \text{ is opened} \\ 0, & \text{otherwise} \end{cases}, \forall i \in I$

Based on the above assumptions, model parameters, and decision variables, the mathematical model for OLRP-FD can be constructed as follows:

Objective functions: - (8) Minimize total cost, including fixed activation costs of warehouses and vehicles, and planned total travel distance costs - (9) Minimize the sum of additional travel distances caused by service "failures" and opportunity loss costs for unserved customers

Constraints: - (10) Fuzzy chance constraint ensuring that when distribution vehicles visit all customers, vehicle capacity and customer demand satisfy the set credibility level - (11) Fuzzy chance constraint indicating that when warehouses serve customers, their capacity and customer demand satisfy the set credibility level - (12) Each customer can be served by at most one vehicle - (13) Eliminates sub-tours - (14) Any vehicle can be activated at most once - (15) Ensures route continuity with equal flow in and out of each point - (16) and (17) Ensure that when a customer is served by a warehouse, there must be a route connecting them - (18) Vehicle maximum travel distance constraint - (19) and (20) Represent relationships between decision variables - (21)-(23) Represent three groups of 0-1 decision variables - (24) Auxiliary variable for eliminating sub-tours

---

2 Hybrid Discrete Mushroom Reproduction Optimization Algorithm for OLRP-FD

The Mushroom Reproduction Optimization (MRO) algorithm [14] is a novel swarm intelligence optimization algorithm based on the growth and reproduction characteristics of mushrooms, which has achieved good results in solving continuous optimization problems. To further broaden the application field of the MRO algorithm, this paper combines it with greedy clustering algorithms, path relinking algorithms, and neighborhood search to design a Hybrid Discrete Mushroom Reproduction Optimization (HDMRO) algorithm for solving OLRP-FD.

HDMRO consists of three components: (a) Initialization: using a greedy clustering algorithm to generate a set of high-quality initial solutions to enable better optimization iterations; (b) Discrete mushroom reproduction algorithm: based on the basic MRO algorithm, combined with the specific characteristics of the OLRP-FD problem, integrated with partial mapping crossover and path relinking algorithms, as well as neighborhood search, to gradually optimize the solution; (c) Stochastic simulation program: simulating actual customer demands to calculate additional distances incurred during vehicle service.

2.1 Basic Mushroom Reproduction Algorithm

The mushroom reproduction algorithm [14] simulates the mechanism of mushroom reproduction and growth, where parent mushrooms spread their spores through the power of wind to explore various reproduction areas, refine the search space, and find better reproduction areas for growth and reproduction activities. The basic mushroom reproduction algorithm was originally designed for continuous optimization problems, with key steps as follows: Before local search, based on the average fitness value of each colony ($\mathrm{Avg}(i)$) and the average fitness value of all colonies (T_{avg}), the relationship between $\mathrm{Avg}(i) + T_{\mathrm{avg}}/c$ and T_{avg} is compared (where c is a fixed threshold, $c \in [1, 10]$) to determine whether some colonies need to be eliminated. If the former is greater than the latter, no parent mushroom deletion is performed; otherwise, parent mushrooms are updated to select better parents to refine the search space for finding excellent solutions.

When updating parent mushrooms, artificial wind is constructed for information exchange and updating among spores. The initial calculation formula for the artificial wind of the j-th offspring mushroom is (25):

$$\mathrm{wind}_j = \beta \times (\mathrm{Mov}(X_i, X_k) - X_i) \times \delta \times (s - \mathrm{rand}) + \delta \times \mathrm{rand} \times (s - s_{\mathrm{avg}})$$

where X_i and X_k are the parents of the i-th and k-th colonies, respectively; β ($\beta \in [1, +\infty)$) is the wind density; δ ($\delta \in (0, 1)$) is a parameter controlling the wind step size; s is the search radius, typically determined based on the specific problem; s_{avg} is the average search radius; and rand is a random number in $(0, 1)$ that adjusts the wind direction parameters.

In addition to the above steps, the MRO algorithm typically uses equation (26) for neighborhood search:

$$X_{ij} = X_i + \mathrm{rand} \times (s - s_{\mathrm{avg}})$$

where X_{ij} is the j-th spore generated by the i-th parent mushroom.

The basic MRO algorithm primarily iterates through equations (25) and (26) for optimization. Specific iterative optimization steps can be found in reference [14].

2.2 Solution Representation and Initialization

2.2.1 Solution Representation This paper adopts natural number encoding. Assume there are m warehouses numbered $1, 2, \ldots, m$; n customers numbered $m+1, m+2, \ldots, m+n$. Then a feasible solution for an OLRP-FD instance with 3 warehouses and 15 customers can be encoded as shown in Figure 2 [Figure 2: see original paper]. If there are no customer numbers after a warehouse number, it means the warehouse is not opened; if there are customer numbers after a warehouse number, it means the warehouse is opened, and the number of times the warehouse number appears equals the number of delivery routes departing from that warehouse. A vehicle departs from the corresponding warehouse and serves customers sequentially from left to right until the next warehouse number appears, at which point the vehicle completes its delivery task. As shown in Figure 2, warehouse 1 is closed, warehouses 2 and 3 are opened; the two delivery routes from warehouse 2 are $2 \to 5 \to 7 \to 4 \to 10$ and $2 \to 6 \to 12 \to 14 \to 13 \to 18$; the delivery route from warehouse 3 is $3 \to 17 \to 16 \to 9 \to 15 \to 16 \to 8$. This encoding method is simple and intuitive, effectively interpreting which customers each warehouse serves and the route each vehicle travels.

2.2.2 Initialization The quality of initial feasible solutions not only affects solution quality but also influences algorithm convergence speed. To obtain better initial feasible solutions, this paper uses a greedy clustering algorithm for initialization, with specific steps as follows:

1. **Cluster customers**: Using greedy thinking, customers are clustered based on vehicle capacity, customer fuzzy demands, and distances between customers. A clustering process can be briefly described as: (a) Generate an empty cluster (cluster capacity equals vehicle capacity) and randomly select an unclustered customer to join the cluster; (b) From unclustered customers, select the customer closest to the most recently added customer in the cluster, and calculate the credibility between this customer' s fuzzy demand and the vehicle' s remaining capacity—if the value is greater than the given DPI, add the customer to the current cluster; otherwise, disregard this customer for the current cluster; (c) Repeat step (b) to continue selecting the customer closest to the last added customer, judging whether it can join the current cluster until the remaining capacity is insufficient to serve any new customers, at which point one clustering is completed. Repeat steps (a)-(c) to generate new clusters until all customers are clustered.

2. **Select open warehouses and assign customer clusters**: First, calculate the cluster centroid using equation (27) based on customer coordinates in each cluster. Then calculate the sum of distances from each warehouse to customer cluster centroids U_i and the sorting index W_i using equations (28) and (29), respectively. Finally, sort warehouses in descending order of W_i values, open the warehouse with the largest W_i, and assign the nearest customer cluster to it, followed by the next nearest cluster, and so on.

When the warehouse' s remaining capacity is insufficient to serve any unassigned cluster, open the next warehouse in the sorting sequence until all clusters are assigned. When assigning a customer cluster to a warehouse, it is necessary to consider whether the credibility between the cluster' s total demand and the warehouse' s remaining capacity is greater than the given API value—if greater, assign the cluster to the warehouse; otherwise, check whether the next unassigned cluster satisfies this condition until the warehouse capacity is insufficient to serve any cluster.

3. **Construct delivery routes**: For each customer cluster and its assigned warehouse, construct a delivery route following the order of warehouse first, then customers.

2.3 Partial Mapping Crossover and Path Relinking Algorithm

To effectively apply the mushroom reproduction algorithm to solve LRP problems and improve its performance, this paper uses partial mapping crossover [24] and path relinking algorithm [20] to replace the global search processing method (25) that uses wind power on parent mushrooms in the basic mushroom reproduction algorithm. The specific method is: first, perform partial mapping crossover between the parent mushroom of the i-th colony and the global optimal solution or the parent mushroom of the k-th colony to obtain a new solution; then use the path relinking algorithm to connect the parent of the current iteration colony with the solution generated by partial mapping crossover to search for better solutions between them.

Partial mapping crossover: Generate a random number $r \in [0, 1]$. If $r \leq 0.5$, perform crossover between the global optimal solution and the parent mushroom of the i-th colony; if $r > 0.5$, perform crossover between the parent mushroom of the i-th colony and the parent mushroom of the k-th colony. Experiments prove that the partial mapping crossover algorithm adopted in this paper can effectively accelerate algorithm convergence.

Figure 3 [Figure 3: see original paper] illustrates the partial mapping crossover process for an OLRP-FD instance with 2 warehouses and 6 customers. As shown in Figure 3: first, randomly select two positions 4 and 7 in parents P1 and P2, copy all information from positions 4 to 7 of the parent individuals to the corresponding offspring individuals Child1 and Child2 (represent remaining positions of offspring individuals Child1 and Child2 with *), obtaining Child1(*) and Child2(*), and a mapping set* $(4 \to 9, 1 \to 6, 5 \to 2, 7 \to 4)$*; second, scan individuals P1 and P2 from left to right—if the code at position i in parent P1 (P2) is not in offspring Child2 (Child1), copy it to the corresponding position in Child2, obtaining Child1' (*) and Child2' (*); third, use the obtained mapping to fill remaining empty positions in Child1' (*) and Child2' (*), e.g., for vacant positions 1, 2, and 3 in Child1' (*), which should contain codes 1, 7, and 5 from parent P2' s corresponding positions, but since these codes already exist in Child1' (*), adjust them according to mapping rules $(1 \to 6, 7 \to 4 \to 9, 5 \to 2)$ to 6, 9, and

2, respectively, obtaining Child1' , and similarly obtain Child2' ; finally, check the codes at position 1 of Child1' and Child2' —if they are customer codes (as shown in Figure 3, the codes at position 1 of Child1' and Child2' are 6 and 5, respectively), randomly select any warehouse (as shown in Figure 3, warehouse 1 is selected) to exchange with them, obtaining final solutions Child1 and Child2.

Path relinking algorithm: The path relinking algorithm primarily exchanges information between two solutions (the current solution X_i serving as a parent spore of the current population and the target solution X_t obtained through partial mapping crossover). The algorithm process can be described as follows: (a) Calculate the Hamming distance HD between initial solution X_i and guiding solution X_t, and set target solution $X^* = X_t$; (b) If HD > 0 and HD is odd, compare codes at the same positions in X_i and X_t from left to right sequentially and exchange different codes; if HD > 0 and HD is even, compare from right to left and exchange different codes; if HD $= 0$, output target solution X^*. The exchange process can be summarized as: assuming the codes at position a differ between X_i and X_t, find position b in X_i where the code equals $X_t(a)$, then set $X_i(b) = X_i(a)$ and $X_i(a) = X_t(a)$; (c) Update HD := HD - 1, exchange current solution X_i with guiding solution X_t, then return to step (b).

The path relinking algorithm can avoid the conversion process of solution information from discrete to continuous and back to discrete that occurs when most continuous optimization algorithms solve discrete problems, effectively preventing information loss.

2.4 Stochastic Simulation Program

As mentioned earlier, since each customer' s demand in the OLRP-FD model is represented by triangular fuzzy variables, after HDMRO optimizes warehouse locations and vehicle routes, this paper uses stochastic simulation [12] to simulate each customer' s actual demand and then calculate opportunity loss costs (costs incurred when there are insufficient warehouse vehicles to serve all customers) and additional vehicle travel distances (when a vehicle traveling along a planned route arrives at a customer and finds its remaining cargo insufficient to meet the customer' s demand, it must return to the warehouse to reload and then serve the customer, generating additional travel distance equal to the round-trip distance between the current customer and the warehouse; or when a vehicle arrives at a customer and finds its remaining cargo insufficient, it must return to the warehouse for reloading, but if the sum of the distance from the current customer back to the warehouse and the previously traveled distance exceeds the maximum distance constraint, the warehouse only needs to dispatch another vehicle to serve the current customer, with the additional distance being the distance from the current customer to the warehouse). The specific steps of the stochastic simulation program are as follows:

(a) Generate actual demand for each customer. For each customer $j \in J$: (i) generate a random number r in interval $[d_{j1}, d_{j3}]$ and calculate its corresponding membership degree μ; (ii) randomly generate a real number α in interval $[0, 1]$—if $\alpha \leq \mu$, adopt the customer' s actual demand as r; otherwise, regenerate r and μ until the above relationship is satisfied. This procedure simulates the actual demand for all customers.

(b) Move along each delivery route generated by HDMRO, calculating additional distances and opportunity loss costs caused by vehicle service "failures" based on actual demands.

(c) Repeat the above two steps 400 times, taking the average additional distance and average loss cost as the final additional distance and loss cost.

2.5 HDMRO Algorithm Flow

The specific steps of the HDMRO algorithm are as follows:

(a) Initialize parameters: number of parent mushrooms (population size M), number of spores generated by each parent N, maximum algorithm iterations G (set $g = 1$), threshold c.

(b) Call the greedy clustering algorithm to generate initial feasible solutions. Each parent mushroom generates N spores through neighborhood search, calculates the fitness value of each individual, and selects the individual with the best fitness value from each parent mushroom and its spores as the new parent, updating each parent mushroom.

(c) Begin formal iteration. If $g < G$, proceed to step (d); otherwise, proceed to step (e).

(d) Call partial mapping crossover to generate a guiding solution required by the path relinking algorithm, then call the path relinking algorithm to connect the parent solution of the i-th population at the current iteration with the guiding solution generated by partial mapping crossover, searching the region between the two solutions. Update the parent mushroom of each colony and the global optimal value.

(e) Neighborhood search. Use the probabilistic method to select among three neighborhoods—parent mushrooms reproduce according to the selected neighborhood to generate spores and search the solution space. Update the parent mushroom of each colony and the global optimal value.

(f) $g = g + 1$—if $g < G$, return to step (c) and continue iterating; otherwise, output the current optimal solution and proceed to step (g).

(g) Simulate the optimized routes using the stochastic simulation program to solve for the corresponding additional travel distances and opportunity loss costs from unserved customers.

The flowchart of the HDMRO algorithm is shown in Figure 7 [Figure 7: see original paper].

3 Computational Tests and Results Analysis

To verify the effectiveness of the hybrid mushroom algorithm in solving the OLRP-FD model, this paper randomly generates two groups of instances of different scales following reference [11]. Customers are randomly generated from three intervals: [1,35], [36,60], and [61,110]; other relevant data are shown in Table 1 . The experimental environment implements the hybrid mushroom algorithm coding and related instance tests in MATLAB R2017a under Windows 10. Relevant parameters: parent population size $N = 40$; offspring number $M = 8$; maximum iterations $G = 600$; threshold $c = 10$.

3.1 DPI Sensitivity Analysis

This section varies DPI values in the interval 0-1 with a step size of 0.1. For convenience in testing and to save computational time, the API value is fixed at 1 (ensuring warehouses can definitely serve all customers assigned to them). Test data are shown in Tables 2 and 3 .

Figures 8 [Figure 8: see original paper] and 9 [Figure 9: see original paper] correspond to Tables 2 and 3, respectively. As shown in Figures 8 and 9, as DPI values increase, the number of activated vehicles increases, leading to increased planned route distances. At this point, the possibility of vehicle service failures decreases, reducing additional travel distances. Warehouse opening costs remain unchanged, and opportunity loss costs are zero because each warehouse has sufficient available vehicles, and additional travel distances do not exceed available vehicle quantities. Therefore, opportunity loss costs are zero in these two instances, indicating that every customer will be served. From Figures 8 and 9, it can be observed that as DPI values change with a fixed API value, the total target cost reaches its minimum when DPI is approximately 0.6.

3.2 Instance Testing

Since customer demands in this study are fuzzy, all customer and warehouse locations are randomly generated, and research on the open location-routing problem model with fuzzy demands is scarce both domestically and internationally, the HDMRO results shown in Tables 2 and 3 cannot be compared with previous studies. Therefore, to evaluate HDMRO's efficiency, this paper chooses to relax equations (10) and (11) to transform fuzzy demand constraints into deterministic demands, and compare with the relaxed lower bound. Meanwhile, the inequality constraint (12) becomes an equality constraint. The relaxed formulas are shown as (30) and (31).

Obviously, when the left boundary of fuzzy demand is the customer' s deterministic demand, the total customer demand decreases compared to the fuzzy demand case, and the utilization rates of warehouses and vehicles also decrease.

Therefore, the total cost under deterministic demand can be concluded to be a lower bound of the corresponding total cost under fuzzy demand. In Table 4 , the first column shows instance scales, the next three columns compare solution quality, and the last three columns compare running times.

As shown in Table 4, the lower bounds solved by CPLEX for the two test instances are 503.08 and 677.31, respectively. The first instance took 4011.34 seconds, while the second instance did not obtain an optimal solution within the 3-hour limit. In contrast, HDMRO solved the two instances in 128.25 seconds and 223.90 seconds, respectively. Therefore, compared with CPLEX, a solver using exact algorithms, this algorithm has higher computational efficiency and can obtain satisfactory solutions within shorter computer running times.

To further verify the effectiveness of HDMRO and the OLRP-FD model, this paper selects 7 groups of standard instances from the standard CLRP benchmark library (http://prodhonc.free.fr/Instances/instances_{us}.htm) for modification and testing. The modification steps are as follows: (a) The default demand in standard instances is the left boundary of fuzzy demand (d_1), the right boundary (d_3) is three times the left boundary, and d_2 is a randomly generated integer in $[1.5d_1, 2.5d_1]$; (b) Compare vehicle capacity Q with the maximum d_3 among all customers—if Q is greater than d_3, keep vehicle capacity unchanged; otherwise, set the maximum d_3 as vehicle capacity. Warehouse capacity P is set to three times the default value. Other parameters use database default values unchanged.

Table 5 shows the lower bounds solved by CPLEX optimization software within 3 hours for 7 modified OLRP instances. Compared with the results of HDMRO solving OLRP with fuzzy demands, HDMRO demonstrates good performance in solving such complex problems. Moreover, the Gap values are at least greater than 19.48% across the 7 instances, indicating that total costs can differ significantly due to uncertain decision-making situations. Therefore, when decision-makers have insufficient information for decision-making, using scientific methods for reasonable prediction and cautious decision-making can avoid enterprise losses to some extent (in this group of instances, customer opportunity loss costs are 20 times the corresponding customer' s actual demand).

To further verify HDMRO' s solution efficiency, this paper selects 16 groups of instances from the CLRP standard database for comparison. Table 6 shows the comparison results using the best values from 10 runs of HDMRO on each instance.

Among the 16 test instances in Table 6, GRASP [25] (Greedy Randomized Adaptive Search Procedure) can obtain optimal solutions for 7 instances, MAPM [26] (Memetic Algorithm with Population Management) for 10 instances, LRGTS [27] (Lagrangian Relaxation-based Granular Tabu Search) for 6 instances, 2-phase HGTS [28] (Two-phase Hybrid Granular Tabu Search) for 10 instances, while HDMRO can obtain optimal solutions for 11 instances. Compared with the other four algorithms, HDMRO can solve more optimal solutions, demonstrating greater advantages in solution quality. Regarding solution time, due to differences in computer hardware and compilation software (the first three referenced algorithms were coded in C++ and tested on a Dell PC Optiplex GX260 with a 2.4 GHz Pentium 4 CPU and 512MB RAM under Windows XP; the 2-phase HGTS algorithm was coded in C++ under Linux 11.04 with 2GB RAM and Intel Core Duo (2.0GHz) configuration), direct comparison is not possible. However, it is certain that HDMRO can obtain satisfactory quality solutions within acceptable time. Therefore, based on the above data, the HDMRO algorithm demonstrates good effectiveness in solving such problems.

3.3 Comparison Between OLRP and CLRP

This section compares modified 7 groups of instances to analyze the differences between open location-routing problems and conventional location-routing problems, as well as their differences under fuzzy demands, to help enterprises decide whether to choose distribution service outsourcing. All instances in this section are solved using HDMRO. Parameters: DPI = 0.6 and API = 1 when demand is fuzzy; vehicle activation cost = 1000; unit distance cost = 100; other parameters remain the same as in Section 3.2. Solution results are shown in Table 7 .

As shown in Table 7, under deterministic demand, the Gap1 between OLRP and CLRP is at least 15.44%, indicating that without considering the last leg of transportation distance and assuming all other logistics-related costs are consistent between the enterprise and third-party logistics companies, decision-makers need to consider the difference between the service fee charged by third-party logistics companies and the cost savings from this distance when deciding whether to outsource logistics distribution services. Under uncertain demand, the Gap2 between OLRP-FD and CLRP-FD (Gap2 = $\frac{\text{CLRP-FD}-\text{OLRP-FD}}{\text{OLRP-FD}} \times 100\%$) is at least 19.66%, indicating that when demand is uncertain, CLRP-FD' s total cost is higher than OLRP-FD's. This shows that although additional travel distances and opportunity loss costs increase total costs, OLRP-FD increases less relative to CLRP-FD, giving enterprises more cost room to consider outsourcing logistics services to third-party providers. Second, when customer scale is 20, Gap2 values are greater than Gap1 values mainly because with fewer customers in this group, vehicles serving assigned customers do not reach the maximum travel distance limit and can successfully serve all customers without warehouses needing to activate more vehicles, thus generating no opportunity loss costs. Meanwhile, considering additional travel distances, CLRP-FD generates higher additional distances than OLRP-FD, making Gap2 greater than Gap1. When customer scale is 50, Gap2 values are smaller than Gap1 values due to maximum distance limits and HDMRO' s route planning, which limit the number of vehicles activated by warehouses, resulting in insufficient vehicles to serve all customers assigned to the warehouse and generating substantial opportunity loss costs. Although planned route costs still dominate total costs, OLRP-FD' s opportunity loss costs are higher than CLRP-FD' s, leading to some reduction in Gap values and making Gap2 smaller than Gap1.

4 Conclusion

This paper establishes a mathematical model for the open location-routing problem with fuzzy demands, minimizing the sum of costs including warehouse opening, vehicle usage, vehicle travel distances, opportunity losses from unserved customers, and additional travel distances. Simultaneously, we propose a hybrid discrete mushroom reproduction algorithm for solving discrete problems, using a combination of partial mapping crossover and path relinking algorithms to replace the operations of refining search space and deleting populations for excellent parent mushrooms in the original algorithm, reasonably avoiding information loss during conversion between discrete and continuous spaces, and enabling better exploration of the solution space to improve algorithm iteration efficiency. Additionally, the probabilistic method is used for neighborhood selection during local search to strengthen local search capabilities. Finally, through DPI sensitivity analysis, comparison between HDMRO and CPLEX results on modified instances, and comparative analysis between OLRP vs. CLRP and OLRP-FD vs. CLRP-FD results, we analyze the possibility of enterprises choosing third-party logistics services and verify the effectiveness of the algorithm and model.

References

[1] Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space [J]. IEEE transactions on Evolutionary Computation, 2002, 6 (1): 58-73.

[2] 张晓楠, 范厚明, 李剑锋. 变动补偿的多模糊选址-路径机会约束模型及算法 [J]. 系统工程理论与实践, 2016, 36 (02): 442-453. (Zhang Xiaonan, Fan Houming, Li Jianfeng. Chance-constrained model and algorithm for LRP with multiple fuzzy variables under change-reward [J]. System Engineering-Theory & Practice, 2016, 36 (02): 442-453.)

[3] 刘长石, 彭怡, 寇纲. 震后应急物资配送的模糊定位-路径问题研究 [J]. 中国管理科学, 2016, 24 (05): 111-118. (Liu Changshi, Peng Yi, Kou Gang. Study of location-routing problem of emergency goods distribution in post-earthquake [J]. Chinese Journal of Management Science, 2016, 24 (05): 111-118.)

[4] 郑斌, 马祖军, 方涛. 应急物流系统中的模糊多目标定位-路径问题 [J]. 系统工程, 2009, 27 (08): 21-25. (Zheng Bin, Ma Zujun, Fang Tao. Fuzzy multi-objective location-routing problem of emergency logistics system [J]. System Engineering, 2009, 27 (08): 21-25.)

[5] Vincent F Y, Lin S Y. A simulated annealing heuristic for the open location-routing problem [J]. Computers & Operations Research, 2015, 62: 184-196.

[6] Pichka K, Bajgiran A H, Petering M E H, et al. The two echelon open location routing problem: Mathematical model and hybrid heuristic [J]. Computers & Industrial Engineering, 2018, 121: 97-112.

[7] Ghaffari-Nasab N, Ahari S G, Ghazanfari M. A hybrid simulated annealing based heuristic for solving the location-routing problem with fuzzy demands [J]. Scientia Iranica, 2013, 20 (3): 919-930.

[8] Fazayeli S, Eydi A, Kamalabadi I N. Location-routing problem in multimodal transportation network with time windows and fuzzy demands: Presenting a two-part genetic algorithm [J]. Computers & Industrial Engineering, 2018, 119: 233-246.

[9] B. Liu. Uncertainty Theory: An Introduction to its Axiomatic Foundations [M]. Springer, Berlin, 2004.

[10] Mehrjerdi Y Z, Nadizadeh A. Using greedy clustering method to solve capacitated location-routing problem with fuzzy demands [J]. European Journal of Operational Research, 2013, 229 (1): 75-84.

[11] Nadizadeh A, Nasab H H. Solving the dynamic capacitated location-routing problem with fuzzy demands by hybrid heuristic algorithm [J]. European Journal of Operational Research, 2014, 238 (2): 458-470.

[12] Zarandi M H F, Hemmati A, Davari S, et al. Capacitated location-routing problem with time windows under uncertainty [J]. Knowledge-Based Systems, 2013, 37: 480-489.

[13] Zhang H, Zhang Q, Ma L, et al. A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows [J]. Information Sciences, 2019, 490: 166-190.

[14] Bidar M, Kanan H R, Mouhoub M, et al. Mushroom Reproduction Optimization (MRO): a novel nature-inspired evolutionary algorithm [C]// 2018 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2018: 1-10.

[15] Wang H, Du L, Ma S. Multi-objective open location-routing model with split delivery for optimized relief distribution in post-earthquake [J]. Transportation Research Part E: Logistics and Transportation Review, 2014, 69: 160-179.

[16] 代颖, 马祖军, 朱道立. 震后应急物资配送的模糊动态定位—路径问题 [J]. 管理科学学报, 2012 (07): 64-74. (Dai Ying, Ma Zujun, Zhu Daoli. Fuzzy dynamic location-routing problem of emergency goods distribution in post-earthquake [J]. Journal of Management Sciences in China, 2012 (07): 64-74.)

[17] 马祖军, 代颖, 李双琳. 带限制期的震后应急物资配送模糊多目标开放式定位-路径问题 [J]. 系统管理学报, 2014, 23 (5): 658-667. (Ma Zujun, Dai Ying, Li Shuanglin. Fuzzy multi-objective open location-routing problem with deadlines in post-earthquake relief deliveries [J]. Journal of Systems & Management, 2014, 23 (5): 658-667.)

[18] 王海军, 杜丽敬, 马士华. 震后应急物流系统中双目标开放式选址-路径问题模型与算法研究 [J]. 管理工程学报, 2016, 30 (02): 108-115. (Wang Haijun, Du Lijing, Ma

Shihua. Dual-objective open location-routing problem in the post-earthquake emergency logistics system: model and algorithm study [J]. Journal of Industrial Engineering and Engineering Management, 2016, 30 (02): 108-115.)

[19] Yu X, Zhou Y, Liu X F. A novel hybrid genetic algorithm for the location routing problem with tight capacity constraints [J]. Applied Soft Computing, 2019, 85: 105760.

[20] Mauricio G C R R, Celso C. OPTIMIZATION BY GRASP: Greedy Randomized Adaptive Search Procedures [M]. SPRINGER, 2018.

[21] Zadeh L A. Fuzzy sets as a basis for a theory of possibility [J]. Fuzzy sets and systems, 1978, 1 (1): 3-28.

[22] Zadeh L A. The concept of a linguistic variable and its application to approximate reasoning—I [J]. Information sciences, 1975, 8 (3): 199-249.

[23] A. Kaufmann. Introduction to the Theory of Fuzzy Subsets [M]. Academic Press, New York, 1975.

[24] Puljić K, Manger R. Comparison of eight evolutionary crossover operators for the vehicle routing problem [J]. Mathematical Communications, 2013, 18 (2): 359-375.

[25] Christian Prins, Caroline Prodhon, Roberto Wolfler Calvo. Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking [J]. 4or A Quarterly Journal of Operations Research, 4 (3): 221-238.

[26] Prins C, Prodhon C, Calvo R W. A Memetic Algorithm with Population Management (MA|PM) for the Capacitated Location-Routing Problem [J]. 2006.

[27] Prins C, Prodhon C, Ruiz A, et al. Solving the Capacitated Location-Routing Problem by a Cooperative Lagrangean Relaxation-Granular Tabu Search Heuristic [J]. Transportation Science, 2007, 41 (4): p. 470-483.

[28] John Willmer Escobar, Rodrigo Linfati, Paolo Toth. A two-phase hybrid heuristic algorithm for the capacitated location-routing problem [J]. Computers & operations research, 2013, 40 (1): 70-79.

Note: Figure translations are in progress. See original paper for figures.