IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received November 25, 2021, accepted December 19, 2021, date of publication January 6, 2022, date of current version January 14, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3140947

Data-Driven Bus Route Optimization Algorithm Under Sudden Interruption of Public Transport

RUISONG LIU AND NING WANG
School of Information Engineering, Shandong Management University, Jinan 250357, China

Corresponding author: Ning Wang (wangning200658@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772068, in part by the Shandong Social Science Planning Research Project under Grant 19BJCJ18, in part by the Project of Shandong Management University's Scientific Research Sailing Plan under Grant QH2020Z04, and in part by the Ph.D. Research Fund of Shandong Management University under Grant SDMU201709.

ABSTRACT With the continuous expansion of urban traffic operation scale, public transport emergencies occur from time to time, causing serious traffic jams and potential safety hazards in a short time. In view of this, a new adaptive bus route optimization strategy based on the emergency demand responsive public transport is proposed. Firstly, in order to improve the fine-grained passenger carrying capacity of emergency demand responsive public transport and build a clustering model of passenger information, this paper proposes an adaptive clustering algorithm, which considers the main influencing factors such as vehicle capacity, passenger travel time window and the number of stations visited. Aiming at minimizing the cost of vehicle operation and passenger traffic, a multi-objective optimization model of emergency bus route is constructed based on Vehicle Routing Problems with Time Windows (VRPTW) to ensure the operation efficiency of emergency bus. Secondly, a Modified Adaptive Large Neighborhood Search with Nearest Vehicle Dispatch (NVD) algorithm (MALNSN) is proposed, which is an extension of the Adaptive Large Neighborhood Search algorithm (ALNS), by improving the generation rules of initial solutions with NVD and operator selection strategy with Modified Choice Function (MCF), and the effectiveness of algorithm is analyzed according to the Solomon benchmark. The average gain of the proposed MALNSN algorithm is 17.11% higher than that of the original algorithm. Finally, based on the actual road network, experiments are carried out to compare the proposed algorithm with the representative algorithms. The experimental results show that the MALNSN algorithm proposed in this paper can not only ensure the stability of the algorithm, but also formulate a reasonable route optimization strategy in a shorter time, effectively reducing the consumption of transport capacity resources, improving the operation efficiency of public transport and increasing the accessibility of public transport. The theoretical analysis was consistent with the experimental results.

INDEX TERMS VRPTW, transport interruption, responsive bus, clustering, route planning.

I. INTRODUCTION

In recent years, with the rapid development of intelligent transportation in China, a large number of transportation resources have been transformed into data resources, of which public transport resources have increased in the share year by year. However, the continuous expansion of urban traffic operation scale, the growing passenger demand and high-intensity operation have brought great pressure to the urban traffic system, as a result of which, the sudden failures and equipment failures occur from time to time, resulting in public transport interruption. After the interruption, the traffic capacity of the road section is insufficient, and passengers are stranded and overstocked in the public stations, resulting in additional waiting time and transfer time that will lead to delays for passengers. The occurrence of public transport interruption will have a significant impact on passengers and transportation companies [1]. Therefore, the research in this field is a hot issue at present and has high academic and commercial value.

For sudden public transport interruption, the strategies of shortening the departure interval or increasing the number of vehicles in this section are usually adopted. Given the problems in the traditional public transport such as

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

inconvenient transfer, low density and even blind spots, emergency demand responsive public transport arises at the right moment, which flexibly adjusts the transport capacity according to passengers' personalized travel needs, calculates the optimal line in real time by passenger flow and virtual stations, and quickly and dynamically allocates public transport capacity resources to achieve the optimal overall efficiency, effectively make up for the mismatch between transport capacity and demand of traditional public transport in specific areas and periods, improve the operation efficiency of public transport, reduce the consumption of transport capacity resources and increase the accessibility of public transport.

In the case of sudden operation interruption, how to scientifically organize public transport vehicles to evacuate stranded passengers in time and reduce passenger travel delay has important theoretical and practical significance. As the emergency demand responsive public transport generally takes the distance as the charging basis and focuses on the personalized needs of passengers who hope to reach the destination in a shorter time, the public transport operators want to solve the temporary emergency problems with less operating resources [2]. According to the needs of stranded passengers, the temporary emergency bus route is planned in a scientific and reasonable way, and the bus dispatching is optimized by reallocating vehicle resources to meet the real-time travel needs of passengers and make rational use of vehicle resources [3]. For vehicle routing optimization scheduling, a NP-Hard problem, heuristic algorithm is usually adopted to get a better solution [4].

The main contributions of this paper include the following aspects: (1) For passenger information clustering, an adaptive density clustering algorithm based on Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is proposed to make the division of passenger spatial clustering center more reasonable; (2) An improved adaptive large neighborhood search algorithm (MALNSN) is proposed; (3) This paper, abstracting the vehicle route optimal scheduling problem into VRPTW class problem, takes the cost of vehicle operation and passenger traffic as the basis of evaluation, and puts forward an improved fitness operator method for the vehicle route optimal scheduling problem.

II. RELATED WORK

Urban rail transit plays an important role in the urban smart transportation system by virtue of its characteristics of safety, punctuality, speed, comfort and large passenger capacity. In the case of traffic interruption, if the stranded passengers in the station cannot be evacuated in time, it will greatly affect the normal operation of the ground transportation network. Therefore, efficient evacuation scheduling scheme has become the focus of current research. Gu *et al.* [5] established a two-stage model aiming at minimizing bus connection time and passenger delay time, which optimized the emergency bus connection scheme based on fixed paths. J. D. Wang *et al.* [6] established an emergency bus dispatching model under flexible paths with the goal of minimizing the total evacuation time and the average passenger delay time, which was solved with ideal point method and the genetic algorithm for solution. Compared with the traditional fixed route vehicle scheduling, this scheme can effectively improve the distribution efficiency and the service quality. Y. J. Zheng *et al.* [7] established a two-stage optimization model under the principle of efficiency and fairness, applied it to emergency bus under flexible paths, and designed an improved tabu search algorithm to solve the model. Therefore, it is feasible to formulate evacuation dispatching scheme based on emergency bus. This paper will carry out further research on this basis.

For emergency vehicle scheduling, most scholars use VRPTW as the framework of abstract model, aiming to provide services for all customers within a specific time interval. M. Bryan David Galarza *et al.* [8], combining demand response and fixed bus service, applied the Large Neighborhood Search heuristic algorithm (LNS) to the route optimization scheduling of real-time responsive public transport to optimize the system performance. The advantage of this method is that the algorithm is fast and can search for a better solution in a short time. However, it takes a single objective as the scheduling evaluation index, which results in that the impact of other factors on vehicle scheduling is not considered in many aspects. F. In order to optimize the bus route, Lang *et al.* [9] improved the flower pollination algorithm where the proportion of the average travel time and transfer times of passengers is taken as the optimization goal, and the improved initial population generation method and the improved pollen pollination algorithm framework is adopted to obtain a set of better bus routes. G. Rongge [10] integrated the decision-making considerations of bus route planning, route selection between nodes and passenger allocation to propose a hybrid meta heuristic method to adapt to the temporal and spatial heterogeneity in travel demand by means of tabu search and variable neighborhood search. In order to solve the VRPTW, most scholars have carried out further exploration through the combination of multiple algorithms and heuristics. Y. Shen [11] proposed a hybrid swarm intelligence algorithm based on Hybrid Ant Colony System (ACS) algorithm and Brain Storm Optimization (BSO) algorithm to solve VRPTW with the goal of minimizing the total distance. M. He [12], improving the pheromone update strategy of ant colony algorithm and setting the conditions and operators of variable neighborhood search, proposed an Adaptive Variable Neighborhood Search Ant Colony Algorithm (AVNSACA), which optimizes the problem of insufficient pheromone in the early stage and prevents the algorithm from falling into local optimization, but increases the computational complexity. G. Li [13] proposed an Improved Tabu Search algorithm (I-TS) based on greedy algorithm, in which, the probability distribution of vehicle driving and service time is introduced, and a modified stochastic programming model is established to minimize the allocation cost. J. Zhang [14] proposed an

Evolutionary Scattering Search Particle Swarm Optimization algorithm (ESS-PSO) aiming at minimizing the total driving distance, setting the path segment as the speed of particles, and reconstructing the update rules based on the concept of "destruction and reconstruction", which improves the overall performance of the algorithm, but increases the complexity of the algorithm. Y. Marinakis[15] proposed a Multiple Adaptive Particle Swarm Optimization (MAPSO) algorithm, which starts with the random values of parameters and adjusts all parameters in the iterative process according to certain conditions, thus improving the performance of the algorithm without increasing the complexity of the algorithm. G. Srivastava [16] proposed a Genetic Algorithm II based on Non-dominant Sorting (NSGA-II), which designs crossover and mutation operators by using the characteristics of the problem and the attributes of each target. C. Chen Yang [17] proposed an auction bidding method to balance the performance bonus awarded to all drivers, which not only achieved the balance of performance bonus, but also planned the shortest route for each driver, making the strategy more reasonable. W. Jiahai [18] proposed a hybrid multi-objective Memetic Algorithm for solving multi-objective where a two-stage strategy is designed to improve the comprehensive performance of convergence and diversity, providing another idea for heuristic solution.

As a fast optimization heuristic algorithm, neighborhood search has attracted extensive attention of researchers, and has been widely used in VRPTW through the improved heuristic algorithm Y. Jiang [19] proposed the Simulated Annealing algorithm of Variable Neighborhood Search (VNS-SA) to solve the VRPTW like problem, aiming at reducing the conflict between delivery cost and service level. Meanwhile, C. Jiumei [20] proposed a Variable Neighborhood Search (VNS) method with local search and jitter key steps, taking a variety of costs as the optimization goal. Grigorios [21] proposed a multi-objective optimal scheduling algorithm based on large neighborhood search (MOLNS), which defined the destruction and repair operator for the purpose of minimizing the number of scheduled vehicles and the total driving distance, and proved the effectiveness of the algorithm through simulation experiments. The advantage of this model is that its algorithm is fast, which contributes to searching the optimal solution in a short time and its disadvantage is that LNs heuristic is easy to fall into local optimization. Hellsten [22] applied the Adaptive Large Neighborhood Search heuristic algorithm (ALNS) to line scheduling, aiming at minimizing the policy execution time, and adopted adaptive operator selection to make the system more robust. B. Pan [23] developed a subdivision evaluation method to speed up the feasibility test of a given vehicle route where an efficient hybrid Adaptive Large Neighborhood Search algorithm with Tabu Search (ALNS-TS) was achieved to explore the feasible and infeasible solution space.

For neighborhood search heuristic to solve VRPTW, Adaptive Large Neighborhood Search (ALNS) algorithm is one of the more effective adaptive methods [22]–[24].

TABLE 1. Comparison of algorithms for solving VRPTW.

References	Methodology	Advantages	Disadvantages
[7]	Emergency bus dispatching algorithm based on Tabu Search	·Multi objective optimization ·Strong global search ability ·The algorithm is easy to implement	·The stability of the algorithm is weak
[8]	Emergency bus dispatching algorithm based on large neighborhood search	·Fast convergence ·Strong search ability	·Single objective optimization
[9]	Bus route optimization algorithm based on improved pollen pollination	·Multi objective optimization ·Strong search ability	·High operation cost
[12]	Vehicle route optimization algorithm based on adaptive variable neighborhood search ant colony	·Strong global search ability ·Comprehensive consideration of multiple factors	·High operation cost ·High time complexity
[15]	Vehicle route optimization algorithm based on multiple adaptive particle swarm optimization	·Strong search ability ·Fast convergence	·Single objective optimization
[21]	Multi-objective vehicle route optimal scheduling algorithm based on large neighborhood search	·Multi objective optimization ·Fast convergence ·Strong search ability	·Easy to fall into local optimization
[22]	Vehicle route optimization algorithm based on adaptive large neighborhood search	·Fast convergence ·Strong search ability ·The algorithm is easy to implement	·Single objective optimization

In the process of algorithm search and optimization, ALNS uses a variety of ring breaking and repair operators, increasing the measurement of the effect of each operator and adopts the roulette mechanism to select the best operator. Compared with the roulette mechanism, the Choice Function (CF) [25] introduces a better selection technology, which scores the heuristic algorithm according to the thinking of three different metrics. Drake proposed a special concept of selection function, and named it Modified Choice Function (MCF) [26], which is superior to the original choice function in the set combination problems, including optimizing routes to minimize the total travel time. In this paper, the modified choice function is incorporated into the operator selection stage of ALNS algorithm to make ALNS algorithm more effective in selecting destruction and repairing operators.

For emergency response public transport, the current method pays more attention to the maximization of the interests of the public transport operator, fully considering the minimum travel time and operation times of each route, but taking less consideration of the needs of passengers. For example, the waiting time of passengers is usually not fully taken into account in the route planning process. Taking passengers as the center, this paper takes passengers' transportation cost and bus operation cost as the optimization goal of this paper.

III. PROBLEM DESCRIPTION AND ADAPTIVE BUS ROUTE OPTIMIZATION SOLUTION STRATEGY

A. PROBLEM DESCRIPTION

As an important transportation field for the implementation of smart transportation strategy, urban rail transit has an important position in the urban system due to its large passenger capacity, and effective alleviation of urban traffic pressure. In the event of sudden public traffic disruption, failure to evacuate stranded passengers in time will greatly affect the normal operation of ground transportation network, thus causing security risks. Thus, how to formulate reasonable basis quickly and effectively for stranded passengers of emergency evacuation data strategy is the key issue to deal with interruption events. For evacuation strategy, the expectation of passengers is to arrive at their destination in time with the minimum transportation cost and that of bus operators is to meet the needs of passengers with the minimum transportation cost. In this paper, it is suggested that while satisfying passengers to arrive at their destinations in time, emergency response bus resources and bus routes could be optimized, so that the evacuation strategy of stranded passengers can adaptively plan better emergency bus routes according to passenger data under different circumstances, thereby improving the service quality of public transportation, reducing transportation resources and operating costs, and ensuring smooth traffic.

The adaptive bus line optimization solution strategy is shown in Figure 1. Firstly, passenger density clustering analysis is performed based on the travel time and space demands proposed by stranded passengers. Secondly, density clustering analysis is used to remove density noise and promptly notify noise passengers that they do not meet the emergency response demand, and then a variety of clustered data is derived from passenger data consistent with the Solomon benchmark to form virtual stations. Finally, the space-time demand of passengers obtained by clustering is used to optimize station access order and route selection and to plan routes. The planned routes should be considered from the following points:

(1) Emergency vehicles must serve all virtual stations and meet the service time window of passengers and virtual stations.

(2) The number of passengers carried by each emergency vehicle shall not be less than the minimum number of passengers carried by the vehicle.

(3) Minimize the length of emergency vehicle service routes.

(4) Minimize bus operating costs.

(5) Minimize passenger transportation costs.

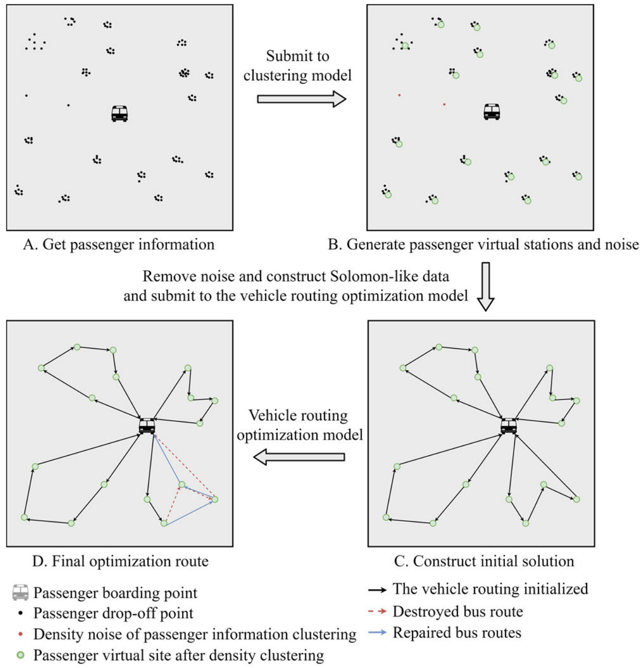

FIGURE 1. **Problem description example diagram.**

B. PASSENGER INFORMATION CLUSTERING MODEL

In response to traffic interruptions, emergency response buses are used to evacuate passengers to their desired destinations. In order to improve the granularity of bus passenger loading and ensure the efficiency of emergency bus operations, it is necessary to cluster the time and space information of passengers. Through the density processing of the demand information submitted by passengers (including travel starting point, ending point and expected arrival time), the demand points with similar starting points and ending points are classified into the same category, and the cluster center of the starting point (ending point) is used as the getting-off point of the customized bus.

DBSCAN algorithm is a density-based spatial clustering algorithm proposed by M. ester [27] in 1996 to improve the fine-grained passenger carrying capacity of real-time demand responsive public transport and ensure the operation efficiency of public transport, in which, parameters *Eps* and *MinPts* play a very important role. Due to the uncertainty of passenger space information in each clustering, fixed parameter values cannot be adapted. This paper proposes an adaptive mechanism for this.

The candidate *EPS* and *MinPts* parameter list is generated by using the distribution characteristics of the denoised and attenuated data set, and the corresponding *EPS* and *MinPts* are selected as the initial density threshold according to the denoising level in the interval where the number of clusters tends to be stable. For the noise data generated by clustering under the condition of density threshold, the same method is used to generate a candidate parameter list, select the optimal parameters, and obtain a new density threshold. This step is then repeated until the number of noise data or density threshold is below a certain degree.

1) SELF-ATTENUATION GENERATES EPS PARAMETER LIST

Eps list is generated by K-Average Nearest Neighbor with Attenuation Term Algorithm, which is an extension of KNN.

The algorithm calculates the k-nearest neighbor distance between each data point in data set D and its *k-th* nearest data point, and averages the k-nearest neighbor distances of all data points to obtain the k-average nearest neighbor distance of the data set. The specific steps to generate the parameter list are as follows:

Step 1 Calculate the Euclidean distance from each point in dataset D to other points to form the distance distribution matrix $Dist_{n\times m}$, as shown in (1).

$$Dist_{n\times n} = \{dist\ (i, j) \mid 1 \le i \le n, 1 \le j \le n\} \quad (1)$$

where n is the number of data points in D; $dist\ (i, j)$ is the distance between point i and point j in the dataset, and $Dist_{n\times n}$ is a real symmetric matrix of $n \times n$.

Step 2 Sort the elements of each row of $Dist_{n\times n}$ in ascending order.

Step 3 Average the *k-th* column of the sorted $Dist_{n\times n}$ to $\bar{D_k}$, and use it as a candidate *Eps* parameter after subtracting the self-attenuation term, as shown in (2).

$$Eps_k = \left(1 - \lambda^2\right) \bar{D_k} \quad (2)$$

where λ is the self-attenuation coefficient ($0 \le \lambda \le 1$), and is valued as 0.3 in this paper. After all values are calculated, the *Eps* parameter list is obtained, as shown in (3).

$$Eps_{list} = \left\{Eps_k \mid 1 \le K \le n\right\} \quad (3)$$

2) GENERATE MINPTS PARAMETER LIST FROM SELF-ATTENUATION

According to Equation (4), the *MinPts* list is generated by the mathematical expectation based on the self-attenuation term.

$$MinPts_K = \frac{(1-\lambda)}{n} \sum_{i=1}^{n} P_i \quad (4)$$

where λ is the self-attenuation coefficient ($0 \le \lambda \le 1$), n is the total number of objects in the dataset, and P_i is the number of neighborhood objects in the *Eps* neighborhood of the *i-th* object. After all values are calculated, the *MinPts* parameter list is obtained, as shown in (5):

$$MinPts_{list} = \{MinPts_K \mid 1 \le K \le n\} \quad (5)$$

3) MULTI DENSITY ADAPTIVE DETERMINATION OF OPTIMAL PARAMETERS

Firstly, the parameter list of *Eps* and *MinPts* is obtained with the improved method of self-attenuation, and the *k-th* parameter pair ($0 \le K \le n$) is selected, that is (Eps_k, $MinPts_K$). DBSCAN algorithm is input to cluster the data set, and the relationship between the number of clusters and K value is obtained. When the cluster number of clustering results tends to be stable, the optimal parameters are inversely selected by K value. This paper redefines the judgment criterion that the result tends to be stable. X represents the number of continuous clusters with the same number of clustering results:

(1) When the continuous X (its initial value is set as 6 in this paper) times are the same, it is considered that the clustering result tends to be stable, the stable interval is obtained (the first stable interval is taken in this paper), and the number of clusters is recorded as the optimal number of clusters. If there is no same number of clustering results for consecutive X times, the same number of clusters is searched for consecutive $X - 1$ times.

(2) If $X < 3$ in (1) is still not found, the stable interval with the fluctuation range of cluster number within 1 should be searched instead. After finding the stable interval, the interval endpoints $start_K$ and end_K are calculated. In this paper, the K value in the stable interval is selected according to the needs of identifying the noise level, and realized by setting the noise level, which can be divided into three types: *less*, *normal* and *more*(default). The selection of K value is shown in the (6).

$$K = \begin{cases} start_K, & level = more \\ \frac{(start_K + end_K)}{2}, & level = normal \\ end_K, & level = less \end{cases} \quad (6)$$

Generally, the noise level is set as *less* on the data set with uniform density distribution, *more* on the data set with large difference in density distribution, and *normal* in other cases.

C. BUS ROUTE OPTIMIZATION MODEL

Emergency response public transport can calculate the optimal path in real time according to the passenger demand and virtual station, and quickly carry out the dynamic allocation of public transport capacity resources to achieve the optimal overall efficiency. In this paper, a bus route optimization model is designed to minimize the cost of passenger traffic and the timeliness of reaching the target point, as well as the investment by bus operators.

1) MODEL PARAMETERS

In this paper, G=(V, A) is used to represent the road network, where V represents the node, which is composed of the virtual station set S generated by clustering and the parking lot point O, that is, $V = S + O$, and A represents the arc set. Node $i(i\in S)$ in set S corresponds to non-negative weight attributes: service time s_i and service time window $[T_i^{Arr}, T_i^{Dep}]$, where T_i^{Arr} is the earliest arrival time of the vehicle, and T_i^{Dep} is the latest departure time of the vehicle. Nodes $r(r\in V)$ and $s(s \in V)$ in set V correspond to the passenger travel demand $q_{r,s}$ and the passenger travel cost $f_{r,s}$ respectively. Node j in set O corresponds to the parking lot service time window $[u_j, l_j]$, u_j and l_j respectively indicate the time when the service starts and ends in the depot, which means that the vehicles need to start and return to the depot within $[u_j, l_j]$. Each arc (i, j) in set A corresponds to a set of optional paths $P_{i,j}$, and each path $\varphi_{i,j,p}(p\in P_{i,j})$ is associated with distance $d_{i,j,p}$ and travel time $t_{i,j,p}$. K represents the vehicle set, and each vehicle k corresponds to a non-negative weight attribute: rate v, maximum capacity M, number of passengers Q_{ik} when arriving at station i, and minimum passenger capacity λ,

TABLE 2. Decision variables.

variables	meaning
$x_{i,j,k}$	If vehicle k drives the section (i,j), then $x_{i,j,k}=1$, otherwise $x_{i,j,k}=0$
$x_{i,j,k,p}$	If the vehicle k selects a path $\varphi_{i,j,k,p}$, then $x_{i,j,k,p}=1$, otherwise $x_{i,j,k,p}=0$
$y_{r,s,k}$	Number of passengers transported by vehicle k from station r to s
$\xi_{i,k}$	Time when vehicle k arrives at station i
$e_{i,k}$	Time when vehicle k leaves station i
$w_{i,k}$	Capacity of vehicle k at station i

unit departure cost c_1, unit time operation cost c_2, and unit distance operation cost c_3. The definition of decision variables is shown in Table 2.

2) OBJECTIVE FUNCTION AND CONSTRAINTS

The emergency demand responsive bus routing optimization problem is a NP-Hard problem, which can form a mixed integer programming model. The model optimization is aimed at lowering the cost of bus operation and passenger transportation. The mathematical model is as follows:

$$\min z = \alpha C^{Bus} + \beta C^{Passenger} \tag{7}$$

$$\begin{aligned} C^{Bus} &= c_1 \sum_{k\in K}\sum_{i\in O}\sum_{j\in S} x_{i,j,k} \\ &+c_2 \sum_{k\in K}\sum_{i\in V}\sum_{j\in V}\sum_{p\in P_{i,j}} x_{i,j,k,p}t_{i,j,p} \\ &+c_3 \sum_{k\in K}\sum_{i\in V}\sum_{j\in V}\sum_{p\in P_{i,j}} x_{i,j,k,p}d_{i,j,p} \end{aligned} \tag{8}$$

$$C^{Passenger} = p_1 \sum_{k\in K}\sum_{r\in S}\sum_{s\in S} f_{r,s}\gamma_{r,s}^k + p_2 \sum_{k\in K}\sum_{r\in S}\xi_{r,k}\sum_{s\in S}\gamma_{r,s}^k \tag{9}$$

$$\sum_{k\in K}\sum_{i\in V} x_{i,j,k} = 1, \quad \forall j\in S \tag{10}$$

$$\sum_{i\in O}\sum_{r\in S} x_{i,r,k} - \sum_{s\in S}\sum_{j\in O} x_{s,j,k} = 0, \quad \forall k\in K \tag{11}$$

$$\sum_{r\in S}\sum_{s\in S} \gamma_{r,s,k} \geq \lambda, \quad \forall k\in K \tag{12}$$

$$\sum_{k\in K} \gamma_{r,s,k} = q_{r,s}, \quad \forall r,s\in S \tag{13}$$

$$\begin{aligned} &Q_{j,k} = Q_{i,k} + \left(\sum_{r\in S}\gamma_{r,i,k} - \sum_{s\in S}\gamma_{r,s,k}\right)x_{i,j,k}, \\ &\qquad \forall i,j\in S;\ \forall k\in K \\ &Q_{j,k} = 0, \quad \forall i\in O;\ \forall k\in K \\ &Q_{j,k} \leq M, \quad \forall i\in S;\ \forall k\in K \end{aligned} \tag{14}$$

$$\begin{aligned} &\xi_{i,k} \geq \xi_{i,k} + (t_{i,j}+s_i)\,x_{i,j,k} - B\left(1-x_{i,j,k}\right), \\ &\qquad \forall i,j\in V;\ \forall k\in K \\ &\mu_{i,k} \geq u_i, \quad \forall i\in O;\ \forall k\in K \\ &\xi_{i,k} \leq l_i, \quad \forall i\in O;\ \forall k\in K \end{aligned} \tag{15}$$

$$\sum_{j\in V} x_{i,r,k} = 1 \Rightarrow T_i^{Arr} \leq \xi_{i,k} \leq \mu_{i,k} \leq T_i^{Dep}, \quad \forall i\in S;\ \forall k\in K \tag{16}$$

$$\mu_{i,k} = \xi_{i,k} + s_i, \quad \forall i\in S;\ \forall k\in K \tag{17}$$

where α, β represent the weight, whose values are constants between 0 and 1, and their size represent the concern of bus operation cost and passenger transportation cost. p_1 and p_2 represent the fitness coefficients of passenger travel cost and passenger waiting time respectively, and B is a large constant.

Objective function: Equation (7) defines the minimization objective, which is composed of bus operation cost and passenger traffic cost; Equation (8) represents the operating cost of public transport, including departure cost, running time cost and running distance cost; Equation (9) represents passenger traffic cost, including passenger travel cost and passenger waiting time. Constraints: Equation (10) indicates that the site can be accessed only once; Equation (11) indicates that the vehicle departs from the station and returns; Equation (12) specifies the minimum passenger capacity of each vehicle; Equation (13) assigns constraints to passengers; Equation (14) is the vehicle capacity constraint; Equation (15) ensures that the time of vehicle departure from the station is feasible; Equation (16) is expressed as a time window constraint; Equation (17) indicates that the vehicle departure time is the sum of the vehicle arrival time and the service time.

IV. ALGORITHM DESIGN AND OPTIMIZATION

A. TRADITIONAL ADAPTIVE LARGE NEIGHBORHOOD SEARCH ALGORITHM

Adaptive large neighborhood search algorithm (ALNS) is an algorithm proposed by ropke and pisinger in 2006 [24], which is an extension of the Large Neighborhood Search algorithm (LNS) proposed by Shaw (1988) [28]. The algorithm is superior to most heuristics in execution efficiency and is suitable for formulating emergency strategies. Compared with other intelligent optimization algorithms, ALNS algorithm has relatively loose requirements for the optimization model, and has no differentiable requirements for the objectives of the optimization model. The setting of optimization objectives basically does not affect the efficiency of the algorithm itself. At the same time, corresponding operators can be designed according to specific optimization problems. Different search strategies can be realized by neighborhood operations of different operators. In the optimization process, through the adaptive selection mechanism of a given operator, the performance of the operator can be recorded and scored in the ALNS algorithm, and then the selection probability of the operator in the next iteration can be determined through the feedback mechanism. The operator with better performance is more likely to be selected for neighborhood operation. By adding a random selection operator, the algorithm can avoid falling into local optimal solution to a great extent.

ALNS algorithm is described in Table 3.

TABLE 3. Adaptive Large Neighborhood Search Algorithm.

Algorithm Adaptive Large Neighborhood Search

Construct a feasible solution x by Greey; set $x^b = x$
repeat
  Choose a destroy neighborhood d and a repair neighborhood r using roulette wheel selection based on previously obtained scores π_j
  Generate a new solution x^t from x using the heuristics corresponding to the chosen destroy and repair neighborhoods
  if x^t can be accepted **then**
    $x = x^t$
  end if
  if $c(x^t) < c(x)$ **then**
    $x^b = x^t$
  end if
  Update scores π_j of d and r
until Stop criteria is met
return x^b

TABLE 4. Nearest vehicle dispatch algorithm.

Algorithm Nearest Vehicle Dispatch

Initial an empty solution s
for **i** $\in N$
  according to the distance between i and vehicle k, ascending sort vehicle set M;
  for k $\in$ M
    find the bestPath for inserting i into k (meet the constraint);
    if bestPath $\neq$ null **then**
      update the vehicle k's path in s;
    end if
  end for
return s

B. IMPROVED INITIAL SOLUTION CONSTRUCTION

As a heuristic search algorithm, ALNS algorithm needs to construct the initial solution and perform iterative optimization in neighborhood space. The traditional greedy construction method has the problems of slow construction and unsatisfactory effect. J. Jaeyong [29] proposed two commonly used construction heuristic algorithms as the initial solution, namely the Nearest Vehicle Dispatch heuristic algorithm (NVD) and the Best Insertion algorithm (BI) heuristic. The two algorithms mainly use the idea of greed to construct the initial solution.

After encoding the algorithm and trying different initial solutions, it is found that ALNS has no strong dependence on the initial solution, that is, there is no obvious correlation between the quality of the solution and the selection method of the initial solution. Therefore, NVD algorithm which can quickly obtain the feasible solution is selected as the initial solution of ALNS algorithm.

The general flow of NVD algorithm is shown in Table 4: For each station, find the vehicle closest to the point and the best insertion position of the station satisfying the constraints on the existing path of the vehicle. If the vehicle cannot be given a feasible path, find the next nearest vehicle for the same calculation, and so on until a suitable vehicle and path are found. If all vehicles cannot meet the constraints, transfer to the next station.

C. OPERATOR DESIGN

1) DESTROY OPERATORS

In ALNS, the destroy operator can randomly destroy part of the current solution [30]. In this paper, three destruction operators (random destruction heuristic, worst cost destruction heuristic and Shaw destruction heuristic) are used. Given a current solution s and the number of damaged sites q, it is estimated that customer points will be randomly selected from the current solution s according to the rules of different operators to remove and update the current solution until a new solution is obtained after q sites are removed.

(1) Random destruction heuristic, which randomly selects n sites from the current solution and removes them. This operator has no sorting rules and often produces a poor set of removal points, but it helps to disperse the neighborhood search and avoid falling into local optimization.

(2) Worst cost destruction heuristic. Remove n stations with large increments by calculating the increment of stations to path cost. The increment is calculated by Equation (18).

$$c_{m,r} = D(r) - D_{-m}(r) \tag{18}$$

where $c_{m,r}$ represents the increment of station m to r, r represents the current path of point m, $D(r)$ represents the cost value of the current path, and $D_{-m}(r)$ represents the path cost value assuming that the path removes the customer point m. This equation means that the current path cost of customer point m minus the path cost assuming that the path removes point m. The worst-case removal operator arranges the site set S in descending order according to its increment, and then selects the customer points in turn for removal.

(3) Shaw destruction heuristic. The core idea of this operator is to remove similar sites, because similar points are easier to regroup and may yield more solutions. If you choose to delete very different sites, you are likely to restore them to the original location (because this is the only feasible repair solution) or to a poor location when repairing the site.

When the operator circulates, it first needs to randomly select a customer point I from s, and then arrange the sites in s in descending order according to the similarity with point I. In this operator, the similarity of stations is calculated as follows:

$$L(i,j) = \varphi d_{ij} + \kappa \left|T_{ik} - T_{jk}\right| + \omega \left(1 - \frac{\left|K_i \bigcap K_j\right|}{min\left\{\left|K_i\right|, \left|K_j\right|\right\}}\right) \tag{19}$$

where d_{ij} represents the distance between i point and j point, T_{ik} indicates the time when vehicle k starts service at site i, and K_i represents the properties of site i. $L(i,j)$ consists of three parts: distance gap, time window gap and attribute gap, respectively corresponding to the weights φ, κ and ω. The greater the $L(i,j)$, the greater the difference between point i

and point j. Therefore, the stations in S are arranged in the ascending order according to this value in the Shaw removal operator.

2) REPAIR OPERATORS

In ALNS, the repair operator reconstructs the part destroyed by the destroy operator to realize neighborhood search, so as to obtain a set of solutions [30]. At the same time, it is necessary to judge whether the reconstructed route satisfies the model constraints. In this paper, four repair operators are used: random repair heuristic, greedy repair heuristic, regret-k repair heuristic and best repair heuristic.

(1) Random repair heuristic. The operator is calculated serially. In each iteration, a point is randomly selected from the set of idle stations, and then a route is randomly selected from the current solution. A path satisfying the constraints is found at this point for insertion. If there is no such path, the next route is randomly selected to find a feasible solution. If there is no feasible path, the site will not be serviced. If all sites are processed, the algorithm ends. The addition of random repair operator makes the search process diversified to avoid falling into local optimization.

(2) Greedy repair heuristic. Insert one site per iteration, which is determined by finding the site with the smallest global insertion increment. The increment is calculated as (20).

$$c_{m,r} = D_{+m}(r) - D(r) \tag{20}$$

where $D_{+m}(r)$ represents the increased cost after site m is inserted into path r, and $D(r)$ represents the cost of the original path. Then, Equation (21) is defined to find the global minimum.

$$c_m = min_{r \in R} c_{m,r} \tag{21}$$

where r represents the path set corresponding to all vehicles, and c_m means the global minimum insertion cost of the insertion site m and its corresponding insertion location. The site i selected in the iteration process passing $min_{r \in R} c_i$ decide to insert the point according to its global optimal position. The iteration terminates if all sites are inserted, or none can be inserted.

(3) Regret-k Repair Heuristic. On the basis of greedy repair heuristic, the site to be processed is selected according to the regression value, the first k optimal paths are considered, and the site with the largest gap between the optimal path and other $k - 1$ paths is found. Compared with greedy, the core idea of this operator is to select the site with the largest regret cost if it is not inserted this time. Equation (22) obtains the stations corresponding to the regret values, and inserts them one by one.

$$Regret_i = \arg max_{i \in N} \left\{ \sum_{j=1}^{k} \left(c_{i,z_{ij}} - c_{i,z_{i1}} \right) \right\} \tag{22}$$

Among them, $Regret_i$ represents the regression value of customer point i selected in each iteration, z_{ij} represents the path with the j smallest insertion cost for customer point i among all vehicle paths. Comparatively, when k increases, it can be found earlier that the possibility of inserting a customer point with advantages becomes limited.

(4) Best Repair Heuristic. It is carried out in a serial mode, similar to BI. The operator processes the idle sites one by one, to find the path with the smallest insertion increment that satisfies the constraints for each idle site.

D. OPERATOR SELECTION OPTIMIZATION

1) MODIFIED CHOICE FUNCTION (MCF)

The Modified Choice Function is an effective extension of the original Choice function proposed by Drake for Cowling [25] in 2012 [26], with the idea of dynamically controlled heuristic selection based on three different measures. So, if the operator is selected, it must have a higher score F_t.

The first measure f_1 reflects the past performance of each heuristic, as shown in (23).

$$f_1\left(h_j\right) = \sum_n \phi^{n-1} \frac{I_n\left(h_j\right)}{T_n\left(h_j\right)} \tag{23}$$

where $I_n\left(h_j\right)$ represents the increment of the objective function, $T_n\left(h_j\right)$ represents the time required for heuristic h_j to call n to generate a solution, and ϕ represents a parameter in the interval [0,1], highlighting the recent performance.

The second measure f_2 tracks a pair of heuristics (h_k, h_j) by considering their past performance in continuous selection, as shown in (24):

$$f_2\left(h_j\right) = \sum_n \phi^{n-1} \frac{I_n\left(h_k, h_j\right)}{T_n\left(h_k, h_j\right)} \tag{24}$$

where $I_n\left(h_k, h_j\right)$ represents the increment of the objective function, $T_n\left(h_k, h_j\right)$ indicates the time required to call the heuristic h_j immediately after h_k to call n to generate the solution.

The third measure f_3 records the time required since the heuristic h_k was last invoked $\tau\left(h_j\right)$. This heuristic algorithm, which is inactive for a certain period of time, provides a choice.

$$f_3\left(h_j\right) = \tau\left(h_j\right) \tag{25}$$

Therefore, the correction function is as Equation (26):

$$f_t\left(h_j\right) = \phi_t f_1\left(h_j\right) + \phi_t f_1\left(h_j\right) + \delta_t f_3\left(h_j\right) \tag{26}$$

where t represents the number of calls to the heuristic h_j, and indicates that the initiation method used has been improved. Measures f_1 and f_2 strengthen the search process, and the measure f_3 diversifies the search by providing inactive heuristics that can be selected. The above can be achieved by combining parameters ϕ_t and δ_t, among which, ϕ_t is a strengthening parameter, weighting f_1 and f_2 respectively, δ_t is relative to f_3, so it is defined to control the degree of diversification. In each iteration, if the target value increases,

TABLE 5. Modified adaptive large neighborhood search with NVD algorithm.

Algorithm Modified Adaptive Large Neighborhood Search with NVD
1. Construct a feasible solution x by **NVD**; set $x^b = x$
2. **repeat**
3. Choose a destroy neighborhood d and a repair neighborhood r using the modified choice function based on the obtained scores F_t
4. Generate a new solution x^t from x using the heuristics corresponding to the chosen destroy and repair neighborhoods
5. **if** x^t can be accepted **then**
6. $x = x^t$
7. **end if**
8. **if** $c(x^t) < c(x)$ **then**
10. $x^b = x^t$
11. **end if**
12. Update scores π_j of d and r
13. **until** Stop criteria is met
14. **return** x^b

ϕ_t increases and δ_t decreases. On the contrary, when the target value is not improved, ϕ_t decreases, δ_t increases. Parameters ϕ_t and δ_t are expressed as follows:

$$\phi_t(h_t) = \begin{cases} 0.99, & \textit{if the objective value improves} \\ max\{\phi_t - 0.01, 0.01\}, & \textit{otherwise} \end{cases} \tag{27}$$

$$\delta_t(h_t) = 1 - \phi_t(h_t) \tag{28}$$

2) THE MODIFIED ADAPTIVE LARGE NEIGHBORHOOD SEARCH WITH NVD (MALNSN)

The core idea of the Modified Adaptive Large Neighborhood Search with NVD algorithm (MALNSN) is to explore the search neighborhood space efficiently using more large fields. In order to make the algorithm more efficient, NVD algorithm is used as the initial solution. In selecting damage and repair operators, the core idea of ALNS is to track the score of each operator in the iterative process by using score π_j to balance the performance of the operator, and to select a probability of $\frac{\pi_j}{\sum_i \pi_i}$ operator J using the roulette mechanism. Although in each iteration, the score of each operator φ_j will be updated and the probability of the selection operator will be recalculated according to the score, and many iterations must be made during the search to adjust the score of the random operator and jump out of the local optimum. In order to further balance the diversity and efficiency of the search neighborhood field, the roulette selection mechanism of the original ALNS is no longer used, and the modified choice function (MCF) is combined with the operator selection stage of ALNS algorithm, while preserving the whole search process.

MALNSN algorithm is shown in Table 5:

E. COMPLEXITY ANALYSIS

According to the algorithm description in Table 5, the MALNSN algorithm is mainly divided into two steps: the first step is to construct the initial solution by using NVD algorithm. Assuming that the number of virtual stations is N and the number of scheduled vehicles is M, according to the NVD algorithm description given in Table 4, it can be seen that in the double cycle, the total execution times of the first layer cycle is the number of virtual stations and the total execution times of the second layer cycle is the number of scheduled vehicles. Therefore, the time complexity of the first step is $O(N \cdot M)$. The second step is iterative optimization. The optimization process is essentially a double cycle, and the number of cycles needs to be specified, which is assumed to be C. In the process of using MCF to find the operator inside the loop, the execution times are $N \log N$, so the time complexity of the second step is $O(C \cdot N \log N)$. Therefore, the time complexity of MALNSN algorithm is $O(N \cdot (M + C \cdot \log N))$.

V. EXPERIMENTAL ANALYSIS

In order to further verify the effectiveness of the model, artificial data experiments and simulation experiments were conducted on MALNSN algorithm with reference to Solomon's benchmark [31]. MALNSN algorithm is compared with the unmodified ALNS algorithm using artificial data. In the simulation experiment, it is compared with PSO-GA, PSO-ACO, TS, SA and ALNS algorithms to prove its effectiveness.

A. ENVIRONMENT SETTINGS

This experiment was carried out on a laptop with 11th Gen Intel (R) core (TM) i5-1135g7 @ 2.40GHz and 16GB ram windows 11, and the JDK version is 1.8.0_291.

B. MODEL PARAMETER SETTING

1) BASIC PARAMETERS

Considering the actual situation, in the model parameters, the rated maximum passenger capacity of the responsive bus is $M = 50$, the speed is $v = 30$ in the idle state (i.e. the speed from parking to the first virtual station), and the average operating speed when carrying passengers is $v = 25$. In the vehicle operation cost Equation (8), the unit departure cost is set as $c_1 = 80$, the unit time operation cost as $c_2 = 60$, and the unit distance operation cost as $c_3 = 1.8$. In Equation (9) of passenger traffic cost, $p_1 = p_2 = 0.5$.

2) FITNESS WEIGHT

In the bus route optimization model proposed in this paper, Equation (7) represents the optimization objective of the model, in which the setting of each weight coefficient has an important impact on the overall model. In order to obtain more effective parameters, this paper evaluates the model according to different parameters settings. Changes in fitness values under different parameters are shown in the Figure 2.

According to the experimental results, in Equation (7), the model parameters are $\alpha = 0.6$, $\beta = 0.8$, and a better solution can be obtained. Since the sum of α and β is 1, the parameters are set to 0.44 and 0.56.

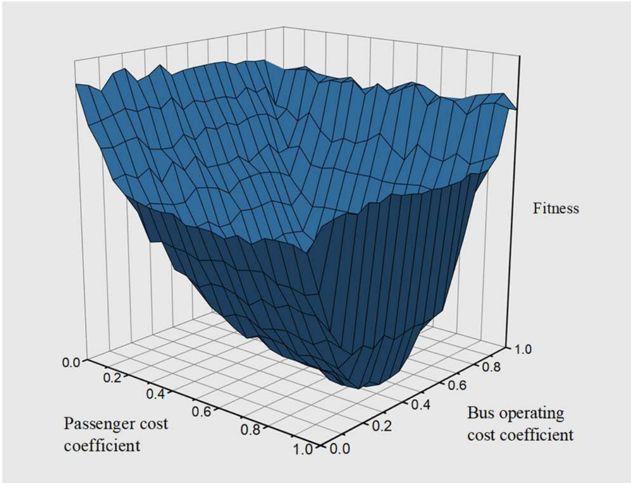

FIGURE 2. Change trend graph of fitness value.

C. ALGORITHM EFFECTIVENESS ANALYSIS

In order to test the performance of MANLSN, this paper refers to the instances of Solomon's benchmark [31] and Homberger's benchmark [32] to test the algorithm and ALNS algorithm through a group of examples. Both algorithms adopt the destroy rule and repair rule mentioned above, in which the K value of the regression-k repair heuristic operator is set to 2, and 10 experiments are carried out for each instance. Each experiment runs 20000 iterations, and the results are averaged.

Table 6 compares the target value and execution time of MALNSN and ALNS algorithms in the selected Solomon instance group. The first column represents the instance group, where set R contains random customer problems, set C contains cluster customer problems, and set RC contains random customers and cluster customer problems. The second and third columns represent the target values and algorithm execution results obtained from the initial solution of ALNS algorithm. The fourth and fifth columns show the target values and algorithm execution results obtained from the initial solution of MALNSN algorithm. The sixth and seventh columns respectively represent the time gain average percentage (Gap) of MANNSN of the original structure with the initial solution method and the MANNSN using NVD to construct the initial solution. In the table, the calculated time gain average percentage (Gap) is expressed as the absolute difference between MALNSN and ALNS execution time divided by ALNS execution time.

It can be seen from the Table 6 that MALNSN method is superior to the ALNS method on a whole with better results that the average gain percentage can reach 17.11%. Since the modified choice function adopts three different measures to select the operator, and the roulette mechanism uses the recent performance of the operator to select the most appropriate operator, MALNSN requires more execution time, but with the increase in instance size, the execution time of this

TABLE 6. Experimental comparison data of MALNSN algorithm and ALNS algorithm.

Instance	ALNS		MALNSN		Gap(%)	
	Initial solution	ALNS solution	Initial solution	MALNSN solution	Original initialization	NVD initialization
R101	1747.12	1650.80	1742.02	1645.79	22.82	20.34
C101	929.21	828.94	899.21	828.94	21.84	19.98
RC101	1793.01	1708.80	1781.01	1701.21	22.50	20.23
R201	1310.64	1253.23	1310.54	1253.23	20.85	19.29
C201	682.52	591.56	682.32	591.56	19.72	18.21
RC201	1516.79	1406.94	1517.02	1406.94	20.48	18.86
R121	4896.01	4819.12	4894.72	4805.74	15.91	14.91
C121	2807.20	2704.57	2804.14	2689.87	15.86	14.86
RC121	3725.27	3606.06	3717.25	3587.78	15.63	14.63
R221	4603.41	4513.10	4581.72	4503.84	15.47	14.73
C221	2063.76	1931.44	2064.62	1912.46	15.04	13.97
RC221	3681.32	3605.40	3653.61	3597.64	15.66	14.34
R141	10741.56	10639.75	10750.27	10423.68	12.73	11.84
C141	7308.45	7152.06	7205.15	7137.58	11.96	11.12
RC141	9682.31	9127.15	9685.71	8712.49	12.57	11.87
R241	9982.12	9758.46	9982.38	9553.65	11.95	11.23
C241	4845.71	4116.33	4849.20	3910.53	11.94	11.62
RC241	7985.49	7471.01	7971.18	6958.34	11.55	10.95
R161	23027.25	22838.65	23027.49	22132.63	8.99	8.34
C161	14306.51	14095.64	14307.45	13592.87	8.97	8.28
RC161	18512.35	17924.88	18511.72	17214.93	8.56	8.43
R261	22210.38	21945.30	22212.09	20932.66	8.94	8.34
C261	8324.17	7972.98	8313.60	7756.57	8.93	8.37
RC261	14968.21	14817.72	14959.17	14089.38	8.88	8.23
R181	39892.48	39315.30	39882.18	38315.30	4.91	4.89
C181	25773.19	25183.22	25663.32	25183.22	4.95	4.86
RC181	32630.95	32266.62	32635.95	31316.62	4.91	4.81
R281	39260.42	33164.93	39253.42	30641.13	4.92	4.83
C281	12322.29	11687.06	12124.65	11687.06	4.93	4.75
RC281	26302.25	22117.73	26302.27	22116.08	4.90	4.73
R1101	58014.95	56652.13	57960.76	55242.34	1.93	1.78
C1101	43154.79	42583.21	43146.78	42468.22	1.99	1.95
RC1101	49389.80	48326.63	49383.19	47567.14	1.97	1.80
R2101	46157.92	45624.33	46125.26	43996.32	1.98	1.79
C2101	17533.88	16872.96	17529.68	16879.24	1.97	1.93
RC2101	35918.22	35271.37	35872.99	33106.37	1.92	1.85

algorithm will converge to that of the ALNS algorithm. Although the proposed NVD algorithm is basically the same as the original algorithm in constructing the initial solution, the Gap can be narrowed by constructing the initial solution. In this experiment, the average Gap of the initial solution constructed with NVD is 93.25% of that without NVD. Therefore, using NVD instead of the original algorithm to construct the initial solution can effectively improve the execution efficiency of the algorithm, which is conducive to the formulation of emergency evacuation strategy.

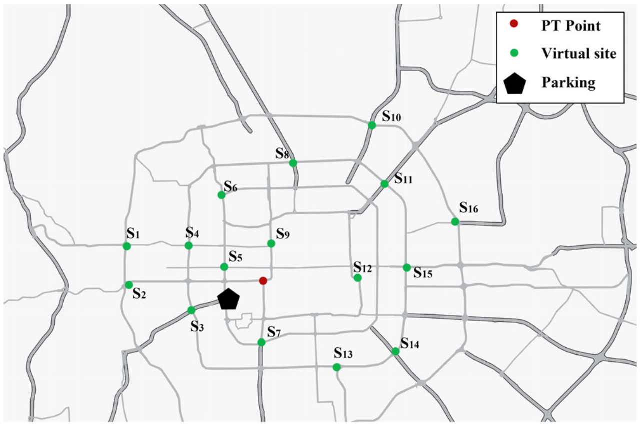

FIGURE 3. Traffic network disruption information.

D. SIMULATION EXPERIMENT

1) EXPERIMENT ENVIRONMENT SETTING

Taking an operation interruption in a road section of Beijing urban road network as an example, the operation interruption time is 2 hours to obtain the space-time needs of passengers. Based on Solomon's benchmark [31], the passenger target is clustered into 16 virtual stations through the passenger information clustering model, as shown in the Figure 3.

In each virtual site, the site time window is calculated by (29) and (30).

$$T_{S_i}^{Arr} = minT_i^{Arr}, \quad i \in S_i \tag{29}$$

$$T_{S_i}^{Dep} = maxT_i^{Dep}, \quad i \in S_i \tag{30}$$

where S_i represents the *i-th* virtual site, T_i^{Arr} represents the earliest expected arrival time of a vehicle per passenger at the virtual site S_i, T_i^{Dep} represents the latest expected departure time for each passenger in the virtual site S_i. The service time of each passenger is set as 2s in the experiment. The service time and time window information of each station are shown in Table 7.

TABLE 7. Virtual site information.

Site number	DEMAND	Ready time	Due time	Service time
S1	33	890	989	66
S2	24	816	879	48
S3	16	55	156	32
S4	30	703	806	59
S5	19	15	60	38
S6	28	559	764	56
S7	26	10	77	51
S8	25	250	329	51
S9	27	489	650	55
S10	25	361	406	49
S11	31	55	154	63
S12	20	647	726	41
S13	24	30	95	49
S14	23	571	616	46
S15	23	392	421	46
S16	32	478	525	64

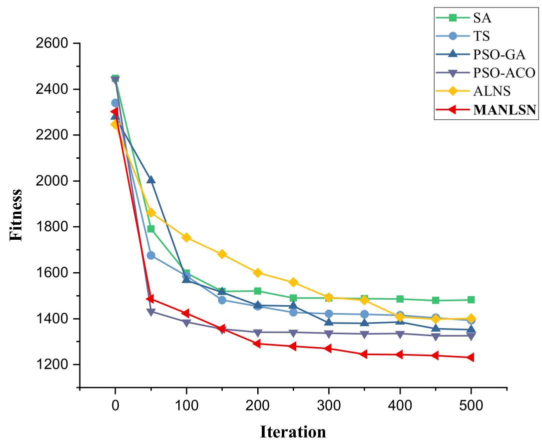

FIGURE 4. Objective function value comparison.

2) ANALYSIS OF EXPERIMENTAL RESULTS

We simulate each algorithm for 10 times and calculate the objective function value of the optimal scheduling strategy obtained by each algorithm to analyze the performance of the proposed algorithm. In 10 experiments, every 50 iterations are recorded for 500 iterations. The average value of the recorded target value is shown in Figure 4.

This paper takes the bus operation cost and passenger traffic cost as the objective function to make them better as a whole, so as to reflect the effective performance of the algorithm. It can be analyzed from Figure 4 that in this simulation example, the MALNSN algorithm proposed in this paper reduces the target values by 16.96%, 11.59%, 8.98%, 7.01%, and 12.80% respectively compared with SA, TS, PSO-GA, PSO-ACO and ALNS algorithms. The results show that compared with other algorithms, MALNSN algorithm performs better in target optimization. In the initial stage of iteration, the optimization method of the combination of multiple operators can expand the search range and improve the search ability of the global optimal solution by selecting the correction function, thus improving the iterative efficiency and making the algorithm converge faster. In the middle and later stages of the iteration, the algorithm is easy to fall into local optimization because the whole route tends to be stable. However, the third measure of the modified function can make the algorithm increase the search range using Random operator and find a new route to adapt to the optimization goal when the fitness value tends to be stable.

In this paper, the number of iterations of all algorithms is set to 500, and each algorithm is tested for 10 times. The results are shown in Figure 5. It can be seen that the MALNSN algorithm proposed in this paper can obtain better results than other algorithms in each experiment, and has stronger stability. In addition, compared with the improved ALNS algorithm, MALNSN algorithm has better

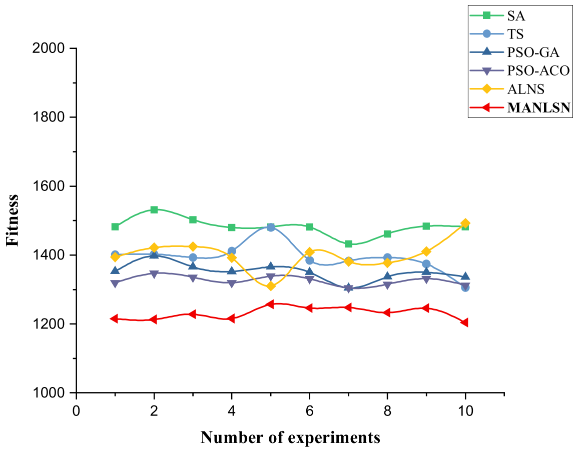

FIGURE 5. The optimization results of each algorithm in 10 experiments.

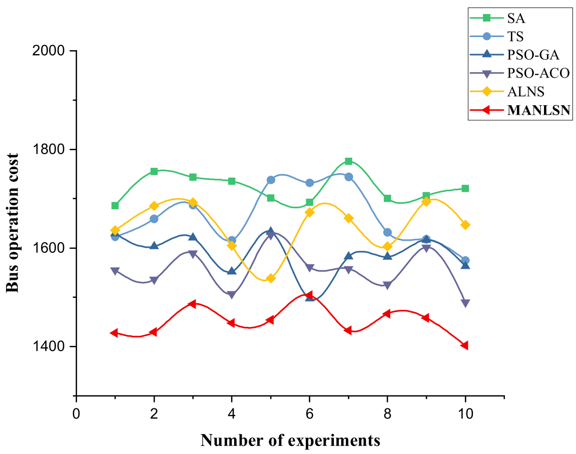

FIGURE 6. Comparison of bus operation cost.

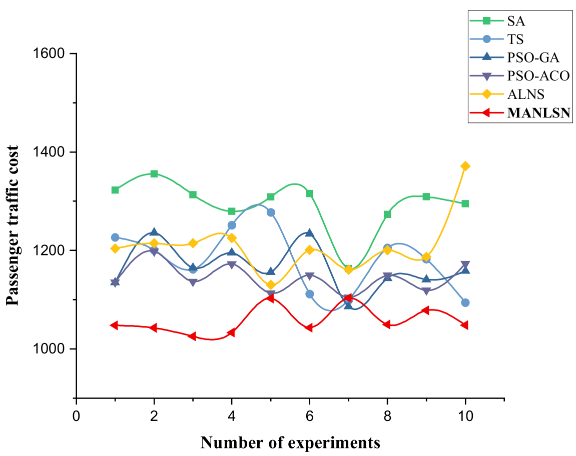

FIGURE 7. Comparison of passenger traffic cost.

performance in both optimization results and algorithm stability.

Through 10 experiments, the bus operation cost and passenger traffic cost are shown in Figure 6 and Figure 7 respectively. In addition, their calculation methods are shown in Equation (8) and Equation (9) respectively.

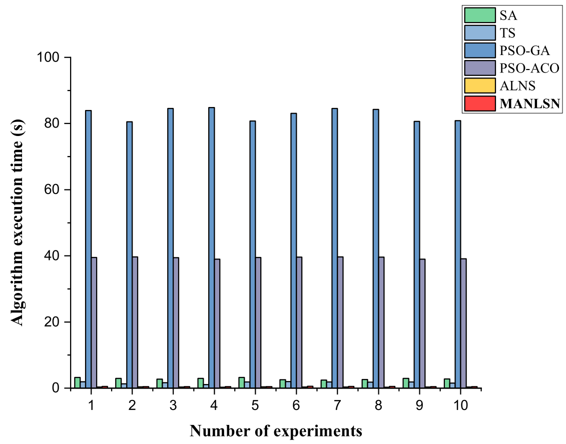

FIGURE 8. The execution time of each algorithm in 10 experiments.

From the experimental results in Figure 6 and Figure 7, we can see that the MALNSN algorithm can not only obtain lower bus operation cost, but also lower passenger traffic cost compared with five heuristic algorithms, which further verifies the multi-objective optimization ability of the MALNSN algorithm.

Under the setting of the experimental parameters, the average time spent by the optimal scheduling strategy of each algorithm is shown in Figure 8.

As can be seen from Figure 8, the MALNSN algorithm, similar to the ALNS algorithm, has a very short execution time compared with other algorithms. Combined with Figure 4, even if PSO-ACO algorithm can get the results with MALNSN, its execution time is about 40 times that of MALNSN algorithm. In traffic interruption, it is very important to find a more effective scheduling strategy as soon as possible. Therefore, MALNSN algorithm has more advantages in dealing with traffic interruption.

During the process of executing the optimization strategy, as the departure times of each algorithm are unknown, the average running time of the scheduled vehicles in the strategy can be used as the standard to evaluate the local optimization of the optimization strategy. In 10 experiments, the average execution time of the optimization strategy generated by each algorithm is shown in Figure 9, and the calculation method is shown in (31).

$$exetime_{Strategy} = AVG\left(T_k\right), \quad k = 1, 2, 3, \ldots \tag{31}$$

where $exetime_{Strategy}$ represents the average running time of the scheduled vehicles in the optimization strategy, T_k represents the time from the departure of the *k-th* vehicle from the parking lot to returning to the parking lot again in the current strategy.

Since the problem solved in this paper is abstracted as VRPTW, and the bus needs to arrive at each station within the specified time window, there is no significant difference in the execution time of the strategies generated by each algorithm. It can be seen from Figure 9 that MALNSN algorithm outperforms other algorithms, and the average of 10 experimental results can be obtained. Compared with SA,

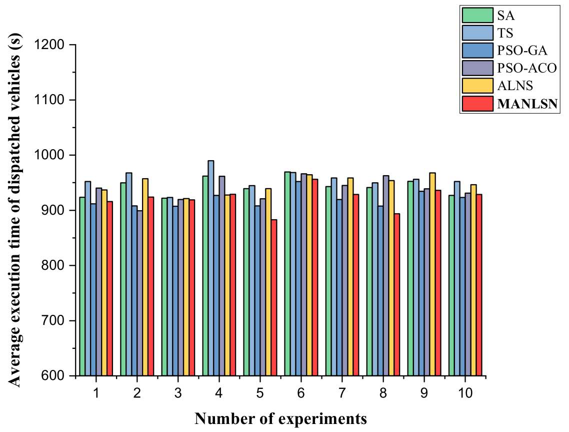

FIGURE 9. Comparison of average running time of scheduled vehicles in optimization strategy.

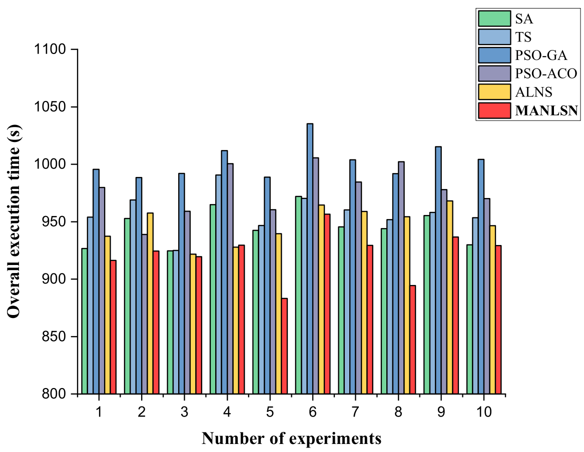

FIGURE 10. Overall time-consuming comparison of optimal scheduling.

TS, PSO-GA, PSO-ACO and ALNS algorithms, MALNSN algorithm reduces the average running time by 2.38%, 3.62%, 0.08%, 1.85% and 2.72% respectively.

In 10 experiments, the overall time consumption of optimal scheduling of each algorithm is shown in Figure 10.

The overall time-consuming of optimal scheduling includes the user submitting data to the platform after the interruption, the platform formulating the vehicle scheduling strategy and the total time required for all vehicles in the strategy to return to the parking lot. The length of time is an important factor in evaluating the service quality of the model. When there are too few departures or stations are unevenly distributed, some users will take longer to reach the target station, affecting the overall service. From the Figure 10, compared with SA, TS, PSO-GA, PSO-ACO and ALNS algorithms, the MALNSN algorithm proposed in this paper saves 2.63%, 3.76%, 8.23%, 5.25% and 2.89%. Of the overall time, shortens the overall arrival time of passengers and improves the overall service quality.

VI. SUMMARY AND NEXT WORK

This paper summarizes the current research situation of solving public transport interruption under the background of intelligent transportation, and puts forward the research ideas combined with the actual lifestyle. Starting from the actual scene of traffic interruption, an adaptive bus line optimization solution strategy is proposed by applying emergency response bus to traffic interruption.

Firstly, in order to improve the fine-grained passenger carrying capacity of public transport and ensure the operation efficiency of emergency public transport, an improved adaptive DBSCAN algorithm is proposed as the passenger information clustering model, taking the influencing factors such as vehicle capacity, passenger travel time window and the number of access stations into consideration. The spatiotemporal demands submitted by passengers are adaptively clustered into data conforming to Solomon's benchmark. Secondly, considering the expectations of operators and passengers, in order to reduce the consumption of transport capacity resources and improve the operation efficiency of public transport, a multi-objective emergency bus route optimization scheduling model and bus route optimization solution strategy are constructed with the goal of minimizing the cost of vehicle operation and passenger traffic. Thirdly, by improving the initial solution generation rule with NVD and operator selection strategy with MCF, an improved adaptive large neighborhood search algorithm (MALNSN) is proposed. The computational experiments are carried out according to the Solomon benchmark. The average benefit of the proposed algorithm is 17.11% compared with the original algorithm. Finally, based on the actual road network and passenger travel time-space demand, the experimental verification is carried out. The results show that compared with the representative algorithms, the MALNSN algorithm proposed in this paper can not only ensure the stability of the algorithm, but also formulate a reasonable path optimization strategy in a short time, reduce the consumption of transportation resources and improve the operation efficiency of public transport.

And compared with other commonly used heuristics, the proposed algorithm MALNSN can select different operators to search the optimal solution in the optimization process. Although the method of selecting operators is improved in this paper, more complex path optimization problems may occur in practical application (such as multiple interruption points, congestion and weather factors in some roads, etc.), and the existing operators cannot find the optimal solution in the early stage of iteration. Therefore, they are the main research contents in future work, such as designing targeted operators for this kind of problems to dynamically adjust the internal parameters of each operator and solving problem of weaker single objective etc. In addition, we will consider using real-time data instead of current data to validate the effectiveness of the algorithm.

ACKNOWLEDGMENT

This research project was conducted by the Key Laboratory of TCM Data Cloud Service of Shandong Management University, the research group on Big-Data-driven abnormal behavior detection technology, and the Ph.D. research fund of Shandong Management University. These authors contributed to the work equally, so both authors are regarded as the first author.

REFERENCES

[1] M. Yap and O. Cats, ''Predicting disruptions and their passenger delay impacts for public transport stops,'' *Transportation*, vol. 8, no. 4, pp. 1703–1731, Apr. 2020, doi: 10.1007/s11116-020-10109-9.

[2] H. Shuang, F. Hui, Z. Jiahong, L. Junzhou, and Z. Weiliang, ''Modelling and simulation of hierarchical scheduling of real-time responsive customised bus,'' *IET Intell. Transp. Syst.*, vol. 14, no. 1, pp. 1615–1625, Nov. 2020, doi: 10.1049/iet-its.2020.0138.

[3] B. A. Kumar, G. H. Prasath, and L. Vanajakshi, ''Dynamic bus scheduling based on real-time demand and travel time,'' *Int. J. Civil Eng.*, vol. 17, no. 9, pp. 1481–1489, Sep. 2019, doi: 10.1007/s40999-019-00445-y.

[4] E. Raafat and A. Hadeer, ''A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants,'' *Comput. Ind. Eng.*, vol. 140, Feb. 2020, Art. no. 106242, doi: 10.1016/j.cie.2019.106242.

[5] W. Gu, J. Yu, Y. Ji, Y. Zheng, and H. M. Zhang, ''Plan-based flexible bus bridging operation strategy,'' *Transp. Res. C, Emerg. Technol.*, vol. 91, pp. 209–229, Jun. 2018.

[6] J. D. Wang, Z. Z. Yuan, and S. B. Ning, ''Optimization model of emergency bus dispatching in response to operational disruptions of urban rail transit,'' *J. Transp. Syst. Eng. Inf. Technol.*, vol. 19, no. 4, pp. 149–154+163, 2019, doi: 10.16097/j.cnki.1009-6744.2019.04.022.

[7] Y. J. Zheng, G. U. Wei, J. I. Yu-Xiong, and H. J. Zhang, ''Optimizing a bus bridging strategy considering efficiency and equity,'' *J. Transp. Syst. Eng. Inf. Technol.*, vol. 19, no. 2, pp. 94–101, 2019, doi: 10.16097/j.cnki.1009-6744.2019.02.014.

[8] M. B. D. Galarza, S. Kenneth, and V. Pieter, ''A large neighborhood search algorithm to optimize a demand-responsive feeder service,'' *Transp. Res. C, Emerg. Technol.*, vol. 127, Apr. 2021, Art. no. 103102, doi: 10.1016/j.trc.2021.103102.

[9] F. Lang, C. Hui, and G. Ying, ''An improved flower pollination algorithm to the urban transit routing problem,'' *Soft Comput.*, vol. 24, no. 7, pp. 5043–5052, Aug. 2019, doi: 10.1007/s00500-019-04253-3.

[10] G. Rongge, Z. Wenyi, G. Wei, and R. Bin, ''Time-dependent urban customized bus routing with path flexibility,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2381–2390, Sep. 2020, doi: 10.1109/tits.2020.3019373.

[11] S. Yang, L. Mingde, Y. Jian, S. Yuhui, and M. Martin, ''A hybrid swarm intelligence algorithm for vehicle routing problem with time windows,'' *IEEE Access*, vol. 8, pp. 93882–93893, 2020, doi: 10.1109/access.2020.2984660.

[12] H. Meiling, W. Zhixiu, W. Xiaohui, and P. Yongtao, ''An adaptive variable neighborhood search ant colony algorithm for vehicle routing problem with soft time windows,'' *IEEE Access*, vol. 9, pp. 21258–21266, 2021, doi: 10.1109/access.2021.3056067.

[13] L. Guoming and L. Junhua, ''An improved tabu search algorithm for the stochastic vehicle routing problem with soft time windows,'' *IEEE Access*, vol. 8, pp. 158115–158124, 2020, doi: 10.1109/access.2020.3020093.

[14] J. Zhang, F. Yang, and X. Weng, ''An evolutionary scatter search particle swarm optimization algorithm for the vehicle routing problem with time windows,'' *IEEE Access*, vol. 6, pp. 63468–63485, 2018.

[15] M. Yannis, M. Magdalene, and M. Athanasios, ''A multi-adaptive particle swarm optimization for the vehicle routing problem with time windows,'' *Inf. Sci.*, vol. 481, pp. 311–329, May 2019, doi: 10.1016/j.ins.2018.12.086.

[16] S. Gaurav, S. Alok, and M. Rammohan, ''NSGA-II with objective-specific variation operators for multiobjective vehicle routing problem with time windows,'' *Expert Syst. Appl.*, vol. 176, Mar. 2021, Art. no. 114779, doi: 10.1016/j.eswa.2021.114779.

[17] C. Chen-Yang, Y. Kuo-Ching, L. Chung-Cheng, Y. Chumpol, and C. Wan-Chen, ''An auction bidding approach to balance performance bonuses in vehicle routing problems with time windows,'' *Sustainability*, vol. 13, p. 9430, May 2021, doi: 10.3390/su13169430.

[18] W. Jiahai, R. Wenbin, Z. Zizhen, H. Han, and Z. Yuren, ''A hybrid multiobjective memetic algorithm for multiobjective periodic vehicle routing problem with time windows,'' *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 11, pp. 4732–4745, Nov. 2020, doi: 10.1109/tsmc.2018.2861879.

[19] J. Yang, W. Xin, and C. Yao, ''Designing a flexible catering system for high-speed railway considering departure time selection and time deadline constraints,'' *IEEE Access*, vol. 8, pp. 44300–44317, 2020, doi: 10.1109/access.2020.2978180.

[20] J. Chen, B. Dan, and J. Shi, ''A variable neighborhood search approach for the multi-compartment vehicle routing problem with time windows considering carbon emission,'' *J. Cleaner Prod.*, vol. 277, Dec. 2020, Art. no. 123932, doi: 10.1016/j.jclepro.2020.123932.

[21] G. D. Konstantakopoulos, S. P. Gayialis, E. P. Kechagias, G. A. Papadopoulos, and I. P. Tatsiopoulos, ''A multiobjective large neighborhood search Metaheuristic for the vehicle routing problem with time windows,'' *Algorithms*, vol. 13, no. 10, p. 243, Sep. 2020.

[22] E. O. Hellsten, D. Sacramento, and D. Pisinger, ''An adaptive large neighbourhood search heuristic for routing and scheduling feeder vessels in multi-terminal ports,'' *Eur. J. Oper. Res.*, vol. 287, no. 2, pp. 682–698, Dec. 2020.

[23] P. Binbin, Z. Zhenzhen, and L. Andrew, ''A hybrid algorithm for time-dependent vehicle routing problem with time windows,'' *Comput. Oper. Res.*, vol. 128, Dec. 2020, Art. no. 105193, doi: 10.1016/j.cor.2020.105193.

[24] S. Ropke and D. Pisinger, ''An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows,'' *Transp. Sci.*, vol. 40, no. 4, pp. 455–472, 2006.

[25] P. Cowling, G. Kendall, and E. Soubeiga, ''A hyperheuristic approach to scheduling a sales summit,'' in *Proc. Int. Conf. Pract. Theory Automated Timetabling*, 2000, pp. 176–190.

[26] J. H. Drake, E. Zcan, and E. K. Burke, ''An improved choice function heuristic selection for cross domain heuristic search,'' in *Proc. 12th Int. Conf. Parallel Problem Solving Nature*, 2012, pp. 1–5.

[27] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, ''A density-based algorithm for discovering clusters in large spatial databases with noise,'' in *Proc. AAAI*, 1996, pp. 1–6.

[28] P. Shaw, *Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems*. Berlin, Germany: Springer, 1998.

[29] J. Jaeyoung, R. Jayakrishnan, and P. J. Young, ''Dynamic shared-taxi dispatch algorithm with hybrid-simulated annealing,'' *Comput.-Aided Civil Infrastruct. Eng.*, vol. 31, no. 4, pp. 275–291, 2015, doi: 10.1111/mice.12157.

[30] D. Sacramento, D. Pisinger, and S. Ropke, ''An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones,'' *Transp. Res. C, Emerg. Technol.*, vol. 102, pp. 289–315, May 2019.

[31] M. M. Solomon, ''Algorithms for the vehicle routing and scheduling problems with time window constraints,'' *Oper. Res.*, vol. 35, no. 2, pp. 254–265, 1987.

[32] H. Gehring and J. Homberger, ''A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows,'' in *Proc. EUROGEN*, 1999, pp. 57–64.

RUISONG LIU was born in Jinan, Shandong, in 2000. He is currently pursuing the degree with the School of Information Engineering, Shandong Management University, majoring in software engineering, with strong hands-on development ability. During the University, he actively participated in the research project of his tutor, published one article, and won many awards in the national computer competition. His main research interests include cloud computing and intelligent transportation.

NING WANG was born in Yantai, Shandong, in 1978. She received the B.E. degree in computer science and technology from the Shandong University of Technology, Zibo, China, in 2003, the M.Sc. degree in computer software and theory from Yunnan Normal University, Kunming, China, in 2006, and the Ph.D. degree in communication and information system from the University of Science and Technology Beijing, Beijing, China, in 2015. She is currently a Teacher with the School of Information Engineering, Shandong Management University. She is also a Professor. Her research interests include cloud computing and big data analysis.