IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 7, 2019, accepted March 29, 2019, date of publication April 24, 2019, date of current version May 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2913003

Route Network Design of Community Shuttle for Metro Stations Through Genetic Algorithm Optimization

JIE XIONG[1], BIAO CHEN[1], YANYAN CHEN[1], YANG JIANG[2], AND YAO LU[1]

[1]Beijing Key Laboratory of Traffic Engineering, Beijing University of Technology, Beijing 100124, China
[2]School of Management, Shenyang University of Technology, Shenyang 110870, China

Corresponding author: Yanyan Chen (cdyan@bjut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 71601006 and Grant 71801160, in part by the Science and Technology Development Foundation of Beijing Municipal Education Commission under Grant KM201710005030, and in part by the MOE (Ministry of Education in China) Liberal arts and Social Sciences Foundation under Grant 18YJC630158.

ABSTRACT This paper investigates an issue for optimizing community shuttle network linked with metro stations. Considering a set of trip demands originating from several bus stops within a large community area, a mixed integer optimization model is formulated to obtain the optimized shuttle route network and the frequency of each route simultaneously, to minimize the total transit system cost, including user and supplier costs. To solve the problem, a solution framework consisting of three primary components is proposed: an initial route network scheme is developed to ensure that any generated network is feasible; a network analysis procedure is performed that assigns the demand of each stop to a set of paths on the generated network according to a certain proportion; a heuristic algorithm is proposed to optimize the service frequency of each route; a genetic algorithm consisting of a set of specifically designed operators is proposed to guide the solution population evolving process. A real-life study is tested successfully using the proposed solution framework. The related sensitivity analysis and impacts of some important variables on the solution are performed.

INDEX TERMS Community shuttle, route network optimization, metro station, genetic algorithm.

I. INTRODUCTION

Owing to the rapid development of metros in urban and suburban areas currently, bus suppliers are required to adjust the bus route network constantly to co-operate with the metro smoothly, and more community shuttle routes linked with metro services have appeared, especially in the suburbs. However, many community shuttle routes are designed empirically and lack of theoretical foundation. Therefore, we focus on optimizing the community shuttle network from the theoretical level to improve their feeding role and achieve the complementary advantages of these two transit modes, thus enhancing the service level of the transit microcirculation system.

In the last few decades, many scholars have performed numerous investigations regarding the optimal transit network design. The route network and frequency setting problems are two primary problems in transit system optimization [1], [2]. Some have optimized the bus route network considering both problems simultaneously [3]–[6]. For example, [5] developed a solution for the optimal routing design problem to minimize the total cost, including user and supplier costs; [4] investigated the transit network of a suburb in Hong Kong to address both problems. They formulated a mixed integer optimization model, and obtained the optimal layout of the route network and the optimal frequency setting plan simultaneously.

In terms of model formulation, the factors and determination of the objective function considered by the researchers are different, such as maximizing the passenger attraction [7], minimizing the total travel time [8], maximizing the total scope of transit stops [9], and minimizing the route length [10]. Most studies had considered both user and supplier costs, and set their sum as the objective

The associate editor coordinating the review of this manuscript and approving it for publication was Walter Didimo.

function [4], [11], [12]. Particularly, [4] used the weighted mixed integer optimization model that expressed the diverse weights between passenger cost and supplier cost to represent the relationship between passengers and suppliers in different cases.

In terms of algorithm design, researchers have proposed numerous heuristic or metaheuristic algorithms to solve feeder bus network optimization problems, such as simulated annealing algorithm [13], genetic algorithm [4], [14], hybrid enhanced artificial bee colony algorithm [6], and a complex two-phase heuristic algorithm [11]. Reference [11] proposed a two-phase heuristic algorithm to optimize the route network design and frequency setting problems. Reference [4] proposed a genetic algorithm with two types of crossover operators and four types of mutation operators. Reference [6] proposed a hybrid enhanced artificial bee colony algorithm to solve the same problem.

This paper focuses on the community shuttle route network design problem to minimize the sum of the user and supplier costs. To solve the problem, we first developed a route network generation scheme to generate a feasible solution consisting of a set of shuttle routes randomly; and subsequently, a genetic algorithm (GA) is designed to guide the solution population evolving process and obtain an optimized solution. Our major contributions are the following: (1) in the model formulation, we stipulated that all the bus stops must be passed through by at least one route to provide a more convenient service for residents living in different locations in an area. (2) A new fleet size adjustment process is proposed to determine the fleet size of each route precisely by an analytical approach, considering both the overall fleet size constraint of the system and the vehicle capacity constraint. (3) a set of crossover, mutation, and repair operators are designed to enable genetic manipulation to solve the problem smoothly.

The remainder of this paper is organized as follows: Section 2 describes the community shuttle network optimization problem and a mixed integer optimization model is formulated. In Section 3, a solution generating method is addressed to ensure that any randomly generated route network passes through all the stops and passengers are assigned by a logit model. Further, a heuristic algorithm is proposed to adjust the total fleet size. Section 4 demonstrates the proposed genetic algorithm to solve the specific problem. A real-life example is presented in Section 5, along with the computational results and numerical analysis. Finally, conclusions and future work are presented in Section 6.

II. MODEL FORMULATION

A. PROBLEM DESCRIPTION

The problem is to design an optimized shuttle route network in a certain large community area to transport the residents living in different locations to the metro stations nearby more efficiently and conveniently, while considering the supplier cost of the bus operator. Owing to the specific characteristics of the community shuttle, the route can be shorter and slightly circuitous compared to the conventional bus routes. Therefore, we added a constraint that all the stops in the area must be passed through by at least one route to ensure the accessibility of the network, whereas the passengers' in-vehicle time may be increased by some unnecessary detours. Meanwhile, a high service frequency of a route will reduce the waiting cost of passengers, whereas the operator cost will increase with a larger fleet size. Consequently, it is necessary to solve the problem from the perspective of minimizing the total cost in the transit microcirculation system, including the passengers' in-vehicle cost, waiting cost, and supplier cost. By solving the problem, an optimized layout of the route network and the service frequency of each route can be obtained simultaneously.

Some assumptions for developing the model are as follows: (1) every OD trip in the area starts at a bus stop, and terminates at a metro station; (2) each route originates from a bus depot and terminates at a metro station, that is, each shuttle route is connected to at least one metro station; (3) all the routes are serviced by a homogenous fleet of vehicles, and the vehicle speed and dwell time at each stop are constant; (4) the numbers of passengers arriving at the bus stops are randomly distributed during the study period for each OD pair.

B. OBJECTIVE FUNCTION

The objective function in this study consists of two primary components: the user cost and the_supplier cost. It can be expressed as follows:

$$C_T = C_S + C_U \tag{1}$$

where C_T = the total cost; C_S = the supplier cost; C_U = the user cost.

The supplier cost is related to both the route and service frequency. Therefore, it can be formed as follows:

$$C_S = \gamma_S \cdot \sum_{k \in R} F_k \tag{2}$$

$$F_k = \frac{2\,(L + T_k \cdot N_k \cdot V_b)}{H_k \cdot V_b} \tag{3}$$

where γ_S = the cost of unit vehicle; R = the set of routes; F_k = the fleet size of route k, L = the length of route k (one way); T_k = the dwell time at the stops of route k; N_k = the number of stops on route k; V_b = the bus speed; H_k = the headway of route k.

The user cost can be further divided into two components: the in-vehicle cost C_I and the waiting cost C_W:

$$C_U = C_I + C_W \tag{4}$$

The in-vehicle cost is related to the route layout, and in this research, route-based trip assignment can be utilized since all the bus stops are covered by at least one route. However, for an OD pair, the route passes through the origin node (bus stop) may not via the real destination node (metro station) since more than one metro station is considered in our model. When this case happens, we assume that the passengers transfer at the metro station the route passing through to their real destination node. Therefore, the in-vehicle time on the metro needs to be considered when calculating the in-vehicle time on route k for this OD pair. The in-vehicle cost thus can be expressed as follows:

$$C_I = \sum_{i \in O} \sum_{j \in D} \frac{\gamma_I \cdot \sum_{k \in R} M_{ij}^k \cdot D_{ij}^k}{V_b} \tag{5}$$

$$M_{ij}^k = \frac{e^{-t_{Ik}}}{\sum_{k \in R_i} e^{-t_{Ik}}} \cdot M_{ij} \tag{6}$$

where $i, j =$ the origin and destination nodes of each trip (a bus stop or a metro station); $D_{ij}^k =$ the travel distance along route k from i to j; $M_{ij}^k =$ the number of passengers from i to j on route k; $M_{ij} =$ the total number of passengers from i to$j$; $\gamma_I =$ the in-vehicle time value of passengers; $t_{Ik} =$ the in-vehicle travel time on route k(maybe includes the in-vehicle time on the metro); $O, D =$ the set of all the origin and destination nodes; $R_i =$ the set of routes passing through i.

The waiting cost C_W is only related to the bus headway, and can be expressed as follows:

$$C_W = \gamma_W \cdot \sum_{i \in O} \sum_{j \in D} \sum_{k \in R} \frac{H_k}{2} \cdot M_{ij}^k \tag{7}$$

where $\gamma_W =$ the waiting time of passengers.

As stated above, only C_W and C_S are related to the headway. By taking the first derivative of C_T on both sides with respect to H_k and setting it to 0, we can obtain the corresponding headway H_{k1} as follows:

$$H_{k1} = \sqrt{\frac{2 \cdot \gamma_S \cdot (L_k + T_k \cdot N_k \cdot V_b)}{\gamma_W \cdot V_b \cdot \sum_{i \in O} \sum_{j \in D} M_{ij}^k}} \tag{8}$$

The H_{k1} obtained through the formula above ensures the minimized total cost for each route. However, it neglects the segment loading factor, that is, the loading on some route segments may exceed the vehicle capacity if we used the obtained H_{k1}. For solving this problem, we use the method proposed by [5] and [15], and set an upper limit of the headway of route k, H_{k2}, as follows:

$$H_{k2} = \frac{P}{\max\left\{d_{i1i2}^k \cdot x_{i1i2}^k\right\}} \tag{9}$$

where $P =$ the vehicle capacity; $d_{i1i2}^k =$ the segment loading of route k between node $i1$ and $i2$ per hour. If $i1$ and $i2$ are adjacent stops on route k, $x_{i1i2}^k = 1$; otherwise, $x_{i1i2}^k = 0$. The fleet size of route k is determined by (10):

$$H_k = \min\left\{H_{k1}, H_{k2}\right\} \tag{10}$$

C. CONSTRAINTS

The constraints of the problem are given as follows:

$$\sum_{k \in R} \sum_{j \in S} x_{0j}^k = 1 \tag{11}$$

$$\sum_{k \in R} \sum_{i \in S} \sum_{s \in S0} x_{is}^k = 1 \tag{12}$$

$$\sum_{k \in R} \sum_{i \in S} x_{ij}^k = \sum_{k \in R} \sum_{i \in S} x_{ji}^k \; (\forall j \in S) \tag{13}$$

$$\sum_{j \in S} x_{ij}^k \leq 1 (\forall i \in S, \forall k \in R) \tag{14}$$

$$\sum_{k \in R} \sum_{j \in S} x_{ij}^k \geq 1 (\forall i \in S) \tag{15}$$

$$L_{min} \leq L_k \leq L_{max} (\forall k \in R) \tag{16}$$

$$\sum_{k \in R} F_k \leq F_{max} \tag{17}$$

Constraints (11) and (12) are route constraints; they constraint that the origin of each route is the depot, and the destination is one of the metro stations. Constraint (13) ensures that any node on an available route has one preceding node and one following node (except for the origin and destination nodes). Constraint (14) prevents a cyclic route or sub-route in the solution. Constraint (15) ensures that all the stops exist in the solution. Constraint (16) ensures that the length of each route is within a reasonable interval. Constraint (17) ensures that the total fleet size cannot exceed its upper limit.

III. ROUTE NETWORK GENERATION AND ANALYSIS PROCEDURE

A. ROUTE NETWORK GENERATION

The route network generation is constructed to yield a feasible route network satisfying the constraint that all the stops appear in the network. Additionally, a feasible solution consisting of a set of feasible routes is randomly generated based on the topological structure of the road network. Let S_0, S, M indicate the depot, set of bus stops, and set of metro stations, respectively. We define *Solution* as the current solution found, *Route* as the current route generated, and *CS* as the set of bus stops that have been covered. We define $D = \min\{\max_{D_m}, m \in M\}$, and $\max_{D_m}$ is the maximal distance from the metro station m to the nodes that can be connected to m directly on the topological structure of the road network. The specific steps are as Algorithm 1:

When a set of feasible routes is generated using the above method, a trip assignment method is utilized to assign each OD pair to a set of routes passing through the origin node by (5). Based on it, the optimal fleet size and the corresponding headway of each route are determined by (8). However, the total fleet size constraint is not considered during this process. That means the total fleet size may exceed its upper limit when the optimal headways corresponding to the minimized total cost of the routes are applied. For solving this problem, a fleet size adjustment process is developed to ensure that the total fleet size constraint can be satisfied with the consideration of the segment loading factor.

B. FLEET SIZE ADJUSTMENT

As stated above, when the total fleet size of the route network is given, the sum of the optimal fleet size corresponding

TABLE 1. Route network generation algorithm.

Algorithm 1: Route Network Generation

1: Input: max_D_m ← set of maximal distance from the metro station m to the nodes;
D← minimum distance in max_D_m;
S_0 ← bus depot;
CS ← set of bus stops that have been covered;
M ← set of metro stations;
$Next\_node$ ←set of possible nodes the current route can travel to from node r;
$d_{r,s}$ ←the shortest distance from r to s on the topological network, $s \in S \backslash CS$;
$path$ ← path from node r (not included) to metro station m;
$Next\_node$←Φ; $Route$←S_0; CS←Φ; n←1;

2: **Repeat**

3: r ← the last node in $Route$;

4: **for** each element in s **do**

5: **if** $d_{o,r} + d_{r,s} \leq L_{max} - D$ **then**
$Next\_node$←$Next\_node + s$;

6: **end if**

7: **end for**

8: **if** $Next\_node = \Phi$ **then**

9: Choose a metro station $m \in M$, $Route$ ← [$Route$, $path$]; $Solution$←$Solution$ + $Route$, $n \leftarrow n + 1$, $Route \leftarrow S_0$;

10: **else**

11: Select a $node$ from $Next\_node$ randomly and generate the shortest path s_1 from r (not included) to the $node$ (included), $Route$← [$Route$, s_1], CS←CS+s_1;

12: **end if**

13: **Until** $CS = S$;

to the minimized total cost of each route may exceed this given limit. When this situation occurs, a fleet size adjustment process needs to be carried out to make the total fleet size constraint be satisfied by reducing the fleet size of some certain routes. This process may cause overloading on some segments, especially for the routes with '$H_k = H_{k2}$'. To solve this problem, we added an overloading penalty to the original objective function. Before solving the penalty, we define a penalty function of the vehicle capacity, $g(x) = \omega(\alpha^x - 1)$, where ω and α are coefficients ($\omega > 0$, $\alpha > 1$); x is the segment loading. By adjusting ω and α, $g(x)$ can become close to 0 when $0 \leq x \leq P$, and increase significantly when $x > P$. Hence, the overloading penalty can be described as:

$$C_P = \sum_{k \in R} \sum_{i1 \in O} \sum_{i2 \in O \cup D} g(d_{i1i2}^k \cdot H_k) \cdot x_{i1i2}^k \tag{18}$$

This process aims at minimizing the increase of the waiting cost and overloading penalty due to the reduction of the fleet size as much as possible. The detailed procedures are described as follows:

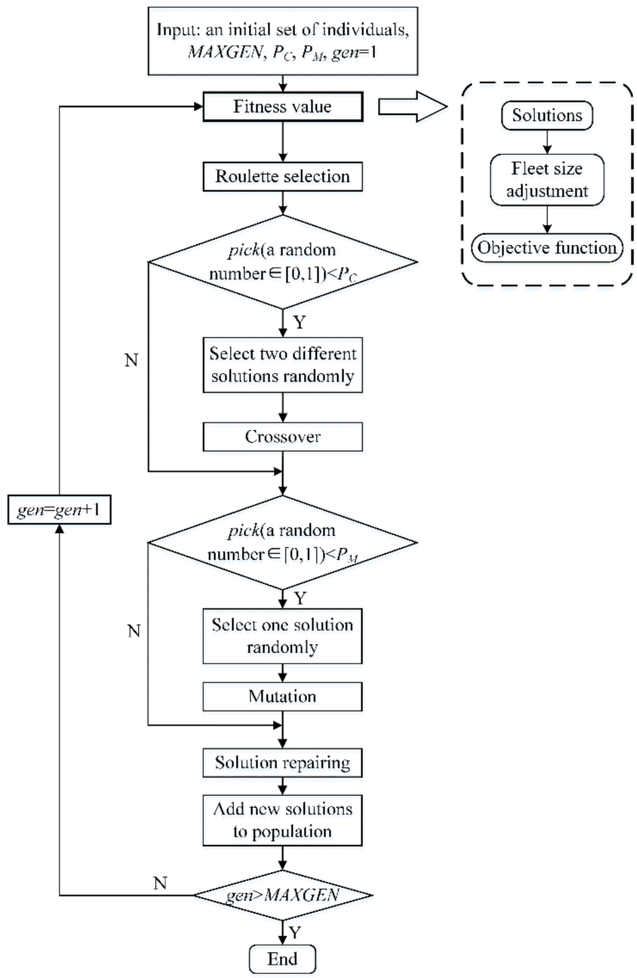


FIGURE 1. Genetic algorithm process.

IV. GENETIC ALGORITHM

In this section, we design a set of suitable operators to ensure that the GA can conduct smoothly for solving the specific problem. The input variables include an initial set of individuals, crossover probability (P_c), mutation probability (P_m), the maximum number of iterations (*MAXGEN*). Roulette selection is applied to select a sub-set of individuals according to the fitness functions. The parameters P_c, P_m control the probability of implementing the crossover and mutation operators. A solution repair process needs to be added after the crossover and mutation procedures to ensure all the new generated solutions feasible. The whole algorithm terminates when the current iteration (*gen*) exceeds *MAXGEN*. Detailed procedures are shown in Fig.1.

A. FITNESS FUNCTION AND SELECTION

In the iterative process of the GA, the algorithm selects the individuals with large fitness values to create new individuals continuously. Here we use the reciprocal of the objective function as the fitness function of the algorithm, that is, $f = 1/C_T$.

TABLE 2. Fleet size adjustment algorithm.

Algorithm 2: Fleet Size Adjustment

Input: $TF \leftarrow$ the total fleet size of a solution;
$max\_TF \leftarrow$ the upper limit of the total fleet size;
$F=(\boldsymbol{F_1}, \cdots, \boldsymbol{F_k}, \cdots, \boldsymbol{F_n}) \leftarrow$ the fleet size of each route;
$Adjusted\_Route \leftarrow$ the set of candidate routes of which the fleet size can be adjusted, and be initialized as an empty set;
$\Delta Cost = (\boldsymbol{\Delta C_W}, \boldsymbol{\Delta C_S}, \boldsymbol{\Delta C_P}) \leftarrow$ record the changing of some related costs due to the fleet size reduction, including the change of the waiting cost (positive), the overloading penalty (positive), and the supplier cost (negative).
Repeat
  if $TF > max\_TF$, **then**
    for $k \leftarrow 1$ to n, **do**
      $F_k' = F_k - 1$, and calculate the new headway H_k' according to (3)
      if H_k' can still satisfy the vehicle capacity constraint, **then**
        $Adjusted\_Route \leftarrow Adjusted\_Route + k$
      end if
    end for
    if $Adjusted\_Route$ is not an empty set, **then**
      for $m \leftarrow 1$ to $|Adjusted\_Route|$, **do**
        $\Delta Cost \leftarrow \Delta Cost + (\Delta C_W{}^m, \Delta C_S{}^m, \Delta C_P{}^m)$ //record the changing of the costs of the current route after removing a vehicle, and add it into $\boldsymbol{\Delta Cost}$
      end for
    else
      for $m \leftarrow 1$ to n, **do**
        $\Delta Cost \leftarrow \Delta Cost + (\Delta C_W{}^m, \Delta C_S{}^m, \Delta C_P{}^m)$
      end for
    end if
    Record route l: $(\Delta C_W{}^l, \Delta C_S{}^l, \Delta C_P{}^l) = \min(\Delta C_W, \Delta C_S, \Delta C_P)$
    $F_l \leftarrow F_l - 1$
  end if
  Update TF
Until $TF \leq max\_TF$

The selected probability of an individual is based on the fitness f. The probability of selecting the i^{th} individual, P_i, can be formulated as:

$$P_i = \frac{f_i}{\sum f_i}(i \in NIND) \tag{19}$$

where $NIND$ = the population size, and f_i = the fitness value of the i^{th} individual. $NIND'(= NIND \times GGAP)$ individuals will be selected from the population in each generation; $GGAP$ = the generation gap. A roulette wheel is used to select the individuals according to their probabilities. Two individuals are selected to perform the crossover operator.

B. CROSSOVER

A crossover operator is crucial for searching an optimized solution. In this study, we design a crossover operator through changing the route structure.

The crossover operator aims to exchange the sequences of some intermediate stops of two routes with the same destination in different solutions, where the sequences of the intermediate stops are regarded as the smaller blocks. The scheme is shown in Fig.2. This operation is to explore better route structures with the same destination to reduce the total cost.

It is worth noticing that when the crossover operator is conducted, new connections between two nodes that cannot be connected directly according to the topology network may be generated. When this situation occurs, we use the Dijkstra's Algorithm to obtain the shortest path between the two nodes, and replace the two nodes on the route with the shortest path (represented by a series of nodes). Detailed procedures are as Algorithm 3:

It's also worth noticing that the new generated routes may not satisfy the route length constraint. If a new route is shorter than the lower limit or exceed the upper limit of length, a length penalty will be added to the objective function.

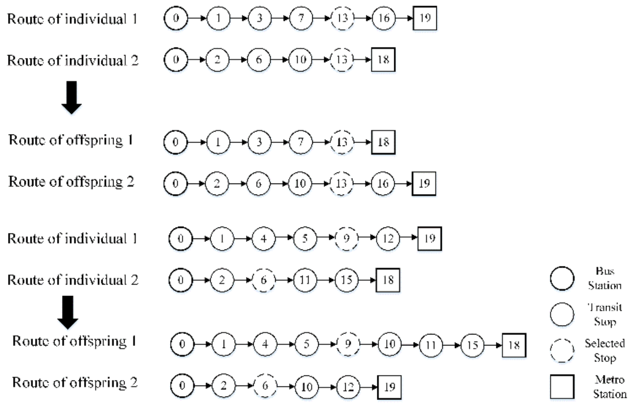


FIGURE 2. Representation scheme of the crossover operator.

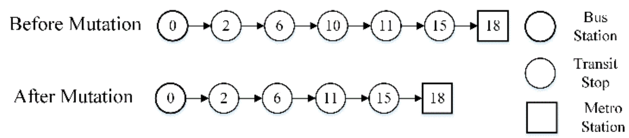


FIGURE 3. Representation scheme of the mutation operator.

C. MUTATION

The mutation operator is required to facilitate exploration in the large space when we use the GA. For the specific solution structure of the problem, a mutation operator based on removing some nodes is proposed for the GA, which is described in Fig.3. For a selected route, we first calculate the travel distance along the route between every two nodes, and subsequently verify whether it is equal to the shortest distance between the two nodes based on the Dijkstra's algorithm. If not, the current path between the two nodes may be replaced with the shortest path according to a certain preset probability. Detailed procedures are given in Algorithm 4.

When implementing the mutation operator above, a solution may not be changed if the path between every two nodes on the selected route is the shortest one. When this situation occurs, another route in this solution will be selected to implement the above process. The whole process repeats until a path replacement implemented successfully or the maximum number n of attempts of selecting routes reached.

After the path replacement process, we also need to check the subloop and the length constraint. The related operation is the same as the ones in the crossover operator.

D. REPAIR OPERATION

After the genetic operators, some offsprings may violate the constraint of covering all the stops. Therefore, a repair operator is proposed to repair these infeasible solutions. First, stops that do not appear in the solution as set C are identified. Next, select a node in C randomly to be inserted into a route at a certain position. If the node cannot be inserted directly at the position, the Dijkstra's algorithm will be used to obtain the shortest path, and the path will be inserted at that position.

TABLE 3. Crossover operator.

Algorithm 3: Crossover Operator

Input: *Parent*1 ← a solution selected from the population;
*Parent*2 ← a solution different from *Parent* 1 selected from the population;
Distance ← a matrix recording the connectivity and the distance (if connected directly) between every two nodes in the topology network;
Select *Route*1 from *Parent*1, and *Route*2 from *Parent*2;
if there exists at least one same stop in *Route*1 and *Route*2, **then**
  Node ← a randomly selected node from the same stops of *Route* 1 and *Route* 2;
  Divide *Route*1 into two parts from *Node*, representing as *sub_Route*1, *sub_Route*2;
  Divide *Route*2 into two parts from *Node*, representing as *sub_Route*3, *sub_Route*4;
  Exchange *sub_Route*1 and *sub_Route*3 (*sub_Route*2 and *sub_Route*4);
  Remove the loops in the new generated *Route* 1 and *Route* 2;
else
  *Node*1 ← a node randomly selected from *Route* 1;
  *Node*2 ← a node randomly selected from *Route* 2;
  Divide *Route*1 into two parts from *Node*1, representing as *sub_Route*1, *sub_Route*2;
  Divide *Route*2 into two parts from *Node*2, representing as *sub_Route*3, *sub_Route*4;
  if the last node of *sub_Route*1 cannot connect with the first node of *sub_Route*4 directly according to *Distance*, **then**
    *Path*1← the shortest path from the last node of *sub_Route*1 to the first node of *sub_Route*2 using the Dijkstra's Algorithm;
    *Route*1←*sub_Route*1+ *Path*1 + *sub_Route*4;
  end if
  if the last node of *sub_Route*3 cannot connect with the first node of *sub_Route*2 directly according to *Distance*, **then**
    *Path*2← the shortest path from the last node of *sub_Route*3 to the first node of *sub_Route*2 using the Dijkstra's Algorithm;
    *Route*2←*sub_Rout*3+ *Path*2 + *sub_Route*2;
  end if
  Remove the loops in the new generated *Route*1 and *Route*2;
end if
Update *Route*1 and *Route*2 in the two solutions;
Update *Parent*1 and *Parent*2 as *Offspring*1 and *Offspring*2 respectively

The route and position selected to conduct the repair operator must satisfy the following requirements: (1) any node on the route will not appear more than once; (2) the route

TABLE 4. Mutation operator.

Algorithm 4: Mutation Operator

1: Input: *Solution* ← a randomly selected solution in the population after crossover operator;
prob ← a preset probability for implementing the path replacement;
2: Select a *Route* from *Solution* randomly, represent the number of nodes in *Route* as $|Route|$
3: flag=0; //record whether a replacement has been implemented
4: **for** $n \leftarrow 1$ to $|Route|$-2
5: **for** $i \leftarrow 1$ to $|Route|$-2, **do**
6: **for** $j \leftarrow 3$ to $|Route|$, **do**
7: **if** the path from the i^{th} node to the j^{th} node is not the shortest path, **then**
8: $k \leftarrow$ a random number$\in[0,1]$;
9: **if** $k < prob$, **then**
10: *New_path* ← the shortest path from the i^{th} node to the j^{th} node;
11: Update *Route*: replace the current path with *New_path*;
12: break;
13: **end if**
14: **end if**
15: **end for**
16: break;
17: **end for**
18: **end for**
19: Remove the loops in the new generated *Route*
20: Update *Solution*

length will not exceed the upper limit after the node or the corresponding path is inserted. If we have exhausted all the possible situations but cannot obtain a feasible route or a position to insert, a new route will be established to cover the corresponding node in set C.

Finally, remove the node from set C and verify whether $C = \Phi$; if not, repeat the process above until $C = \Phi$.

V. A REAL-LIFE EXAMPLE AND NUMERICAL ANALYSIS

The real-life network presented in our experiment is the *Tiantongyuan* Community in Beijing. Using IC card data, we can obtain the passenger data at each stop for one hour during the early peak hours. To conduct our experiment, we use Matlab 2016a to code the program.

A. REAL-LIFE EXAMPLE

The study area, with the size of 2.6 $km \times 2.0$ km, is shown in Fig.4 (a). In Fig.4 (b), Twenty nodes, including a bus depot (node 0), seventeen bus stops (nodes 1–17), and two metro stations (nodes 18,19) are included in the topology network. The realistic solution, including five routes, can be seen in Fig.5 and the corresponding trip demand data of the two directions are presented in Table 5 and Table 6. The

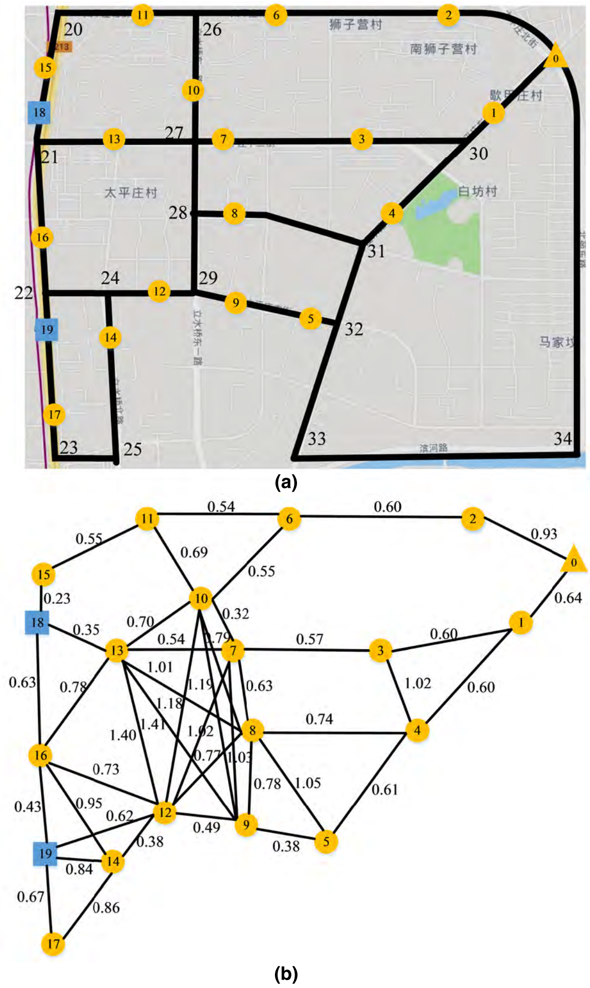


FIGURE 4. **(a) The road network of the study area. (b) The corresponding topology network (km).**

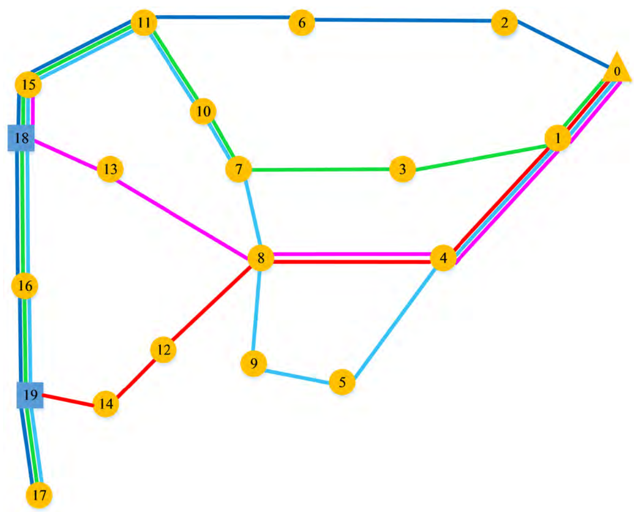


FIGURE 5. **Realistic route network.**

realistic solution, with the total fleet size of 13 vehicles and the headway of 6–12*min*, can meet the demand from each stop to each metro station in some certain degree. The total

TABLE 5. The demands from the stops to the metro stations.

D \ O	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
18	83	42	71	11	88	88	39	6	12	8	2	44	17	57	58	39	51	63
19	69	87	48	33	13	21	58	83	70	19	71	30	75	37	77	73	29	79

TABLE 6. The demands from the metro stations to the stops.

O \ D	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
18	4	7	1	20	33	42	7	49	56	81	46	79	30	1	74	4	7	1
19	19	51	90	82	45	16	39	39	21	30	44	12	21	23	90	19	51	90

TABLE 7. Data of optimized solution.

Routes	Headway(h)	Fleet Size	Total Cost($/h)	Supplier Cost($/h)	Waiting Cost($/h)	In-vehicle Cost($/h)	Penalty($/h)
0-1-3-7-10-11-15-18-16-19-17	0.20	2	8.85×10^3	3.90×10^3	2.60×10^3	1.37×10^3	9.70×10^2
0-1-4-8-12-14-19	0.15	2					
0-1-4-5-9-8-7-10-11-15-18-16-19-17	0.10	5					
0-1-4-8-13-18-15	0.20	2					
0-2-6-11-15-18-16-19-17	0.15	2					

TABLE 8. Data of optimized solution.

Routes	Headway(h)	Total Cost($/h)	Supplier Cost($/h)	Waiting Cost($/h)	In-vehicle Cost($/h)	Penalty($/h)
0-1-3-4-5-9-12-19	0.24	5.63×10^3	2.70×10^3	1.80×10^3	6.24×10^2	5.05×10^2
0-2-6-11-15-18-13-16-14-17-19	0.15					
0-2-6-10-7-12-19	0.24					
0-1-4-8-12-19	0.26					

cost of the realistic solution is $8.85 \times 10^3 \$/h$, with detailed cost shown in Table 7.

Then we optimize the route network in this area according to the trip demand in Table 5 and Table 6. Some related parameters are set as follows: $V_b = 30\ km/h$; $\gamma_W = 4\ \$/h$; $\gamma_I = 3\ \$/h$; $\gamma_S = 300\ \$/h$; $T_S = 1/90\ h$; $P = 45\ pass$; $\omega = 0.14$; $\alpha = 1.08$; $L_{max} = 10\ km$; $L_{min} = 2\ km$; $F_{max} = 13$ (equals to the real case). Some parameters in GA are set as follows: $population\ size = 20$; $gap = 0.9$; $MAXGEN = 2000$; $P_c = 0.8$; $P_m = 0.03$.

The crossover probability and the mutation probability are important parameters that influence the whole evolving process. Generally, a higher crossover probability or mutation operator may make the optimal solution more likely to be found since the expansion of the discovered solution space. However, it will also cost more CPU time definitely. For determining these two parameters, we change P_c from 0.5 to 0.9, and change Pm from 0.01 to 0.05 respectively. Results of the repeated experiments show that the total cost of the optimized solution tends to stable when P_c, P_m falling into the range of 0.8-0.9, 0.03-0.05, respectively. For saving the CPU time, we thus set P_c, P_m as 0.8, 0.03.

The convergence process of the experiment is shown in Fig.6. We can see the curve of the total cost keep stable after 950 generations. The final optimized solution, with the total cost of $5.63 \times 10^3\ \$/h$, is found in this experiment. The detailed results are presented in Table 8. The corresponding route network includes four routes, as shown in Fig.7. Compared with realistic solution, the optimized solution we've found decreases the total cost by 36.38% (the supplier cost, the waiting cost, the in-vehicle cost and the overloading penalty are decreased by 30.77%, 30.77%, 54.45%, 47.94%,

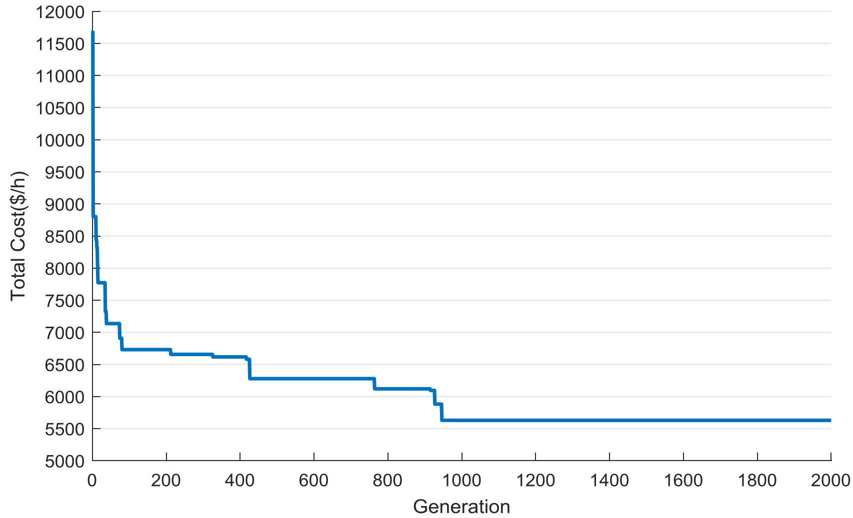


FIGURE 6. Convergence process.

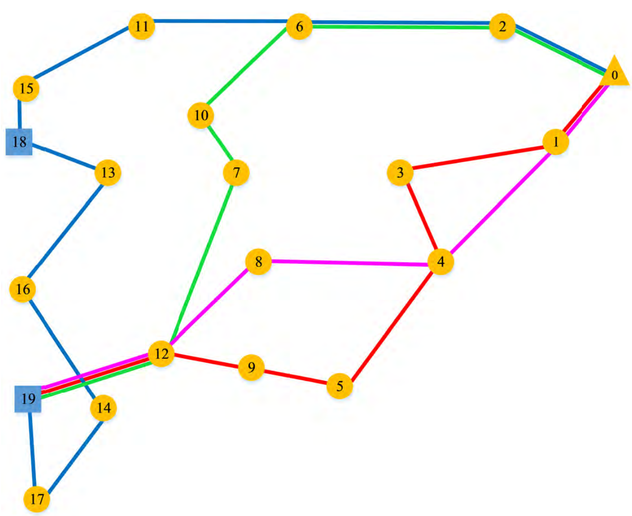


FIGURE 7. Optimized route network.

respectively). All the routes in both of the current solution and the optimized solution can meet the route length constraint, so there is no length penalty in both solutions.

We need to clarify that as a meta-heuristic method, GA may not find out the optimal solution theoretically. However, GA has been proved to be an effective method for solving the optimal routing or network design problem by lots of researchers. For example, [5] designs a GA and a Depth-first Search algorithm to find the optimized solution and the exact best solution respectively. The results indicate that GA can find out the best solution at most of the times. Therefore, we have reasons to believe that GA is effective and can be applied in solving the route network design problem.

B. NUMBERICAL ANALYSIS

We perform some analyses regarding to the relationship between the average route length of a solution and each element of the total cost and average headway. For each newly found feasible solution during the evolving process, we record the average route length, related costs, and average headway. Subsequently, we constructed a set of scatter plots, as shown in Fig 8. The increase in average route length can cause the increase of headway, which can be confirmed by (8). The waiting cost has a linear positive correlation with the headway of the corresponding route; therefore, an increase in the headway results in an increase in the waiting cost. The change in the in-vehicle cost is slight compared to that in the average route length. The user cost also increases with the average route length, because the waiting cost is its primary influence factor. The supplier cost increases because the increase in average route length results in the increase in operating time, which is also confirmed by (3). Finally, the total cost also increases with the average route length in general.

C. ALGORITHM COMPARISON

The algorithm designed in this study can solve the route network design and the frequency setting problem simultaneously. For evaluating the performance, we make a brief comparison of our algorithm with the ones proposed in [4] and [15].

For the trip assignment, we all use the method of the shortest path. In the frequency setting, our study determines an initial set of service frequencies through a theoretical method. Then if the total fleet sizes exceed its upper limit, we repeat removing a fleet according to the principle of the least impact on the total cost until the total fleet size constraint is satisfied. Reference [4] randomly assigns the fleet to each route first, then a heuristic of adjusting the fleet is used until the end condition is met. Actually, this internal heuristic is an ergodic process, which may consume more CPU time than ours because of more iterations. Reference [15] determines

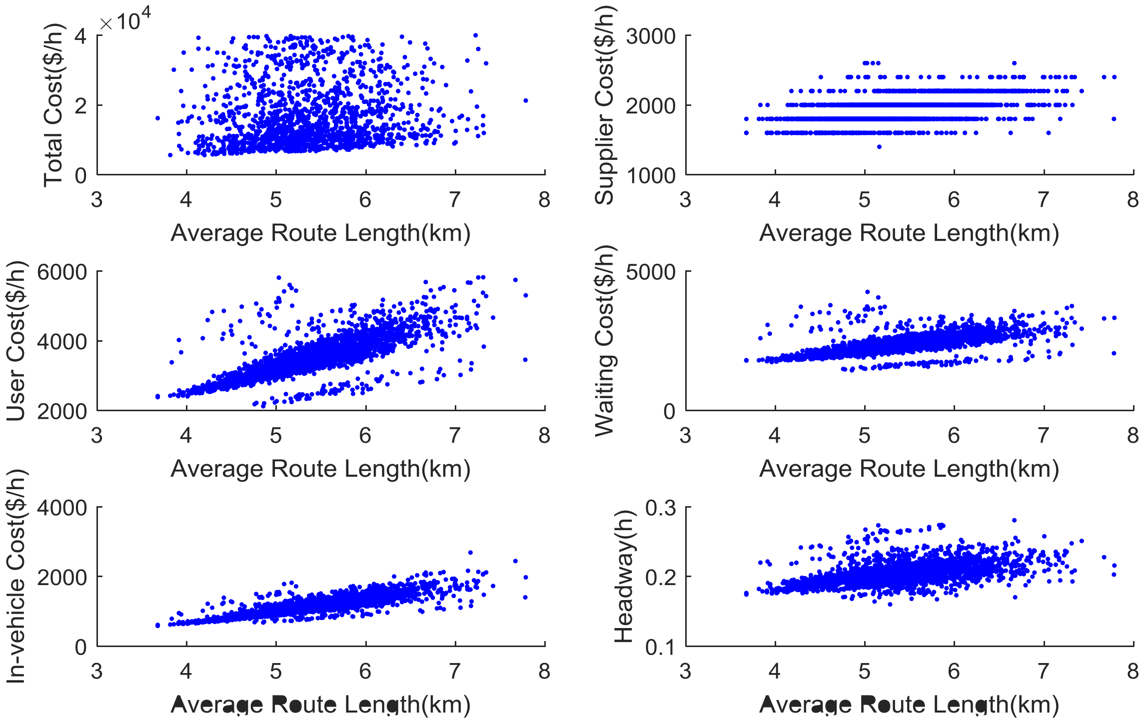


FIGURE 8. Different costs trend of average route length.

frequencies through a theoretical method without considering the total fleet size constraint. That may make the actual supplier cost exceed the investment of the supplier. Therefore, we believe that fleet size determination algorithm proposed in our study is efficient in dealing with the frequencies setting problem while maintaining the preciseness.

For the crossover operator, we first try to find a certain position for crossover operator rather than choosing positions randomly. It is helpful in avoiding infeasible solutions and make the runs of repair operation reduced, and the CPU time can be also saved. Reference [4] proposes a stop crossover operator that chooses two positions randomly, then swaps the middle part of the routes. The randomness of the operators may lead to the CPU time increasing because of a degree of blindness in the improving process. The crossover proposed in [15] swaps two routes at an identical stop, which may consume extra runs in searching routes with common stops. Based on the analysis above, we believe that the operators designed in our GA are more efficient and effective in solving the route network optimization problem compared with some previous works.

VI. CONCLUSION

In this study, we develop an optimal route network design problem for a community shuttle linked with metro service, and formulate a mixed integer model to minimize the total cost, considering a series of constraints such as the route length and all the bus stops are covered. A trip assignment method based on the logistic function and a fleet size adjustment heuristic are embedded in the model formulation to make the network analyze procedure be conducted smoothly. For solving this problem, we propose a GA consisting of some specifically designed operators. Finally, a real-life study is presented to test the algorithm followed by analysis. The key conclusions are summarized as follows:

(1) The proposed GA performed well in solving real-life problems. A significant decrease by 36.38% in the total cost and a clear convergence trend are shown during the evolving process of 2000 iterations. The optimized network and the corresponding headway of each route are obtained simultaneously.

(2) Through the numerical analysis, the average route length demonstrates a positive effect on the average headway, the waiting cost, the user cost, the supplier cost, and the total cost, while the change of the in-vehicle cost is not obvious.

Future research could be performed from the following aspects:

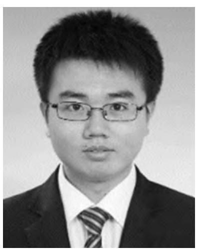
(1) Using a single crossover operator may not be effective in solving this problem. The design of multiple crossover and mutation operators can be enhanced to improve the efficiency and effectiveness of the GA.

(2) In this study, the passengers' access cost and the bus stop service distance were not considered. In the future research, these factors can be incorporated into the model to make the problem more realistic.

REFERENCES

[1] A. Černá, J. Černý, and V. Přibyl, "Bus route design in small demand areas," *Transport*, vol. 26, no. 3, pp. 248–254, Jul. 2011.

[2] J.-D. Sergio, F. Andrés, and G. Antonio, "Optimal fleet size, frequencies and vehicle capacities considering peak and off-peak periods in public transport," *Transp. Res. A, Policy Pract.*, vol. 106, pp. 65–74, Dec. 2017.
[3] S. B. Pattnaik, S. Mohan, and V. M. Tom, "Urban bus transit route network design using genetic algorithm," *J. Transp. Eng.*, vol. 124, no. 4, pp. 368–375, Jul. 1998.
[4] W. Y. Szeto and Y. Wu, "A simultaneous bus route design and frequency setting problem for Tin Shui Wai, Hong Kong," *Eur. J. Oper. Res.*, vol. 209, no. 2, pp. 141–155, Mar. 2011.
[5] J. Xiong, W. Guan, L. Song, A. Huang, and C. F. Shao, "Optimal routing design of a community shuttle for metro stations," *J. Transp. Eng.*, vol. 139, no. 12, pp. 1211–1223, Dec. 2013.
[6] Y. Jiang, W. Szeto, and T. Ng, "Transit network design: A hybrid enhanced artificial bee colony approach and a case study," *Int. J. Transp. Sci. Technol.*, vol. 2, no. 3, pp. 243–260, Sep. 2013.
[7] J. Xiong, W. Guan, and A. L. Huang, "Research on optimal routing of community shuttle connect rail transit line," *J. Transp. Syst. Eng. Inf. Technol.*, vol. 14, no. 1, pp. 166–173, Feb. 2014.
[8] O. Y. Xian, M. Chitre, and D. Rus, "An approximate bus route planning algorithm," in *Proc. IEEE Symp. Comput. Intell. Veh. Transp. Syst.*, Apr. 2013, pp. 16–24.
[9] J.-J. Lin and H.-I. Wong, "Optimization of a feeder-bus route design by using a multiobjective programming approach," *Transp. Planning Technol.*, vol. 37, no. 5, pp. 430–449, Jul. 2014.
[10] S. Wang and X. Qu, "Rural bus route design problem: Model development and case studies," *KSCE J. Civil Eng.*, vol. 19, no. 6, pp. 1892–1896, Sep. 2015.
[11] B. Barabino, "Transit bus route network design: A model and its application in a real network," *WIT Trans. Built Environ.*, vol. 107, pp. 369–382, Jun. 2009.
[12] M. H. Almasi, S. M. Mounes, S. Koting, and M. R. Karim, "Analysis of feeder bus network design and scheduling problems," *Sci. World J.*, vol. 2014, Jan. 2014, Art. no. 408473.
[13] Y. Kuo, C.-M. Luo, and C.-C. Wang, "Bus route design with limited travel time," *Transport*, vol. 28, no. 4, pp. 368–373, Dec. 2013.
[14] S. Ngamchai, D. J. Lovell, and A. M. ASCE, "Optimal time transfer in bus transit route network design using a genetic algorithm," *J. Transp. Eng.*, vol. 129, no. 5, pp. 510–521, Sep. 2003.
[15] S. I. Chien, Z. Yang, and E. Hou, "Genetic algorithm approach for transit route planning and design," *J. Transp. Eng.*, vol. 127, no. 3, pp. 200–207, May 2001.

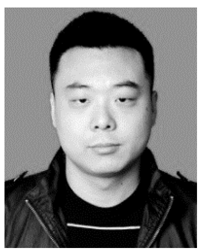


JIE XIONG was born in Harbin, Heilongjiang, China, in 1988. He received the degree from Beijing Jiaotong University, in 2016, and the Ph.D. degree in transportation engineering. He was with the Beijing University of Technology. He is a Lecturer. His research interests include transportation systems engineering and transit optimization.

BIAO CHEN was born in Zhangjiakou, Hebei, China, in 1992. He is currently pursuing the master's degree in transportation engineering. His research interests include transportation systems engineering and transit optimization.

YANYAN CHEN received the degree from the Harbin Institute of Technology, in 1997, and the Ph.D. degree in civil engineering. She was with the Beijing University of Technology. She is a Professor. Her research interests include transportation planning, and traffic energy conservation and emission reduction.

YANG JIANG was born in Shenyang, Liaoning, China, in 1986. He received the degree from Beijing Jiaotong University, in 2014, and the Ph.D. degree in transportation engineering. He was a Research Assistant with the School of Economics and Management, Tsinghua University, from 2015 to 2017. He is a Lecturer. He is currently with the Shenyang University of Technology. His research interest includes the management of railway logistics in relation to intermodal transportation using mathematical modeling methods.



YAO LU received the degree from Southwest Jiaotong University, in 2009, and the Ph.D. degree in transportation engineering from the Beijing University of Technology. His research interests include city group transportation planning and airport traffic planning.