Modeling and Evolutionary Optimization for Multi-objective Vehicle Routing Problem with Real-Time Traffic Conditions

Long Xiao[1, 2], Changhe Li[1, 2, *], Junchen Wang[1, 2], Michalis Mavrovouniotis[3], Shengxiang Yang[4], Xiaorong Dan[5]

[1] School of Automation, China University of Geosciences, Wuhan 430074, China

[2] Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems, Wuhan 430074, China

[3] KIOS Research and Innovation Center of Excellence and Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, 2109, Cyprus

[4] School of Computer Science and Informatics, De Montfort University, Leicester, LE1 9BH, United Kingdom

[5] State Grid Electric Power Research Institute Wuhan Nanrui Co., Ltd., Wuhan 430074, China

xiaolong_cug@qq.com, lichanghe@cug.edu.cn

ABSTRACT

The study of the vehicle routing problem (VRP) is of outstanding significance for reducing logistics costs. Currently, there is little VRP considering real-time traffic conditions. In this paper, we propose a more realistic and challenging multi-objective VRP containing real-time traffic conditions. Besides, we also offer an adaptive local search algorithm combined with a dynamic constrained multi-objective evolutionary framework. In the algorithm, we design eight local search operators and select them adaptively to optimize the initial solutions. Experimental results show that our algorithm can obtain an excellent solution that satisfies the constraints of the vehicle routing problem with real-time traffic conditions.

CCS Concepts

•**Computing methodologies → Artificial intelligence → Search methodologies → Discrete space search**

Keywords

Vehicle Routing Problem; Local Search; Multi-objective Optimization; Constrained Optimization.

1. INTRODUCTION

In 2017, China's total social logistics expenditure accounted for 14.6% of GDP. Every 1% diminution in the total logistics expenses in GDP will reduce hundreds of billions of costs. And with the rapid development of the logistics industry, the vehicle routing problem (VRP) has attracted more and more attention.

Due to different application scenarios, many variations of VRP appeared, such as capacitated VRP (CVRP) [1], VRP with time windows (VRPTW) [2], and so on. But, in the actual distribution process, the dynamic change of traffic conditions is one of the main challenges for planning vehicle routes since traffic conditions directly affect the driving time of vehicles on the road network, thus causing many emergencies. If a vehicle arrives at a delivery location later than the customer's specified latest time, customer satisfaction is significantly reduced. Existing studies that considered traffic factors simply used a random function or periodic random function to generate traffic conditions [3]. However, the traffic factors generated by the stochastic method are far from the traffic conditions in the actual road network since the traffic flow between the road sections is interrelated. In order to make the VRP model closer to the actual engineering situation, this paper proposes an unprecedented VRP that combines the road network topology and the multi-objective VRPTW with a soft time window (MOVRPRTC). The road network data in this model come from the road traffic network of Guanggu in Wuhan, and the order data come from the real online shopping order data after desensitization.

Heuristic algorithms are the only feasible ways to solve practical VRP since exact algorithms could not solve complex VRP in an acceptable time [4]. The heuristic search algorithms applied in VRP mainly focus on genetic algorithm [5,6], ant colony optimization [7], adaptive large neighborhood search [8,9]. Through the investigation of these methods, we found one disadvantage that almost no algorithm adopts any specific mechanisms to handle constraint conditions, i.e., they directly discard infeasible solutions during iterations. In this case, the algorithm can only search in the feasible region, and its exploration ability will be extremely limited. However, there may be better quality solutions in the space outside the feasible region. If we keep these solutions, some operations can be carried out to pull them back to the feasible region and meanwhile keep their quality. To address this disadvantage, we convert the two objectives in MOVRPRTC into constraints first and then make use of a dynamic constrained multi-objective optimization algorithm framework (DCMOEA) [10] to handle constraints. It turns out this method greatly enhances the selection pressure in the multi-objective optimization.

ICMLC'20, February 15–17, 2020, Shenzhen, China.

DOI: http://dx.doi.org/10.1145/12345.67890

* The corresponding author.

There are three contributions to this paper. Firstly, realistic data-based vehicle routing problem considering road network constraints and real-time traffic conditions is constructed, which can be used as test cases for future researchers. Secondly, the dynamic constraint multi-objective optimization framework is applied to the combinatorial optimization problem, which provides a new idea of constraint treatment in the combinatorial optimization problem. Thirdly, several productive local optimizers with adaptive selection schemes are designed, which proved to be very effective in solving our proposed MOVRPRTC.

The remaining sections of this paper are organized as follows. Section 2 describes the modeling method of MOVRPRTC, including the road network topology construction and the vehicle routing problem formulation. Section 3 introduces our proposed optimization algorithm for MOVRPRTC, including the application of dynamic constraint multi-objective evolutionary algorithm framework in VRP and the adaptive local search mechanism. Experimental results are shown in Section 4. Finally, the conclusion is given in Section 5.

2. MODELING OF MOVRPRTC

2.1 Construction of Road Network

In MOVRPRTC, routes can only exist on real roads of the traffic network. The first crucial step of road network construction is to determine the connection between coordinate points in the road. Take the junction shown in Figure 1 as an example. The junction is represented by four coordinate points 4, 5, 8, and 9, which are shared by four roads. Point 4 has two precursor points, i.e., point 1 and 5, and two successor points, i.e., point 3 ad 8. After determining all the connections, the entire topology will be completed.

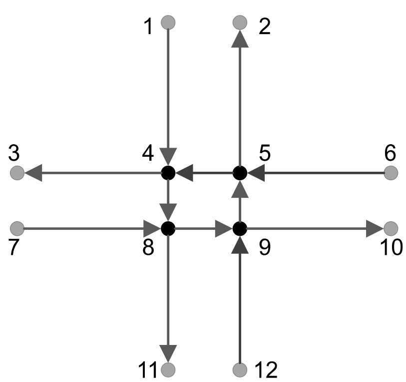

Figure 1. Topology of road junction.

Another critical step of road network construction is to determine the weight variance of each edge with the variance of time. The structure of the road network in a specific area remains almost unchanged in the short term, and most people's daily activities are regular, the traffic conditions vary nearly periodically. So, we can use the traffic trend of any specific day to model the traffic trend. However, considering the difference between the workday and weekend, it is more reasonable to model the traffic trend from the data of a week. In addition, due to some weather anomalies or some other unexpected reasons, the traffic trend may be different on the same day in a different week. So finally, we collected the traffic conditions of the expressway in the Guanggu area within three weeks and used BP neural network to model the traffic trend. Figure 2 shows the traffic trend of a section of the road within 24 hours.

All the data for the construction of the road network come from the Amap traffic status API [11].

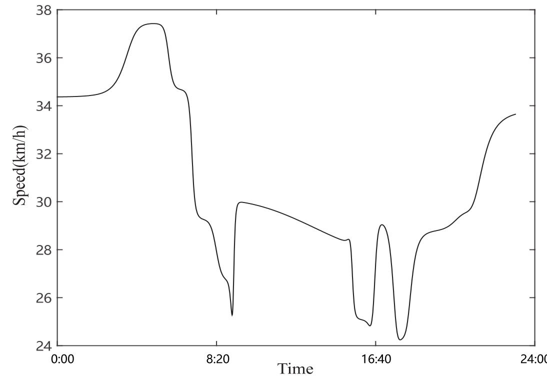

Figure 2. Traffic conditions within 24 hours of a road.

2.2 Formulation of MOVRPRTC

MOVRPRTC is described as follows: in road network topology $G=(V,E)$, $V=\{v_0,\dots,v_N\}$ represents all the N nodes in the road network, $E=\{\langle i,j\rangle | z_{ij}=1\}$ denote all the edges formed by the connection in the road network, where z_{ij} are elements of the matrix $Z_{N\times N}$. If $z_{ij}=1$, the link from v_i to v_j is feasible. The fleet starts from the depot v_0. They need to complete the delivery task of n customers $c_1,c_2,\dots,c_n$ selected from V and then return to depot v_0. Each customer needs to be served once by one vehicle. The length d_{ij} of the path from the node v_i to v_j is fixed, but the travel time $T_{ij}(t)$ of the path between the node v_i and v_j changes with time t. At any time, the load Q_k of the vehicle k cannot exceed its maximum capacity C. The time window for each customer c_i to be served is $[b_i,e_i]$, and if the arriving time $a_i(t)$ of the vehicle arriving at the customer c_i exceeds e_i, the vehicle is allowed to unload, but there will be a delayed penalty to describe customer dissatisfaction. The waiting time of the vehicle will be accumulated if the arriving time $a_i(t)$ of the vehicle arriving at the customer c_i is earlier than b_i. The time for the vehicle to serve the customer c_i is s_i. d_i and p_i denote the delivery demand and pick-up demand of the customer c_i, respectively. According to the above description, the multi-objective optimization problem can be defined as

$$f_1=\sum_{k=1}^{K}\sum_{i=0}^{n}\sum_{j=0,\,j\neq i}^{n} d_{ij}x_{ij}^{k}z_{ij} \quad , \tag{1}$$

$$f_2(t)=\sum_{k=1}^{K}\sum_{i=1}^{n}\max\left\{a_i(t)\cdot m_{ik}-e_i,0\right\} \quad , \tag{2}$$

$$f_3(t)=\sum_{k=1}^{K}\sum_{i=1}^{n}\max\left\{b_i-a_i(t)\cdot m_{ik},0\right\} \quad , \tag{3}$$

where K represents the number of vehicles used. x_{ij}^{k} denotes the number of times vehicle k accesses v_j from the v_i. If the customer c_i has been served by vehicle k, then $m_{ik}=1$, otherwise $m_{ik}=0$. f_1 represents the total distance traveled by all vehicles.

f_2 is the sum of the delay time of all vehicles. f_3 is the sum of the waiting time of all vehicles, which is used to measure the delivery efficiency. All objectives need to be minimized.

The constraints for MOVRPRTC can be defined as

$$g_1 = \sum_{k=1}^{K} m_{ik} = 1, i = 1,2,3,\ldots,n \ , \tag{4}$$

$$\begin{aligned} g_2 = \sum_{i=1}^{n}\left(d_i \cdot m_{ik}\right) - \sum_{i=1}^{n}\left(d_i \cdot m_{ik}\right) + \\ \sum_{i=1}^{n}\left(p_i \cdot m_{ik}\right) \le C, k = 1,2,3,\ldots,K \end{aligned} \ , \tag{5}$$

$$g_3 = a_0(t) \le e_0 \ , \tag{6}$$

where g_1 means that each customer is served once and g_2 means that the vehicle cannot be overload at any time. g_3 means that the time when the vehicle returns to the depot cannot exceed the end of the depot opening time.

Algorithm 1 Framework of ALSDCMOEA

1: Initialize a parent population and set the global generation counter $t = 0$.
2: Initialize the constraint boundary $\vec{\varepsilon} = \vec{\varepsilon}^{(0)}$, and set the problem state $s = 0$ and the local generation counter $t_s = 0$.
3: **if** the population is $\vec{\varepsilon}$feasible **then**
4: Reduce $\vec{\varepsilon} = \vec{\varepsilon}^{(s+1)}$.
5: Update population $\vec{\varepsilon}$-feasibility and other components.
6: **else**
7: $t_s = t_s + 1$.
8: **end if**
9: Generate the offspring population uses ALS().
10: Select the next parent population.
11: $t = t + 1$.
12: **if** s reaches the final state S or t reaches $AbortingT$ **then**
13: Go to 17.
14: **else**
15: Go back to 3.
16: **end if**
17: Output results.

3. OPTIMIZATION ALGORITHM FOR MOVRPRTC

To address MOVRPRTC, we use the dynamic constraint multi-objective genetic algorithm framework to deal with the constraints, NSGA-II [12] for multi-objective selection, and our newly proposed adaptive local search algorithm to optimize candidate solutions.

3.1 Multi-objective Optimization

Multi-objective optimization is the process of simultaneously optimize two or more conflicting objectives. A multi-objective problem can be written as follows:

$$\min F(x) = \left\{ f_1(x), \ldots, f_m(x) \right\} \tag{7}$$

subject to $x \in \Omega$, where Ω is the decision variable space. $F : \Omega \to R^m$ consists of m objective functions.

Given two feasible solutions $x, y \in \Omega$, we say that x dominates y, if $\forall i : f_i(x) \le f_i(y)$ and $\exists j : f_j(x) < f_j(y)$, $i, j \in 1,\ldots,m$. x^* is said to be Pareto Optimal if it is not dominated by any other feasible solution. The goal for multi-objective optimization is to find a set of Pareto Optimal solutions, namely Pareto Set. The set contains a number of non-dominated points in objective space are called Pareto front.

3.2 Framework of ALSDCMOEA

The dynamic constraint multi-objective genetic algorithm framework has three characteristics for MOVRPRTC: 1) a constraint optimization problem equivalent converted into a dynamic constraint multi-objective optimization problem with two types of objectives: a) original objective; b) constraint-violation objective. 2) using the idea of simulated annealing, gradually reducing the constraint boundary aims to handle the constraint difficulty.

Inspired by DCMOEA, f_2 and f_3 in MOVRPRTC can be converted into constraints. Then the problem can be converted into VRPRTC:

$$\min f = \sum_{k=1}^{K}\sum_{i=0}^{n}\sum_{j=0, j\ne i}^{n} d_{ij} x_{ij}^{k} z_{i,j} \tag{8}$$

$$\begin{aligned} & st: \\ & g_1(t) = \sum_{k=1}^{K}\sum_{i=1}^{n} \max\left\{a_i(t)\cdot m_{ik} - e_i, 0\right\} = 0 \\ & g_2(t) = \sum_{k=1}^{K}\sum_{i=1}^{n} \max\left\{b_i - a_i(t)\cdot m_{ik}, 0\right\} = 0 \\ & g_3 = \sum_{i=1}^{n}\left(d_i \cdot m_{ik}\right) - \sum_{i=1}^{n}\left(d_i \cdot m_{ik}\right) + \\ & \qquad \sum_{i=1}^{n}\left(p_i \cdot m_{ik}\right) - C \le 0, k = 1,2,3,\ldots,K \\ & g_4(t) = a_0(t) - e_0 \le 0 \end{aligned} \tag{9}$$

Then, according to the method of DCMOEA which transforms a constrained optimization problem (COP) to a dynamic constrained multi-objective problem (DCMOP), VRPRTC can be converted to DCMOVRPTRC:

$$DCMOVRPRTC^{(s)} \begin{cases} \min\left(f(x), \mathrm{cv}(x)\right) \\ \mathrm{st} : \vec{g}(x) \le \vec{\varepsilon}^{(s)} \end{cases} \tag{10}$$

$$\mathrm{cv}(x) = \frac{1}{4}\sum_{i=1}^{4} \frac{G_i(x)}{\max_{x\in \mathbf{P}(0)}\left\{G_i(x)\right\}} \tag{11}$$

$$G_i(x) = \max\left\{g_i(x), 0\right\}, i = 1,2,\ldots,4 \tag{12}$$

where x in formula (10) denotes a solution of DCMOVRPRTC, which consists of a set of routes. S is a given number of environmental changes, $s = 0,1,\ldots,S$. $\vec{\varepsilon}^{(s)}$ is the dynamic constraint boundary, and s is the environmental state. An environmental change denotes a reduction of the constraint boundary from state s to state $s+1$.

After converting MOVRPTC to DCMOVRPRTC: 1) the original three-objective optimization problem is converted into a single objective optimization problem; 2) the optimization of f_2 and f_3 in MOVRPRTC can be realized by the constraints handling mechanisms in DCMOEA.

The framework of ALSDCMOEA for DCMOVRPRTC is shown in Algorithm 1. The multi-objective optimization algorithms in line 10 is NSGA-II in this paper, and can be replaced with other MOEAs such as MOEA/D [13] and HypE [14].

3.3 Solution Initialization

Before introducing the initialization method, we need to explain the encoding of the solution first. Since MOVRPRTC contains the road network, a solution should include not only the serving sequence of customers but also the detailed traveling sequence of coordinate points of each vehicle. Figure 3 shows an example, where sequence A indicates the serving sequence of the customers, sequence B indicates the traveling sequence of the coordinate points, gray dots between the depot and the customer 1 indicates the point in the road network that the vehicle needs to travel from the depot to the customer 1. The optimization algorithm only operates sequence A to reallocate customers for vehicles and change their serving sequences. Sequence B is only used for evaluation.

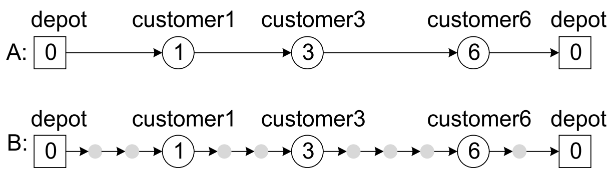

Figure 3. Solution representation.

To get the initialization solution, we use the agglomerative hierarchical clustering (AHC) algorithm to group customers into several clusters, one of which represents the customers served by one vehicle. The advantage of using AHC is that the number of vehicles required can be adaptively adjusted according to the demand of customers.

3.4 Adaptive Local Search for MOVRPRTC

The fundamental tasks of solving MOVRPRTC are the allocation of vehicles for customers and the sequencing of customer servings. In order to address these two tasks, we design eight local search operators and use an adaptive framework to help the algorithm jump out of the local optima. The eight operators showed as below:

LS1: Randomly choose two different customers from a random vehicle, and swap their serving orders;

LS2: Randomly choose two different customers from a random vehicle, and reverse the serving sequence between these two customers (the length of the sequence shall not exceed 3);

LS3: Randomly choose one vehicle and randomly intercept a sequence from it, then insert the sequence into another randomly selected vehicle. Then the entire serving sequence of the inserted vehicle should be re-sorted by nearest neighbor search;

LS4: Find two customers with the maximum wait time and with the maximum delay time, respectively. Insert the customer with the maximum delay time at the position right before the customer with the maximum wait time;

LS5: Randomly select a customer from a random vehicle, and insert it at a random location in another randomly selected vehicle;

LS6: Repeat LS5 several times (no more than 4 times);

LS7: Find the customer with the maximum traveling time, and insert it at the position right before the customer who has the maximum waiting time;

LS8: Find the customer with the maximum traveling time and insert it at a random location.

LS1 and LS2 are designed for changing the serving sequence, and the other operators are designed for changing the allocation of customers. After the local search is operated on a solution, the solution's path will be reconstructed by a specific function. The main content of this function is the A* algorithm, which can get the shortest route for every vehicle according to the real-time traffic conditions.

Algorithm 2 Adaptive Local Search

Input: solution x.
$success\_flag = false$.
for $i = 0 \rightarrow popSize$ **do**
  $x^t = LS\_w(x)$ //$w = 1, ...8$, roulette wheel selection from $LS1$ to $LS8$.
  if x^t dominating x **then**
    $score[w] + +$. //score for operators
    $x = x^t$.
    $success\_flag = true$.
  else
    $x^t = x$.
  end if
end for
if $cnt \geq 100$ **then** //cnt represent the times of x not change
  $score[8] = (1, \ldots, 1)$.
  if $success\_flag == true$ **then**
    $success\_flag = false$.
    $cnt = 0$.
  end if
end if

The pseudo-code of ALS is shown in Algorithm 2. The subtlety part is the adaptive mechanism. At the beginning of the algorithm, each operator has the same weight, and then it will be adjusted according to the performance of the offspring generated by the operator in each iteration. If a better solution is generated, the corresponding operator's score will be incremented by 1. In each iteration, the operators will be selected by the roulette. If the best solution remains unchanged for specific iterations, the algorithm

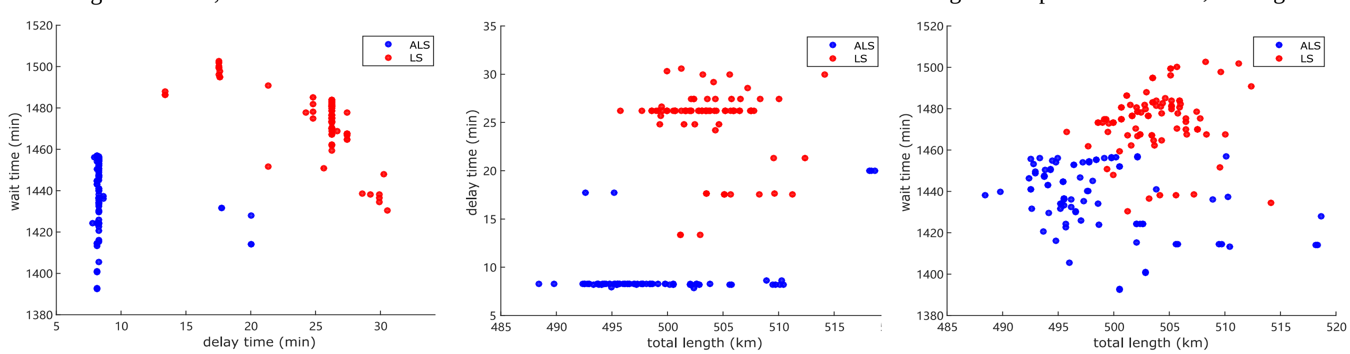

Figure 4. Dominance of final solutions generated by ALS and LS.

will be considered as being converged. At this time, the scores of all operators are reset to 1, and then the algorithm continues until the termination condition is reached.

4. EXPERIMENT

The following experiments are run on a PC equipped with Core-i7 3.4 GHz and 16 GB of RAM. All algorithms are implemented in C++. Problem parameter settings: The opening time of the depot is from 8:00 to 24:00. The vehicle type is the same one, with a maximum capacity of 3 tons. The number of customers is 100. Algorithm parameter setting: The population size is 100, which means that the number of new solutions generated in each iteration is 100. The maximum number of iterations is 10000.

4.1 Performance of Optimization Methods

In order to prove the effectiveness of the adaptive scheme for local search operators, we compare our original algorithm with a modified version without the adaptive scheme. Their operators are the same, except that the modified version randomly selects one of eight operators in each iteration. As shown in Figure 4, the original version is superior to the modified version in terms of the wait time and delay time, and meanwhile not inferior to the modified version in terms of the total length.

Aiming to illustrate the full effect of ALSDCMOEA more intuitively, we compare the best solution of the initial solutions with the best one of the final solutions. The best solution is defined as the solution with the minimum sum of three objectives. Their objectives are shown in Table 1.

Table 1. The objectives of the initial solution and final solution

	total length	wait time	delay time
Initial solution	524.819 km	2270.4 min	714.704 min
Final solution	**500.507 km**	**1392.9 min**	**8.17108 min**
Reduce ratio	4.63%	38.65%	98.86%

5. CONCLUSION

This paper proposed a multi-objective vehicle routing problem with real-time traffic conditions. The traffic condition data and customer demand data in the model all come from real-world data. In order to address this problem, this paper also proposed an adaptive local search algorithm based on a dynamic constrained multi-Objective evolutionary algorithm framework. Experimental results show that ALSDCMOEA can effectively solve MOVRPRTC.

The scale of MOVRPRTC used in this paper is small, and we can increase it in many ways, including extending the coverage of the map, increase the number of customers, increase the density of the road network, etc.. As for the algorithm proposed in this paper, we need to conduct more experiments to prove the effectiveness of the DCMOEA. In addition, to further improve the performance of ALSDCMOEA, some other multi-objective optimization algorithms should also be attempted to use, such as MOEA/D.

6. ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61673355, in part by the Fundamental Research Funds for the Central Universities, China University of Geosciences (Wuhan) under Grant (CUG170603, CUGGC02), in part by the European Union’s Horizon 2020 research and innovation programme under Grant No. 739551 (KIOS CoE) and from the Government of the Republic of Cyprus through the Directorate General for European Programmes, Coordination and Development.

7. REFERENCES

[1] S Lin, Z Lee, K Ying, C Lee. Applying Hybrid Meta-Heuristics for Capacitated Vehicle Routing Problem [J]. *Expert Systems with Applications*, 2009, 36 (21):1505-1512.

[2] S Iqbal, M Kaykobad, M S Rahman. Solving the Multi-Objective Vehicle Routing Problem with Soft Time Windows with the Help of Bees [J]. *Swarm and Evolutionary Computation*, 2015, 24:50-64.

[3] M Mavrovouniotis, S Yang. Ant Colony Optimization with Memory-Based Immigrants for the Dynamic Vehicle Routing Problem [M]. *IEEE Congress on Evolutionary Computation*, Brisbane, Australia, 2012: 1-8.

[4] Y Zhou, J Wang. A Local Search-Based Multiobjective Optimization Algorithm for Multiobjective Vehicle Routing Problem with Time Windows [J]. *IEEE Systems Journal*, 2015, 9 (3):1100-1113.

[5] D M Pierre, N Zakaria. Stochastic Partially Optimized Cyclic Shift Crossover for Multi-Objective Genetic Algorithms for the Vehicle Routing Problem with Time-Windows [J]. *Applied Soft Computing*, 2017, 52:863-876.

[6] P Li, J He, D Zheng, Y Huang, C Fan. Vehicle Routing Problem with Soft Time Windows Based on Improved Genetic Algorithm for Fruits and Vegetables Distribution [J]. *Discrete Dynamics in Nature and Society*, 2015: 1-8.

[7] H Zhang, Q Zhang, L Ma, Z Zhang, Y Liu. A Hybrid Ant Colony Optimization Algorithm for a Multi-Objective Vehicle Routing Problem with Flexible Time Windows [J]. *Information Sciences*, 2019, 490:166-190.

[8] R Liu, Y Tao, X Xie. An Adaptive Large Neighborhood Search Heuristic for the Vehicle Routing Problem with Time Windows and Synchronized Visits [J]. *Computers & Operations Research*, 2019, 101:250-262.

[9] S Chen, R Chen, G Wang, J Gao, A K Sangaiah. An Adaptive Large Neighborhood Search Heuristic for Dynamic Vehicle Routing Problems [J]. *Computers & Electrical Engineering*, 2018, 67:596-607.

[10] S Zeng, R Jiao, C Li, X Li, J S Alkasassbeh. A General Framework of Dynamic Constrained Multiobjective Evolutionary Algorithms for Constrained Optimization [J]. *IEEE Transactions on Cybernetics*, 2017, 47 (9):2678-2688.

[11] Amap API for Traffic Status. https://lbs.amap.com/api/webservice/guide/api/trafficstatus.

[12] K Deb, A Pratap, S Agarwal, T Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II [J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182-197.

[13] Q Zhang, H Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition [J]. *IEEE Transactions on Evolutionary Computation*, 2007, 11(6):712-731.

[14] J Bader, E Zitzler. HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization [J]. *Evolutionary Computation*, 2011, 19 (1):45-76.