Modeling and Solution Algorithm for Fourth-Party Logistics Routing Problem Based on Service Composition

Xiaochuan Min\textsuperscript{a}, Fanchao Meng\textsuperscript{b}, Dianhui Chu and Lei Wang

School of Computer Science and Technology, Harbin Institute of Technology at Weihai, Weihai, 264209, China

\textsuperscript{a}18863282799@163.com, \textsuperscript{b}fcmeng@hit.edu.cn

Abstract. Aiming at the problems of limited transportation resources and high transportation costs of a single third-party logistics supplier, our research proposes a route planning model based on service combination from the perspective of fourth-party logistics, which aims to find the route with minimum total cost under the premise of meeting customer needs. In order to improve solution efficiency, a specific genetic algorithm is designed. The algorithm adopts backtracking idea and adaptive operations, which can avoid infeasible solution caused by strict constraints, while reducing the time required for crossover and mutation operations to repair infeasible solutions. In addition, Dijkstra-based heuristic algorithm, traditional genetic algorithm and traditional ant colony algorithm are implemented. The comparison of computational experiments on different scales instances shows that the specific genetic algorithm can obtain the optimal solution in small-scale instances, and the solution quality in large-scale problems is better than other algorithms.

1. Introduction

With the rapid development of e-commerce and intensification of the competition in logistics market, people are increasingly demanding for logistics services, while the traditional third-party logistics (3PL) can hardly meet the current market demand due to problems like lack of collaboration and integration capabilities [1]. The fourth-party logistics (4PL) has attracted more attention because the ability to integrate different 3PL resources and solve tasks that can’t be accomplished by a 3PL alone.

Research on the fourth-party logistics routing problem (4PLRP) has been carried out with the emergence of 4PL [2]. In previous studies, most of them only considered the single service mode of 3PL suppliers, ignoring the solution of combining multiple service modes of different 3PL suppliers to complete the transportation task. Therefore, this paper considers the route planning, 3PL supplier selection and combination of service modes comprehensively.

This paper establishes a fourth-party logistics route planning model based on service composition (4PLRPS), aiming at the difficulty of searching feasible solutions caused by strict constraints, proposes a specific genetic algorithm with backtracking idea and adaptive crossover and mutation operations. This paper implements the Dijkstra-based heuristic algorithm, genetic algorithm and ant colony algorithm, and proves the effectiveness of model and superiority of algorithms by experiments.

The remainder of this paper is organized as follows: Section 2 introduces the related work. Section 3 presents the problem description and formulation of the proposed model. Section 4 describes the specific genetic algorithms in detail. In Section 5, computational experiments are used to investigate the performance of proposed algorithms. Finally, conclusions are presented in Section 6.
2. Related work
Routing problem is the key issue for 4PL. In previous studies, some of them ignored the characteristics of different transportation modes with different economic benefits, and only focused on the routes optimization and 3PL supplier selection. Huang et al. [1] established a fuzzy program model and designed a two-step genetic algorithm to find an approximate optimal solution. Ren [3] designed an improved ant colony algorithm to solve a 4PL double-objective path integration optimization problem. Min [4] proposed a 4PLRPU model based on uncertainty theory and designed an improved genetic algorithm. Liu et al. [5] studied the 4PL optimization problem from the perspective of scheduling.

Some studies failed to consider the service quality of different 3PL suppliers from the perspective of multimodal transport. Wan et al. [6] constructed a mixed integer program model and designed a hybrid algorithm. Lozann [7] solved the shortest feasible path through sequential algorithm. Jiang et al. [8] adopted a hybrid cross entropy algorithm to solve multimodal transport scheme selection problem.

In addition, some studies failed to enforce the constraints strictly to obtain a feasible solution, but introduce penalty functions to reduce the complexity instead. Li et al. [9] designed a Max-Min ant system to solve the multi-task 4PLRP optimization model. Some studies only use traditional heuristic algorithm to solve the problem without introducing innovative ideas. For example, Lu et al. [10] established the chance constraint program model and used genetic algorithms to solve the problem.

3. Problem description
The relevant definitions of the 4PLRPS model will be explained in this section.

Definition 1 (Customer Task): a Customer Task can be expressed as \( \rho = (b, e, w, C, T, Q) \), where \( b \) and \( e \) represents the nodes of source and destination, \( w \) represents the greater of actual weight and volume weight of the goods, \( C, T \) and \( Q \) are the constraints of cost, time and service quality.

Definition 2 (Shared Service Network): this paper use \( SEN = (P, N, E) \) to express the Shared Service Network, where \( N = \{1, 2, \ldots, n\} \) is node set, the value of \( n \) is the number of nodes; \( P = \{1, 2, \ldots, m\} \) is 3PL supplier set, \( m \) is the number of suppliers; \( S_i = \{1, 2, \ldots, s_k\} \) is the set of service modes of supplier \( k \in P \), \( s_k \) is the number of service modes of supplier \( k \); \( E = \{(i, j, k, l) | i, j \in N, k \in P, l \in S_k\} \) is service line set, and \( e_{i,j}^l \) represents service line from node \( i \) to node \( j \) using the service mode \( l \) belonging to supplier \( k \).

Definition 3 (Task Route): A Task Route refers to a sequence of nodes and service lines from source to destination of a customer task, we use \( r = \{v_1, e_{1,2}^{1,k}, v_2, \ldots, e_{n-1,n}^{k,j}, v_n\} \) to express it, where \( v_i \in N \) is the node \( i (i=1, 2, \ldots, n) \) of the task route, \( n \) is the number of the nodes passed. There are multiple task routes for a customer task, we use \( R_s \) to express the task route set for customer task \( \rho \).

To formulate the 4PLRPS proposed in this paper, parameters are defined as follows:
- \( d_{i,j} \): The distance between node \( i \) and node \( j \).
- \( d_{i,k} \): The distance between supplier \( k \) and \( k' \) when a transfer occurs inside node \( i \).
- \( e_{i,j}^l \): Transportation cost per unit weight per unit distance on service line \( e_{i,j}^l \).
- \( c_{i,j}^{l,k} \): Conversion cost per unit weight from service mode \( l \) owned by supplier \( k \) to service mode \( l' \) owned by supplier \( k' \) when a transfer occurs inside node \( i \).
- \( s_{i,j} \): Transportation speed of service mode \( l \) owned by supplier \( k \).
- \( s_{i,j,k,l} \): Conversion speed of service mode \( l \) owned by supplier \( k \) to \( l' \) owned by supplier \( k' \).
- \( q_{i,j}^{l,k} \): Quality of the service mode \( l \) owned by supplier \( k \) between node \( i \) and node \( j \).
- \( x_{i,j}^{l,k}(r) \): 0-1 variables that represent whether service line \( e_{i,j}^{l,k} \) belongs to task route \( r \).
- \( y_{i,j}^{l,k,l'}(r) \): 0-1 variables that represent whether node \( i \) belongs to task route \( r \) and whether transfer occurs from service mode \( l \) owned by supplier \( k \) to service mode \( l' \) owned by supplier \( k' \).
The goal of 4PLRPS model is to find a task path \( r \) which satisfies constraints and minimizes total service cost. With above definition of variables, the mathematical model can be built up as follows:

\[
\min C(r) = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} w_{ik} \cdot d_{ik} \cdot c_{ij} \cdot x_{ij}^{k,l}(r) + \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} \sum_{l=1}^{n} w_{ik} \cdot d_{ik} \cdot c_{ij} \cdot y_{ij}^{k,l}(r)
\]

\[
s.t.
T(r) = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} d_{ik} / s_{ikj} \cdot x_{ij}^{k,l}(r) + \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} \sum_{l=1}^{n} d_{ik} / s_{ikj} \cdot y_{ij}^{k,l}(r) \leq T
\]

\[
Q(r) = \frac{1}{n_r - 1} \sum_{i=1}^{n_r} \sum_{j=1}^{m} \sum_{k=1}^{l} d_{ik} / s_{ikj} \cdot x_{ij}^{k,l}(r) \geq Q
\]

\[
x_{ij}^{k,l}(r) = \begin{cases} 1 & i \in N; j \in N; k \in P; l \in S_k \\ 0 & \text{else} \end{cases}
\]

\[
y_{ij}^{k,l}(r) = \begin{cases} 1 & i \in N; k \in P; l \in S_k; k' \in P; l' \in S_k \\ 0 & \text{else} \end{cases}
\]

In the formulation, equation (1) is the objective function, i.e. minimizes the total service cost, including transportation costs and conversion costs. Equation (2) means the service time including transportation time and conversion time should not be more than the time constraint \( T \). Equation (3) represents the average service quality should be greater than the quality constraint \( Q \). Equation (4) and Equation (5) are decision variables. Equation (6) is used to ensure selected task route is feasible.

![Figure 1. Shared Service Network diagram](image1)

![Figure 2. Task Route diagram](image2)

Figure 1 depicts a Shared Service Network consisting of 5 nodes and 3 suppliers. Customer Task \( r=1,(5,0.02,12,20,2) \). In the figure, hollow circles represent nodes and solid circles represent suppliers. The line between hollow circles represents service line. 3,(14,22, 3) marked in red indicates the transportation speed of this service line is 14, transportation cost is 22 and quality is 3. The line between solid circles represents the internal transfer inside node. The red circle (3, 2, 1) indicates the distance between supplier 1 and 2 is 3, conversion speed is 2 and conversion cost per unit weight is 1.

According to various data in Figure 1, the total service cost, service time and service quality of all
task routes from source to destination can be calculated. One of the task route found in this problem \( r = \{ 1, e_{1,2}, 2, e_{2,3}, 3, e_{3,5}, 5 \} \) is shown in figure 2. In this task route, \( T(r) = 16.8, C(r) = 9.88, Q(r) = 3 \).

4. Specific genetic algorithm design
The 4PLRPS is a difficult constrained combinational optimization problem. Infeasible solution that is not a route or doesn’t satisfy constraint is the key factors impacting search efficiency. Therefore, the specific genetic algorithm with backtracking idea and adaptive operations is proposed to avoid infeasible solution during chromosome initialization and to reduce the running time of repairing infeasible solution during crossover and mutation. The specific genetic algorithm is described below.

4.1. Chromosome coding.
A chromosome based on three-column variable length coding rule can be defined as \( P = (A, B, C) \), where \( A = \{ v_1, v_2, \ldots, v_{n-1}, v_n \} \) is node sequence set used to store the nodes passing through the task route; \( B = \{ p_1, \ldots, p_{n-1} \} \) is supplier sequence set, store supplier selected between two nodes; \( C = \{ s_1, \ldots, s_{n-1} \} \) is service mode sequence set which store service mode selected after determining supplier on task route. The chromosome coding structure is shown in figure 3, indicating the task route selects service mode 1 of supplier 2 between node 1 and node 3, and so on.

![Figure 3. Schematic diagram of chromosome coding structure](image)

4.2. Chromosome initialization.
Backtracking strategy is introduced during chromosome initialization to avoid infeasible solution that is not a route from source to destination or doesn’t satisfy constraint. The rules are defined as follows.

```
Algorithm 1: Chromosome initialization algorithm
Input: Customer Task \( \rho \)
Output: \( P = (A, B, C) \)
1: push source node into Stack \( N \);
2: flag ← true;
3: outer: while \( TOP(N) \neq \) destination node do
4: cNode ← PEEK(N);
5: if flag && (cNode)\( \in (A(cNode)) \) \&\& (cNode)\( \notin \) null then
6: select node \( n \) from \( (A(cNode)) \) randomly; PUSH(n, N);
7: else pNode ← PEEK(N); POP(N);
8: cNode ← PEEK(N); add pNode into \( A(cNode) \);
9: select node \( n \) from \( (A(cNode)) \) randomly; PUSH(n, N);
10: end if
11: while \( B(cNode,n) \neq \) null then
12: if flag then
13: select supplier \( v \) from \( (B(cNode,n)) \) randomly; PUSH(v, V);
14: else pSup ← PEEK(V); POP(V);
15: cSup ← PEEK(V); add pSup into \( B(cSup) \);
16: select supplier \( v \) from \( (B(cNode,n)) \) randomly; PUSH(v, V);
17: end if
18: while \( C(cSup) \neq \) null then
19: if flag then
20: select service mode \( s \) from \( (C(cSup)) \) randomly; PUSH(s, S);
21: else pSer ← PEEK(S); POP(S); C(cSer) ← PEEK(S); add pSer into \( C(cSer) \);
22: select service mode \( s \) from \( (C(cSer)) \) randomly; PUSH(s, S);
23: end if
24: if time <= cTime \( (N, V, S) \neq \) T && equality <= cQuality \( (N, V, S) \neq \) Q then
25: flag ← true; continue outer
26: else flag ← false
27: end if
28: end while
29: end while
30: end while
31: A ← N; B ← V; C ← S
32: return \( P = (A, B, C) \)
```
Define three stacks $N$, $V$ and $S$ to store the sequence of nodes, suppliers and service modes selected randomly. $A(n)$ represents adjacent nodes set of node $n$, $A'(n)$ represents infeasible nodes set during backtracking. $B(n_1,n_2)$ is defined as the set of all suppliers between node $n_1$ and $n_2$, $B'(n_1,n_2)$ means infeasible supplier set. Definition $C(s)$ represents the set of all service modes owned by supplier $s$ and $C(s)$ represents infeasible service modes set selected by the supplier $s$. Function $cTime(N,V,S)$ and $cQuality(N,V,S)$ are used to calculate service time and service quality of current task route.

4.3. Genetic operations.

The specific genetic algorithm proposed for this problem has the same basic structure as a traditional genetic algorithm. The details of specific genetic algorithm are as follows.

Fitness function: The mapping function is defined as $F(x) = f(x) - f_{min} + r/f_{max} - f_{min} + r$, where $x$ is a chromosome, $F(x)$ is the fitness function, $f(x)$ is the objective function, $f_{min}$ and $f_{max}$ are the minimum and maximum objective function values in the population, and $r \in (0,1)$ is used to prevent division by zero in the computation process. In the proposed algorithm, we set $r=1$.

Selection: We choose chromosomes by the roulette-wheel selection strategy to expedite search. Meanwhile, elite strategy is employed to protect superior chromosomes from damage.

Crossover: First, generate a random cut point for each of two chromosomes. To avoid the invalid crossover caused by splitting of source and destination node, the range of cut point is $[2, L-2]$, where $L$ is length of node sequence set. Then, exchange the segments to produce two new chromosomes. The chromosome is recombined in the way shown in figure 4 if node duplication occurs. Then, determine whether new chromosomes are feasible. Adjust the cut point if both are infeasible. If both are feasible, the one with lower service cost is retained.

Mutation: Mutation doesn’t occur in node sequence. First, generate mutation position according to the length of supplier sequence. Then, select supplier randomly to replace original one to produce new chromosome and determine whether it’s feasible. If infeasible, revoke the change and select a service method randomly at the mutation position to replace original one, then determine whether it’s feasible. If still infeasible, undo changes and repeat the above steps after adjusting the mutation position.

5. Computational experiments

In this section, we generate different test instances firstly, then, analyze the impact of parameters on the algorithm performance so as to determine the optimal parameter combination. Finally, we design comparative experiments to analyze the performance of different algorithms based on the solution results to verify the effectiveness of the proposed algorithm.

Ant colony algorithm has advantages of strong robustness, especially for discrete path optimization problems [3]. The genetic algorithm is one of the most widely used to solve constrained optimization problems [4]. Therefore, this paper implements ant colony algorithm (AC) and genetic algorithm (GA). The 4PLRPS is transformed into a shortest path problem with time and quality constraints by transforming the shared service network into a virtual transportation network, then solved by Dijkstra-based heuristic algorithm (DH). Dijkstra algorithm is used to find the shortest route for service cost without considering constraints. On this basis, heuristic algorithm is used to adjust to meet constraints.

Experiment is carried out in a Lenovo computer, experimental environment is configured as
follows: the processor is Intel (R) Core (TM) i5-4690 CPU @ 3.50GHz 3.50GHz, the memory is set as 8.0GB. All algorithms are implemented in Java programming language and run in eclipse development tool.

5.1. Test instances.

As there is no complete instance of 4PLRPS in related literature, we design four rules in the process of generating test instances according to references [1] [9] [10]. The description will be given below.

Node numbering rules: Nodes generated randomly are numbered from 1 to n according to the distance from coordinate origin, where 1 is the source and n is the destination.

Edge generating rules: If distance between two nodes isn’t greater than \( u*D_n \), edge is generated with 70% probability, otherwise there is no edge, where \( u \) is threshold multiple depending on nodes number, \( D_n = \sqrt{\frac{\text{ucv}}{n^2}} \), here, \( \text{ucv} \) represents the upper limit of coordinate and \( n \) represents nodes number.

Edge supplier generating rules: Assuming the number of suppliers is \( m \), for each edge, all \( m \) suppliers are generated with 20% probability, \([0.8*m]\) suppliers are randomly generated with a probability of 50% and less than \([0.8*m]\) suppliers are randomly generated with 30% probability.

Service mode owned by supplier generating rules: Assuming the number of service mode is \( s \), for each supplier, all \( s \) service modes are generated with 20% probability, \([0.8*s]\) service modes are randomly generated with 50% probability and less than \([0.8*s]\) are generated with 30% probability. In addition, different prices and speeds are set for the same service mode belonging to different suppliers.

5.2. Parameters of algorithms.

Parameter setting is important in evolutionary algorithm and different settings will affect performance of algorithm [11]. For GA, it’s concluded the solution quality is superior when crossover probability is 0.8 and mutation probability is 0.3 through tests in different scale instances, so it’s set as the final parameter. Select different instances (\( n,v,s \)), where \( n,v,s \) is the number of nodes, suppliers and service modes. The curve of cost change with the number of generation (\( NG \)) is shown in Figure 5. It can be seen that the algorithm has obtained a better solution when \( NG=100 \), so set it as final parameter. Select other instances and record broken line of cost with population number (\( NP \)). The results are shown in Figure 6. It can be seen the solution is optimal when \( NP=200 \) for different instances, so it’s determined.

![Figure 5. Tendency of cost with number of generation](image1)

![Figure 6. Broken line of cost with number of population](image2)

For AC, we also select different instances and carry out many experiments under condition of large number of ant and iterations. When pheromone heuristic factor=4, expectation heuristic factor=3 and pheromone residual factor=0.2, the quality of solution is superior, so set it as final parameter combination.
5.3. Test results.
Considering Dijkstra algorithm can only find solution in small-scale problems, we design two comparative experiments of different scale. Since the specific genetic algorithm (SGA) takes into account infeasible solutions due to strict constraints, we test loose and strict constraints in experiment.

We set 9 groups of 10 nodes small-scale instances to test algorithms and the results are shown in Table 1. In the table, m and s represents the number of suppliers and service modes, and T represents the constraint of time. The optimal solution obtained by running each algorithm 10 times in each instance is recorded. The column contains ‘-’ means the algorithm failed to obtain a feasible solution.

Table 1. Comparison table of four algorithms in small-scale instances

| m | s | T   | Dijkstra | SGA | GA | AC  |
|---|---|-----|----------|-----|----|-----|
| 1 | 3 | 2   | 20       | 30.732 | 30.732 | 30.732 | 32.320 |
| 2 | 3 | 15  | 32.230   | 32.230 | 32.230 | 32.230 |
| 3 | 3 | 25  | 16.042   | 16.042 | 16.042 | 16.042 |
| 4 | 3 | 10  | -        | 41.679 | 41.679 | 61.783 |
| 5 | 3 | 20  | 13.730   | 13.730 | 13.730 | 13.730 |
| 6 | 3 | 80  | 28.689   | 24.696 | 26.162 | 25.162 |
| 7 | 4 | 30  | 32.890   | 32.890 | 32.890 | 32.890 |
| 8 | 4 | 20  | 50.761   | 50.761 | 63.298 |
| 9 | 4 | 15  | 16.304   | 16.304 | 16.304 | 16.304 |
| 10 | 4 | 5  | 20.096   | 23.941 | 20.096 | 26.809 |
| 11 | 4 | 10  | 29.119   | 22.030 | 22.030 | 22.082 |
| 12 | 4 | 10  | 45.678   | 45.662 | 50.415 |
| 13 | 4 | 2   | 24.153   | 24.153 | 24.153 |
| 14 | 5 | 10  | -        | 36.887 | 36.887 | 36.887 |
| 15 | 5 | 30  | 13.501   | 13.501 | 13.501 | 26.279 |
| 16 | 5 | 15  | 22.752   | 22.752 | 22.752 |
| 17 | 5 | 4  | 25      | 13.970 | 13.970 | 13.970 |
| 18 | 5 | 10  | -        | 23.941 | 23.941 | 27.842 |

Large-scale instances are set up to test SGA, GA and AC. Each algorithm runs each instance 10 times. Table 2 shows the best solution (BEST), average value (AVE), the worst solution (BAD) and running time (TIME) of three algorithms. The ‘-’ in table 2 represents the number of nodes.

Table 2. Comparison table of three algorithms in large-scale instances

| n  | m  | s  | T     | BEST | AVE | BAD | TIME (ms) | BEST | AVE | BAD | TIME (ms) |
|----|----|----|-------|------|-----|-----|----------|------|-----|-----|----------|
| 30 | 5  | 20 | 32.888| 49.386| 61.675| 81.273| 52.876 | 53.798| 58.800| 53.798| 58.800 |
| 50 | 6  | 20 | 32.888| 49.386| 61.675| 81.273| 52.876 | 53.798| 58.800| 53.798| 58.800 |
| 80 | 6  | 20 | 32.888| 49.386| 61.675| 81.273| 52.876 | 53.798| 58.800| 53.798| 58.800 |
| 100| 5  | 20 | 32.888| 49.386| 61.675| 81.273| 52.876 | 53.798| 58.800| 53.798| 58.800 |
| 120| 5  | 20 | 32.888| 49.386| 61.675| 81.273| 52.876 | 53.798| 58.800| 53.798| 58.800 |
| 150| 5  | 20 | 32.888| 49.386| 61.675| 81.273| 52.876 | 53.798| 58.800| 53.798| 58.800 |

The best result for each instance is marked as bold and underlined.
In table 1, DH and AC obtained 9 best solutions respectively, SGA obtained 16 best solutions and GA obtained all best solutions. The results show in small-scale instances, SGA and GA can find the same best solution as DH when constraints are loose, while AC is unstable. When constraints become stricter, the adjustment strategy of DH reduces its performance or even fails to obtain feasible solution. At this time, SGA and GA are better than the other two algorithms. In table 2, SGA obtained 32 best solutions and 23 average solutions, while AC obtained 4 best solutions and 12 average solutions. It can be seen the overall performance of SGA is better than other algorithms when problem scale is large. Although GA can obtain a solution similar to SGA sometimes, the solution quality becomes unstable as problem scale increases. Sometimes AC is better than the other two algorithms, but it takes more running time, and solution quality isn’t ideal under the condition of strict constraints.

6. Conclusion
This research considers route planning, supplier selection and service mode combination comprehensively. A mathematical model with minimum total cost as goal is established, and a specific genetic algorithm is designed, then, compare it with Dijkstra-based heuristic algorithm, traditional genetic algorithm and ant colony algorithm through different instances. The comparison results show these algorithms have good effects when the problem scale is small and the constraint conditions are loose, but with increase of problem scale and strict constraints, the adaptive genetic algorithm shows its superiority. In summary, the algorithm in this paper can solve the fourth-party logistics routing problem based on the combination of service modes with strict constraints easily, and it also has reference value for combined optimization of transportation modes in multimodal transport.

Acknowledgements
This work is supported by the National Key R&D Program of China under grant 2018YFB1403104.

References
[1] Huang M, Cui Y, Yang S, et al. Fourth party logistics routing problem with fuzzy duration time [J]. International Journal of Production Economics, 2013, 145(1): 107-116.
[2] Cui Y, Huang M, Yang S, et al. Fourth party logistics routing problem model with fuzzy duration time and cost discount[J]. Knowledge-Based Systems, 2013, 50(Complete): 714-721.
[3] Liang R, Min H, Hongfeng W, et al. Integrated optimization of multi-objective routing problem in fourth party logistics [J]. Complex Systems and Complexity Science, 2018, 15(01): 62-67.
[4] Min H, Liang R, Loo H L, et al. Model and algorithm for 4PLRP with uncertain delivery time [J]. Information Sciences, 2016, 330: 211-225.
[5] Liu Q, Zhang C Y, Zhu K R, et al. Novel multi-objective resource allocation and activity scheduling for fourth party logistics[J]. Computers & Operations Research, 2014, 44: 42-51.
[6] Jie W, Shuang W. Multi-objective multimodal transportation path selection based on hybrid algorithm[J]. Journal of Tianjin University (Science and Technology), 2019, 52(03): 285-292.
[7] LOZANO M, STORCHI G. Shortest viable path algorithm in multimodal networks[J]. Transportation Research Part A, 2001, 35(3): 225-241.
[8] Jiang Y, Zhang X, Wang Y. A cross-entropy method for solving selection of multimodal transportation scheme[J]. Journal of Transportation Systems Engineering and Information Technology, 2012, 12(5): 20-25.
[9] Li G, Huang M. The fourth logistics routing optimization algorithm considering multiple transportation modes[J]. Computer Engineering, 2015, 41(03): 273-277.
[10] Lu F, Chen W, Bi H, et al. Research on 4PL routing problem considering stochastic demand and multiple transportation modes [J/OL]. Computer Integrated Manufacturing Systems, 2019(08): 1-25.
[11] Liu Q, Li T. Research on benefits distribution of fourth party logistics based on non-profit organization [A]. 1st International Conference on Intelligent Robotics and Applications, ICIRA2008[C], Wuhan, China, 2008:530-538.