An Adapive Evolutionary Algorithm for Bi-Level Multi-objective VRPs with Real-Time Traffic Conditions

Baojian Chen
School of Automation
China Univsersity of Geosciences
Wuhan 430074, China
2545415252@qq.com

Changhe Li
School of Automation
China University of Geosciences
Wuhan 430074, China
changhe.lw@gmail.com

Sanyou Zeng
School of Mechanical Engineering and Electronic Information
China University of Geosciences
Wuhan 430074, China
sanyouzeng@gmail.com

Shengxiang Yang
School of Computer Science and Informatics
De Montfort University
Leicester LE1 9BH, United Kingdom
syang@dmu.ac.uk

Michalis Mavrovouniotis
KIOS Research and Innovation Center of Excellence and Department of Electrical and Computer Engineering
University of Cyprus
Nicosia 2109, Cyprus
mavrovouniotis.michalis@ucy.ac.cy

***Abstract*—The research of vehicle routing problem (VRP) is significant for people traveling and logistics distribution. Recently, in order to alleviate global warming, the VRP based on electric vehicles has attracted much attention from researchers. In this paper, a bi-level routing problem model based on electric vehicles is presented, which can simulate the actual logistics distribution process. The classic backpropagation neural network is used to predict the road conditions for applying the method in real life. We also propose a local search algorithm based on a dynamic constrained multi-objective optimization framework. In this algorithm, 26 local search operators are designed and selected adaptively to optimize initial solutions. We also make a comparison between our algorithm and 3 modified algorithms. Experimental results indicate that our algorithm can attain an excellent solution that can satisfy the constraints of the VRP with real-time traffic conditions and be more competitive than the other 3 modified algorithms.**

Keywords—bi-level routing problem, multi-objective optimization, constrained optimization, local search

I. Introduction

In recent years, the vigorous development of ecommerce industry has promoted the development of logistics industry. In 2019,Chinese total social logistics cost accounts for 14.7% of gross domestic product while the proportion remains at 8% to 9% in developed countries. vehicle routing problem (VRP) aims to reduce the cost of logistics and improve the satisfactions of customers. The key to the former goal is to reduce the number of used vehicles, select the appropriate type of vehicles, reduce the energy consumption and the driving distance and so on; the key to the later goal is that the vehicle serves the customer within the customer's time window.

The VRP is the general term of a class of problems. The purpose of studying this kind of problem is to determine the appropriate routes of vehicles based on one or more stations for a number of many geographically dispersed customers, and the VRP was first proposed by Dantzig and Ramser [1]. Based on the problem model, a variety of versions of VRP were derived, such as VRP with capacity constraints (CVRP) [2, 3], VRP with picking up and delivery goods (VRPPD), VRP with time windows (VRPTW) [4] and so on. In the process of actual logistics distribution, the traffic conditions directly affect the driving time of vehicles on the road network, which means that the dynamic change of traffic conditions is one of the main challenges for VRPs. Existing studies considered traffic factors simply, they used a periodic or aperiodic random function to generate traffic conditions [5], which has a shortcoming that the traffic flow is independent among road sections. In order to address this shortcoming, a single-level VRP mode based on real traffic conditions is proposed in [6], but the mode is simple and unrealistic. At the same time, the customer's satisfactions will be reduced substantially while the time that a vehicle gets to the customer is later than the customer's latest specified time. Single-level VRP that has been widely studied is far away from our life. Therefore, in order to make the VRP model applicable to real world, we propose a new bi-level VRP that combines the real road network topology, the multi-objective with a soft time windows, and delivery and picking up goods (BLMOVRPRTC).

There are two kinds of algorithms for VRPs in general. They are accurate algorithm and heuristic algorithm, respectively. The typical representative of accurate algorithm is saving routing algorithm [7], but it takes a long time to find global optimal solution. In terms of time consumption, heuristic algorithm is the unique feasible way to solve practical and complex VRPs in acceptable time; heuristic algorithms mainly include the simulated annealing algorithm [8], genetic algorithm [9], particle swarm optimization algorithm [10, 11], ant colony optimization algorithm [12]. Through the research and analysis of these algorithms that have been applied to solve VRPs, it's not difficult to find that they abandon infeasible solutions during iterations, which means that those algorithms only search in the feasible region. Therefore, this will cause that high-quality solutions outside the feasible region can't be found. In order to overcome this disadvantage, we use a dynamic constrained multi-objective evolutionary algorithm framework (DCMOEA) [13] to handle

The work was supported in part by the National Natural Science Foundation of China under Grant No.62076226, in part by the Fundamental Research funds for the Central Universities, China University of Geoscience (Wuhan) under Grant CUGGC02, in part by the Hubei Provincial Natural science Foundation of China under Grant 2015CFA010, in part by the 111 Project under Grant B17040, in part by the European Union's Horizon 2020 research and innovation programme under grant agreement No.739551 (KIOS CoE) and from the Government of the Republic of Cyprus through the Directorate General for European Programmes, Coordination and Development.

constraints. It turns out that this method greatly decreases the pressure of selection solutions in multi-objective optimization.

The main contributions are as follows in this paper. Firstly, we propose a new bi-level VRP model with real road network and real-time traffic conditions. Secondly, the DCMOEA is used to solve constraints in combinatorial optimization problem. Thirdly, we design 3 neighborhoods that include 26 productive local search operators and an adaptive mechanism that can select operators to optimize BLMOVRPRTC, and we also compare our algorithm with 3 modified algorithms. Experimental results show that those operators are effective in solving the proposed BLMOVRPRTC.

The rest of this paper is organized as follows: Section II presents the modeling method of BLMOVRPRTC, including the construction road network topology, the construction of the model of actual traffic conditions and the formulation of the VRP. Section III introduces the optimization algorithm for BLMOVRPRTC. Experimental results are shown in Section IV. Finally, the conclusion is given in Section V.

II. MODELING OF BLMOVRPRTC

A. Construction of Real Road Network

The research object is the bi-level VRP based on the actual traffic conditions for electric vehicles. The route of vehicles is an abstraction of the road in the actual traffic. The core task of road construction is to determine the connection relationship among adjacent road nodes, especially the connection relationship among nodes at the crossroad. Fig. 1 illustrates the connection relationship at the crossroad.

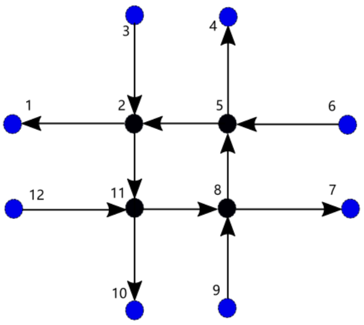


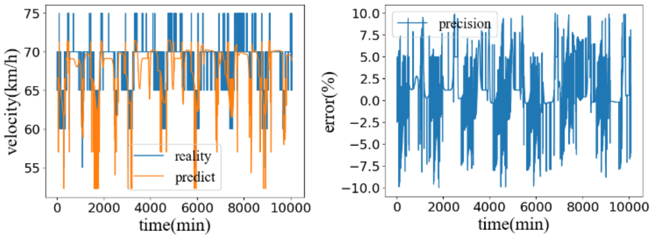
Fig. 1. Connection relationship of road nodes

In Fig. 1, points 2, 5, 8, 11 are used to describe a four-way intersection of two crossing roads. Point 2 has two precursor points which are point 3 and point 5. At the same time, point 2 has two successor points which are point 1 and point 11. The road network topology is completed after defining the connection relationship of road nodes. Finally, the data structure called double linked list is used to store the road network topology. The traffic data is displayed in link: traffic data.

B. Modeling Construction of Traffic Conditions

The velocity of each section in the road network changes with time. In a specific area, the road network structure will not change in a short time and people's daily life is regular. We can use specific traffic data of certain days to predict future traffic conditions. At the same time, considering the difference of traffic conditions between working days and non-working days, it is reasonable to use the previous several weeks' traffic conditions to predict the traffic conditions of the next week. So, we collected 14 weeks' traffic conditions of trunk road in Wuhan and took the traffic conditions of the first 13 weeks as training data, and the remaining traffic conditions as test data. Finally, we adopt the classic backpropagation neural network to predict the traffic trend of the next week.

By calculating the error between the predicted data and the test data, the error is within the range of -10% to 10%. The predicted traffic trend is consistent with the actual traffic trend essentially. Due to the space limitation, the prediction traffic trend and the error of only one road section are presented in Fig. 2.



(a) (b)

Fig. 2. The comparison and relative deviation of true and predict velocity

C. Modeling Construction of BLMOVRPRTC

BLMOVRPRTC is described as follows: in a road network topology $G=(V,E)$, $V=\{v_1,v_2,...,v_N\}$ represents N nodes in the road network. $E=\{<i,j>|e_{ij}=1\}$ expresses the connection status of any two nodes, where e_{ij} are the elements of the matrix $E_{N\times N}$. If it is feasible from v_i to v_j directly, then $e_{ij}=1$, otherwise, $e_{ij}=0$; v_1 represents the depot node in road network. There are two types of electric vehicles in this problem, they are truck and van, respectively. Trucks start from the depot node v_1. Then carry out the service of picking up and sending goods at the distribution centers $dc_1,dc_2,...,dc_m$ selected from V. Finally return to the depot node v_1; meanwhile vans start from the distribution centers dc_i, $i=1,2,...,m$. Then carry out the service of picking up and sending goods at the customers $c_1,c_2,...,c_n$ selected from V. Finally return to distribution center dc_i.

It assumes that each customer can only be served by one van and completed the service once. The distance d_{ij} from v_i to v_j is fixed in road network but the driving time T_{ij} from v_i to v_j is changing with time. The loading capacity of truck and van are L_b and L_s, respectively. L_b and L_s shall not exceed the maximum capacity which are C_b, C_s, respectively at any time. The time window of each customer c_i is $[TW_{li},TW_{ui}]$. If the time $a_i(t)$ that the van gets to the customer c_i exceeds TL_{ui}, the goods are allowed to be unloaded, but the waiting time of the customer will be longer, which causes that customer's satisfactions will be reduced. On the contrary, if the time $a_i(t)$ that the van gets to the customer c_i is earlier than TW_{li}, the vehicle must wait, and the customer can only be served after the time window of the customer c_i is opened. The service time of van for customer c_i is s_i. DW_i and PW_i represent the delivery and picking up weight of the van in the customer c_i. The maximum driving distance of truck and van are $D_{b\max}$ and $D_{s\max}$ at full electricity capacity situation, respectively. $char_b$ and $char_s$ present the charging speed of truck and van, respectively. The charging speed indicates the distance that one vehicle can travel after charging for one minute. We adopt the method of linear summation to balance the driving distance of all trucks and all vans, and coe_s, $coe_s \in [0,1]$ represents a weight coefficient. According to the above description, the multi-objective optimization problem is described as follows:

$$f_1 = coe_b \times \sum_{k_b=1}^{K_b} \sum_{i=0}^{dp_m} \sum_{j=0, j\neq i}^{dp_m} d_{ij} x_{ij}^{k_b} e_{ij} + (1-coe_b) \times \sum_{m=0}^{dp_m} \sum_{k_s=1}^{K_s^m} \sum_{i=0}^{n} \sum_{j=0, j\neq i}^{n} d_{ij} x_{ij}^{k_s} e_{ij} \quad (1)$$

$$f_2(t) = \sum_{m=0}^{dp_m} \sum_{k_s=1}^{K_s} \sum_{i=1}^{n} \max\{a_i(t) \cdot h_{ik_s}^m - TW_{ui}, 0\} \quad (2)$$

$$f_3(t) = \sum_{m=0}^{dp_m} \sum_{k_s=1}^{K_s} \sum_{i=1}^{n} \max\{TW_{li} - a_i(t) \cdot h_{ik_s}^m, 0\} \quad (3)$$

where dp_m represents the number of delivery centers served by trucks, K_b denotes the number of trucks used, K_s^m indicates the number of vans used in the delivery center m. x_{ij}^k is the times of the van k passes from v_i to v_j, if the customer c_i is served by the van k in the delivery center m, then $h_{ik_s}^m = 1$, else $h_{ik_s}^m = 0$. f_1 shows that the total driving distance of all trucks and vans used. $f_3(t)$, $f_2(t)$ states the sum of the waiting time and delay time of all vans used to serve all customers. All objectives need to be minimized.

The constraints except for three objectives in the BLMOVRPRTC are shown as follows:

$$g_1 = \sum_{m=0}^{dp_m} \sum_{k_s=1}^{K_s} h_{ik_s}^m = 1, i = 1, 2, ..., n \quad (4)$$

$$g_2(t) = \sum_{i=1}^{n} \left(DW_i \cdot h_{ik_s}^m\right) - \sum_{i=1}^{n} DW_i \cdot h_{ik_s}^m \cdot b_i(t) + \sum_{i=1}^{n} PW_i \cdot h_{ik_s}^m \cdot b_i(t) \leq C_s \quad (5)$$

$$g_3(t) = \sum_{m=1}^{dp_m} D_m \cdot h_{mk} - \sum_{m=1}^{dp_m} D_m \cdot h_{mk} \cdot b_m(t) + \sum_{m=1}^{dp_m} P_m \cdot h_{mk} \cdot b_m(t) \leq C_b \quad (6)$$

$$g_4(t) = \sum_{k_b=1}^{K_b} \max\{RT_{k_b} - ET_d, 0\} = 0 \quad (7)$$

$$g_5(t) = \sum_{m=1}^{dp_m} \sum_{k_s=1}^{K_s} \max\{RT_{k_s}^m - ET_{k_s}^m, 0\} = 0 \quad (8)$$

where g_1 indicates that each customer c_i can only be served once by one van, $g_2(t)$ and $g_3(t)$ show that trucks and vans shall not be overloaded at any time. $g_4(t)$ denotes the time that all trucks return to depot after completing task is higher than the closing time of the depot. $g_5(t)$ reveals that the time that all vans return to delivery center d $(d = 1, 2, ..., dp_m)$ after completing task is higher than the closing time of the delivery center d.

III. Local Search Algorithm for BLMOVRPRTC

A. Multi-objective Optimization

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

$$\min F(x) = \{f_1(x), ..., f_n(x)\} \quad (9)$$

subject to $x \in \Omega$, Ω is the decision variable space, $F(x)$ consists of n objective functions.

$x_1, x_2 \in \Omega$ are feasible solutions. if $\forall i: f_i(x_1) \leq f_i(x_2)$ and $\exists j: f_j(x_1) < f_j(x_2), i,j \in 1, 2, ..., n$ are met, then x_1 dominates x_2. x^* is the Pareto Optimal if it is not dominated by other solutions. The purpose of multi-objective optimization is to obtain a set of Pareto Optimal, which are called Pareto Set.

B. Framework of DCMOEA

In order to address the difficulties that multi-objectives and multi-constraints bring, we adopt the DCMOEA. The algorithm framework has two advantages. firstly, it can transform a constrained optimization problem into a dynamic constrained multi-objective problem, in which objectives are the original objective and constraint-violation objective. Secondly, inspired by the idea of simulated annealing, it adopts the ε constraint handling method to deal with the constraints. This method expands the search range in the process of evolution, and gradually shrink the constraint boundary to the feasible region. So our problem is transferred from BLMOVRPRTC with 3 objectives to BLDCMOVRPRTC with 2 original objectives that are f_1 and $f_2(t)$, and $f_3(t)$ in original objectives is converted to a constraint. According to the method of DCMOEA which transforms a constrained optimization problem to a dynamic constrained multi-objective problem, BLDCMOVRPRTC with 2 objectives can be converted to BLDCMOVRPRTC with 3 objectives as follows:

$$BLDCMOVRPRTC^s \begin{cases} \min(f_1(x), f_2(x), cv(x)) \\ st: \vec{g}(x) \leq \vec{\varepsilon}^{(s)} \end{cases} \quad (10)$$

$$cv(x) = \frac{1}{5} \sum_{i=1}^{5} \frac{G_i(x)}{\max_{x \in P(0)}\{G_i(x)\}} \quad (11)$$

$$G_i(x) = \max\{g_i(x), 0\}, i = 1, 2, ..., 5 \quad (12)$$

where x is a solution of BLDCMOVRPRTC, this solution consists of a set of driving routes of trucks and vans. S denotes the times of environmental changes, $s = 1, 2, ..., S$, s is the environmental status, and $\vec{\varepsilon}^{(s)}$ is the dynamic constraint boundary. If environment changes one time, then the reduction of the constraint boundary from state s to $s+1$.

There are two changes mainly by this transformation: firstly, the optimization problem with 3 original objectives is converted into the problem with 2 original objectives and one constraint-violation objective; secondly, $f_3(t)$ in

BLMOVRPRTC can be treated as a constraint handled by the ε constraint handling method in DCMOEA.

C. Encoding of Solutions

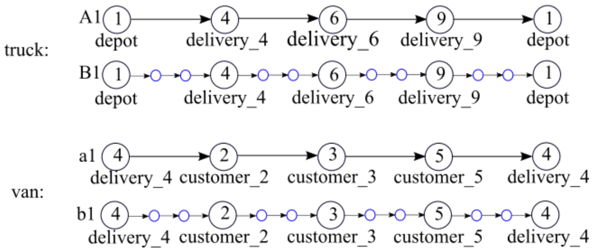


Fig. 3. The encoding of solutions

Because the road network used in this paper is constructed by a series of actual geographic coordinate points, the road network contains different node types, which includes depot node, customer nodes, charging nodes, delivery center nodes and general nodes. So a solution should include not only the serving sequence of delivery centers or customers but also the detailed traveling sequence of coordinate points of all trucks and all vans.

As shown in Fig. 3, where sequences $A1$ and $a1$ denote the serving sequence of the delivery centers and customers, respectively. Sequences $B1$ and $b1$ indicate the traveling sequence of the coordinate points passed by trucks and vans, respectively. The optimization algorithm only operates sequences $A1$ and $a1$ to reallocate delivery centers for trucks and reallocate customers for vans. Sequence $B1$ and $b1$ only are used to evaluate.

D. Initialization of population

The quality of the initial solutions directly affects the performance of the algorithm for local optimization algorithm. High-quality initial solutions can make the algorithm find a better solution in a short time.

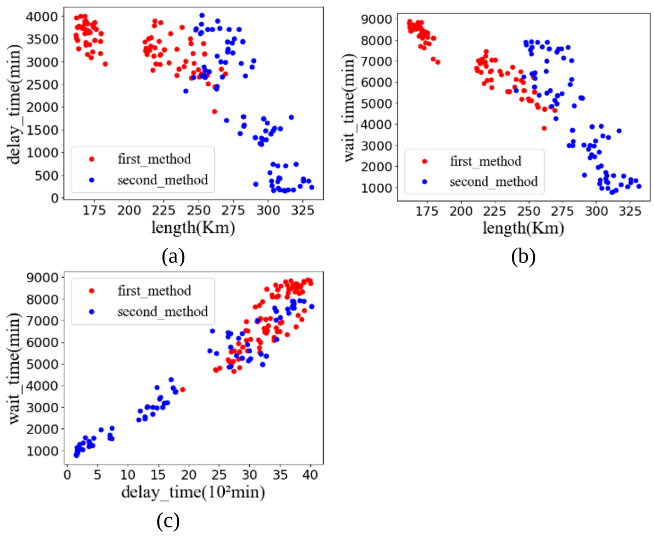


Fig. 4. The initial solutions comparison of different method

In order to obtain high-quality initial solutions, we cluster all customers and delivery centers in three steps in this paper according to the workflow of logistics distribution. The first step is to assign customers to the delivery centers, where we take two methods to accomplish the task. The main difference of them is that the first uses the heuristic information which includes the distance from customers to delivery centers and the time windows of customers while the second adopts the heuristic information that includes the start time of vans and the distance from customers to delivery centers. In the second step, we make full use of the agglomerative hierarchical clustering (AHC) algorithm to cluster all delivery centers into several clusters, one of which means that the delivery centers are served by one truck. In the third step, we use AHC algorithm to cluster customers at same delivery center to several clusters, one of which represents those customers are served by one van. The outstanding advantage of AHC is that the number of trucks and vans required can be adaptively adjusted according to the demand of customers.

It can be seen from Fig. 4 that the first method tends to have shorter driving distance and higher delay time and waiting time. On the contrary, the second method tends to have higher driving distance and shorter delay time and waiting time. In order to increase population diversity, we randomly choose one method of them to construct one initial solution. The result of final clustering can be found in Fig. 5.

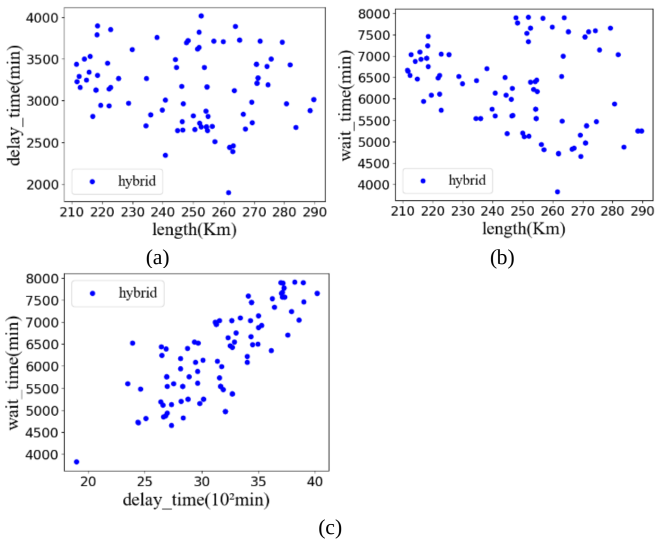


Fig. 5. The population produced by using hybrid method

E. Designing Local Search Operators

As we all know, the core task of BLMOVRPRTC is the allocation of customers and delivery centers for vehicles. 26 local search operators are designed to address the task and 3 neighborhoods are designed according to the range of action of operators. Operators are designed based on three types of neighborhoods. LS_1～LS_7 are designed for changing the serving sequence in one truck or one van. LS_8～LS_11 are designed for changing the allocation of customers at same delivery center. The others are designed for changing the allocation of customers or delivery centers in the whole road network. The purpose of designing neighborhood is to speed up the convergence of the algorithm. After one of those operators is operated on a solution, the solution's route will be reconstructed by using the famous A^* algorithm. According to the characteristics of those operators. The 26 operators are shown as follows.

LS_1: choose two different random delivery centers from a random truck and swap their serving orders.

LS_2: choose two different random customers from a random van and swap their serving orders.

LS_3: put the customer with the largest delay time in front of the customer with the largest waiting time in the same van.

LS_4: choose 2-opt operator to adjust the serving sequence of the truck that has the longest driving distance.

LS_5: choose 2-opt operator to adjust the serving sequence of the van that has the longest driving distance.

LS_6: choose 3-opt operator to adjust the serving sequence of the truck that has the longest driving distance.

LS_7: choose 3-opt operator to adjust the serving sequence of the van that has the longest driving distance.

LS_8: choose one van randomly and intercept a sequence from it randomly, then insert the sequence into another randomly selected van at the same delivery center. Finally, the serving sequence of the inserted van is sorted by nearest neighbor search algorithm.

LS_9: choose a random customer from a random van and choose another customer in another van randomly at the same delivery center, then swap them.

LS_10: repeat LS_9 many times (no more than 4 times)

LS_11: put the customer with the largest delay time in front of the customer with the largest waiting time at the same delivery center.

LS_12: choose one van randomly and intercept a sequence from it randomly then insert the sequence into another randomly selected van in the whole road network, finally, the serving sequence of the inserted van is sorted by nearest neighbor search algorithm.

LS_13: choose one random delivery center from a random truck and choose another one in another truck randomly, then swap them.

LS_14: repeat LS_13 many times (no more than 4 times).

LS_15: choose the delivery center with the largest traveling time from the previous delivery center or depot node, and put it into a truck selected randomly.

LS_16: choose the customer with the largest waiting time, and put it into the van that has the least time difference between the allowing starting serving time of the customer and the starting serving time of vans.

LS_17: choose the customer with the largest waiting time, and put it into a van selected randomly.

LS_18: choose the customer with the largest delay time, and put it into a van selected randomly.

LS_19: choose all customers from one random van, and choose all customers from another van at different delivery center, then swap them.

LS_20: choose the customer with the largest delay time, and put it into the van that has the least time difference between the allowing starting serving time of the customer and the starting serving time of van.

LS_21: choose some customers (no more than 3) from the van with the largest charging time, and insert them to the van with the least charging time.

LS_22: choose some customers (no more than 3) from the van with the largest waiting time, and insert them to the van with the least waiting time.

LS_23: choose the customer who has the largest time difference between the starting serving time of the customer and the starting serving time of vans. Then insert it to the small vehicle that has least time difference between the starting serving time of the customer and the starting serving time of van.

LS_24: choose two solutions from population randomly, then select some delivery centers (no more than 3) from a truck in one solution and select some delivery centers (no more than 3) from another truck in another solution, then swap them.

LS_25: choose two solutions from population randomly, then select some customers (no more than 3) from a van in one solution and select some customers (no more than 3) from another van in another solution, then swap them.

LS_26: choose a customer from the van selected randomly and choose another customer from another van, then swap them.

F. Adaptive Mechanism

Algorithm 1 Adaptive Local Search

input: population pop
$success\_flag = false$
for $i = 0 \rightarrow popsize$ **do**
  $x' = Neighborhood\_n(LS\_n_w(x))$, $n = 1, 2, 3$. LS_n_w represents w operators of neighborhood n.
  if x' dominating x **then**
    $neighborhood\_score[n-1]++$
    $score\_n_w[w-1]++$
    $x = x^t$
    $success\_flag = true$
  else
    $x^t = x$
  end if
end for
if $count \geq 100$ **then** //count represents the times of x not changes
  $neighborhood\_score[3] = (1, 1, 1)$
  $score\_n_w[Nn_w] = (1, 1, ..., 1)$, Nn_w represents the number of operators in neighborhood n.
  if $success\_flag = true$ **then**
    $success\_flag = false$
    $count = 0$
  end if
end if

The pseudo-code of the adaptive mechanism is shown in Algorithm 1. The core part is the adaptive method. At the beginning of the algorithm, each neighborhood and each operator have the same weight (score). They 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 neighborhood's score and operator's score will be increased by 1 in each iteration. The neighborhood and the operators in the same neighborhood will be selected by the roulette method. The algorithm will be considered as being converged when the best solution remains unchanged for specific iterations. The scores of neighborhood and operators are set to 1 after the algorithm converges, and the algorithm will continue to compute until the termination conditions is reached.

IV. EXPERIMENT

The following experiments are run on a PC equipped with core-i7 3.4GHz and 32 GB of RAM, all algorithms are implemented in C++, problem parameter settings: the opening time of the depot is from 6:00 to 24:00 and the opening time of all delivery centers are from 8:00 to 23:00. The weight coefficient coe_b is 0.4. The maximum capacity of truck and van are 1200kg and 250kg, respectively. $D_{b\max}$ and $D_{s\max}$ are 40 and 25 kilometers, respectively. $char_b$ and $char_s$ are 500 and 300 meters per minute, respectively. The number of delivery centers and customers are 6 and 100, respectively. Algorithm parameter setting: the size of population is 100, which indicates that the number of new solutions generated in each iteration is 100. The maximum number of evaluation is 100000.

A. Performance of Optimization Methods

In order to show the performance of our algorithm intuitively, we make a comparison among the best solution of the random initial solutions, the best solution of clustering initial solutions and the best solution of final solutions. the best solution is defined as the solution which has the minimum value of the sum of all normalized objective values. It is not difficult to find that the algorithm can greatly reduce the delay time and waiting time of vans.

TABLE I. THE OBJECTIVE OF THE INITIAL AND FINAL SOLUTIONS

Best solution	*Total length (Km)*	*Delay time (min)*	*Wait time (min)*
random initialization	367.665	3866.38	6635.91
clustering initialization	261.628	1897.25	3823.64
final	349.342	26.09	129.88

B. Performance comparison of Algorithm with/without neighborhood mechanism

In order to prove the effectiveness of the neighborhood mechanism in this paper, we make a comparison between original algorithm and a modified version without neighborhood mechanism. This means that it selects operators from neighborhood selected randomly to optimize solutions in each iteration. All operators are identical in original and modified version, the experimental results are shown in Fig. 6.

As shown in Fig. 6 , the original version is poorer than the modified version slightly in terms of delay time and waiting time, meanwhile superior to the modified version in terms of length substantially.

All individuals in population converge in two places for original version while all individuals converge in one place for modified version. The reason for this phenomenon is that when an operator performs well in the early stage of evolution, its neighborhood will be selected with a high probability in the process of evolution. The operators in the neighborhood also will be selected with a higher probability. So the probability of other operators selected from other neighborhoods will be reduced. At the same time, because the operator with outstanding performance in the early stage has no effect on individual improvement in the later stage in general. The probability of other operators selected from other neighborhood is still relatively low. Therefore, the algorithm is premature and converges to the local optimum because of this greedy idea. There is no neighborhood mechanism in modified version, as a result, each operator will be selected with a larger probability to produce offspring. So the algorithm will not fall into local optimum roughly.

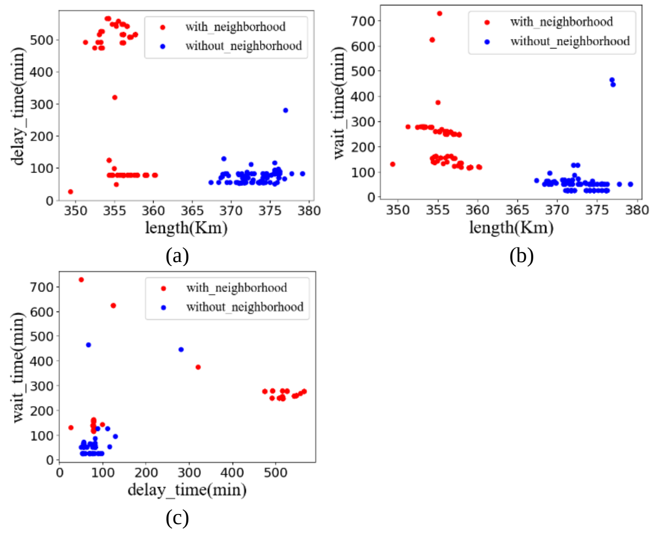


Fig. 6. Dominance of final solutions generated by algorithm with/without neighborhood mechanism

As shown in TAB. II, we make a comparison between the best solution in original version and in modified version. The performance of original version is better than modified version in terms of total length and delay time, and meanwhile inferior to the modified version in terms of the waiting time.

TABLE II. THE BEST SOLUTION OF FLGORITHM WITH/WITHOUT NEIGHBORHOOD MECHANISM

Best solution	*Total length (Km)*	*Delay time (min)*	*Wait time (min)*
original	349.342	26.09	129.88
modified	367.4	55.14	65.77

C. Performance comparison of Algorithm with/without Adaptive mechanism

In order to prove the effectiveness of the adaptive mechanism in this paper, we make a comparison between the original algorithm and a modified version without adaptive mechanism which means that it randomly selects an operator from all operators to optimize solutions in each iteration. All operators are identical in the original version and modified version. The experimental results are given in Fig. 7.

From the perspective of the convergence of the algorithm, the original version is superior to the modified version. The probability of all operators selected is equivalent in the modified version, which is unfair between operators with better performance and operators with worse performance, and this will lead to operators with worse performance being selected to generate offspring that are inferior to the parent individual. So the number of excellent operators used in the original version is much higher than that in modified version. So the convergence speed of the original version is faster than the modified version. As a result, the diversity of algorithm in original version is lower than the modified version. From the point of view of objective values, the original version is better than the modified version in aspect of length and delay time, and worse than the modified version in terms of wait time in general.

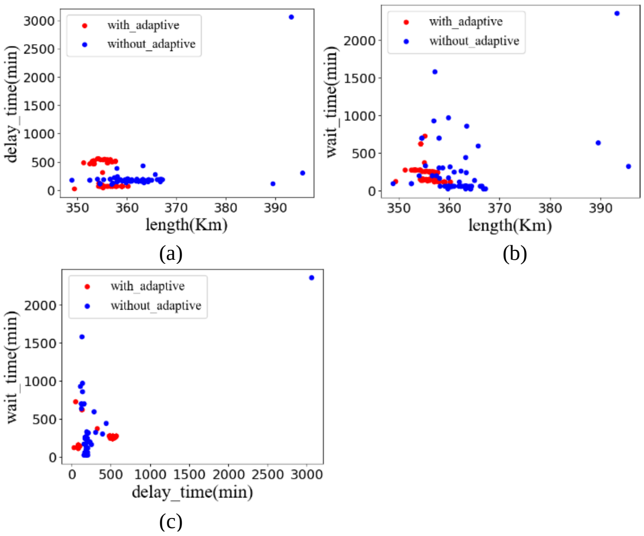


Fig. 7. Dominance of final soltions generated by algorithm with/without adaptive mechanism

TABLE III. THE BEST SOLUTION OF ALGORITHM WITH/WITHOUT ADAPTIVE MECHANISM

Best solution	***Total length (Km)***	***Delay time (min)***	***Wait time (min)***
original	349.342	26.09	129.88
modified	348.863	181.98	97.72

We also compare the dominant proportion p between solutions in the original version and the modified version. The definition of dominant proportion is shown in (12). num_A and num_B represent the number of solutions in original version and modified version. $num_{A_i \prec B}$ is the number of solutions in modified dominated by solution i in original version. From the view of dominant proportion, the performance of original version is better than modified version.

$$p = \left(\sum_{i=0}^{num\_A} \frac{num_{A_i \prec B}}{num_B} \right) / num\_B \qquad (13)$$

TABLE IV. THE DOMINANT PROPORTION OF SOLUTIONS SET IN ORIGINAL VERSION AND MODIFIED VERSION

Version	***Dominating***	***Non-dominated***	***Dominated***	***Equal***
original	7.76%	89.55%	2.69%	0
modified	2.69%	89.55%	7.76%	0

D. Performance comparison of Algorithm with/without DCMOEA

We compare original algorithm with modified version without using DCMOEA for proving the effectiveness of the DCMOEA in our algorithm. It indicates that all objectives and constraints are not converted and constraints is not processed by ε constraints handling method. All operators are identical in original and modified version. The experimental results are shown in Fig. 8.

We know that the advantage of the framework of DCMOEA is that it can transform objectives to constraints and use an advanced ε constraints handling method to deal with constraints. ε feasible method is used to compare offspring with parent. So the length is increasing gradually while the delay time and waiting time are decreasing by inches. It adopts Pareto domination method to compare offspring with parent in the process of evolution in modified version, which increases the pressure of selecting solutions and makes the reduction degree of delay time and waiting time low.

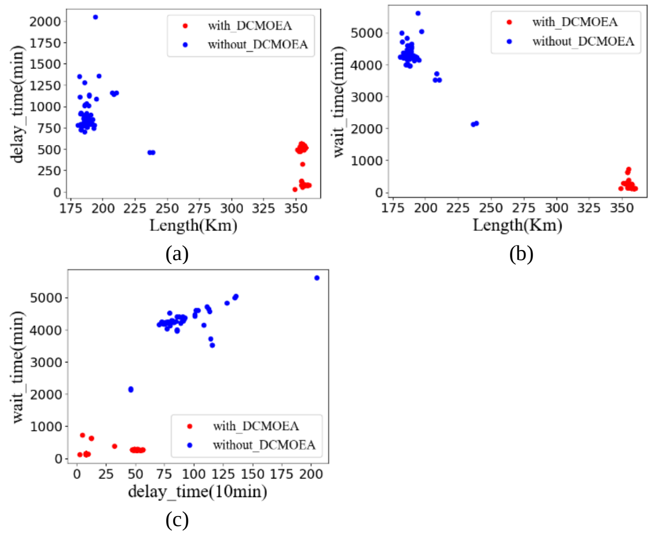


Fig. 8. Dominance of final solutions generated by algorithm with/without DCMOEA

E. Contribution Degree of Local Search Operators

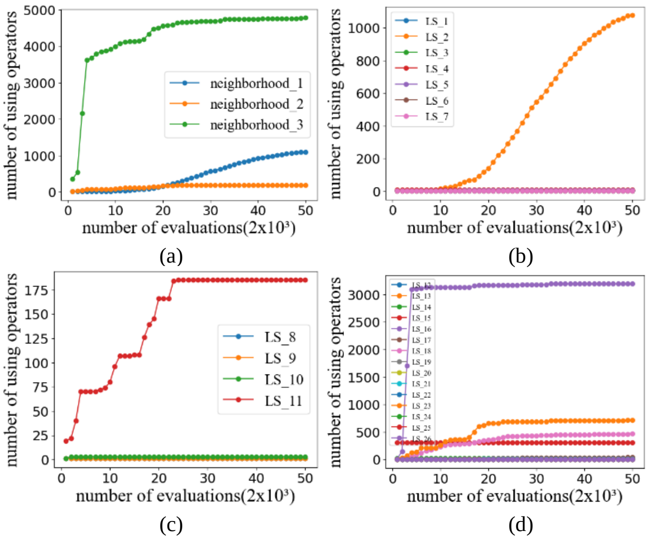


Fig. 9. The successful times of all local search operators used in algorithm without adaptive mechanism

In this paper, we design 26 local search operators. In order to illustrate the contribution degree of all operators, if the offspring individual generated by one operator is better than the parent individual in the process of evolution, then the contribution degree of this operator increase by 1. The contribution degree of all operators is given as in Fig. 9. In order to observe the convergence rate of the algorithm, we select the best individual in the current population every 2000 evaluations. The objective value of those best individuals changes with the number of evaluation are shown in Fig. 10.

As shown in Fig. 9 and Fig. 10, the performance of operator LS_16 in neighborhood 3 is excellent in the first 10000 evaluations. The operator contributes to the reduction of delay time and waiting time greatly. The number of successful evolution of multiple operators is increasing rapidly in the first 40000 evaluations, which leads to the rapid increasing of distance and the rapid decreasing of delay time and waiting time synergistically. The times of successful evolution of all operators tends to be stable after 50000 evaluations except for the local search operator LS_2, which is designed for changing severing orders between two customers at the same van.

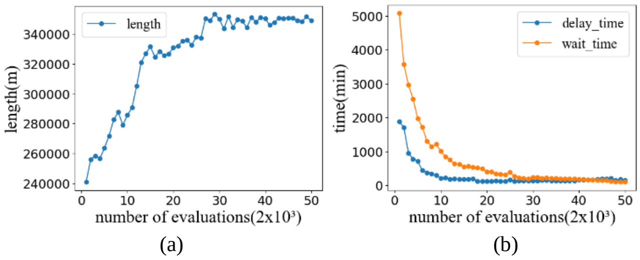


Fig. 10. The objective values of best soluions change with the number of evaluations

V. Conclusion

In this paper, we proposed a new bi-level VRP model with real road network and real-time traffic conditions. Two types of vehicles with different capacities are used to accomplish the task in this problem model. We also design two different clustering methods according to different heuristic information and choose one method of them to construct one solution randomly. In order to solve the above problem, we proposed an adaptive local search algorithm based on a dynamic constrained multi-objective evolutionary algorithm framework. At the same time, the performance of the algorithm is compared with 3 modified versions. Experimental results shown that our algorithm can address this problem effectively and be more competitive than the modified versions.

References

[1] G.B. Dantzig and J.H. Ramser, The truck dispatching problem. Management science. vol. 6,no. 1, pp. 80-91, 1959.

[2] S.W. Lin, Z.J. Lee, K.C. Ying, and C.Y. Lee, Applying hybrid meta-heuristics for capacitated vehicle routing problem. Expert Systems with Applications. vol. 36,no. 2, pp. 1505-1512, 2009.

[3] T. Vidal, Split algorithm in O (n) for the capacitated vehicle routing problem. Computers and Operations Research. vol. 69, pp. 40-47, 2016.

[4] R.A. Russell, Hybrid heuristics for the vehicle routing problem with time windows. Transportation science. vol. 29,no. 2, pp. 156-166, 1995.

[5] M. Mavrovouniotis and S. Yang, Ant colony optimization with memory-based immigrants for the dynamic vehicle routing problem. IEEE Congress on Evolutionary Computation. pp. 1-8, 2012.

[6] L. Xiao, C. Li, J. Wang, M. Mavrovouniotis, S. Yang, and X. Dan. Modeling and Evolutionary Optimization for Multi-objective Vehicle Routing Problem with Real-time Traffic Conditions. in Proceedings of the 2020 12th International Conference on Machine Learning and Computing. Shenzhen, China, pp. 518-523, 2020.

[7] G. Clarke and J.W. Wright, Scheduling of vehicles from a central depot to a number of delivery points. Operations Research. vol. 12,no. 4, pp. 568-581, 1964.

[8] F.Y. Vincent, A.A.N.P. Redi, Y.A. Hidayat, and O.J. Wibowo, A simulated annealing heuristic for the hybrid vehicle routing problem. Applied Soft Computing. vol. 53, pp. 119-132, 2017.

[9] B.M. Baker and M.A. Ayechew, A genetic algorithm for the vehicle routing problem. Computers and Operations Research. vol. 30,no. 5, pp. 787-800, 2003.

[10] S. Geetha, G. Poonthalir, and P.T. Vanathi, A hybrid particle swarm optimization with genetic operators for vehicle routing problem. Journal of advances in Information Technology. vol. 1,no. 4, pp. 181-188, 2010.

[11] Y. Marinakis, M. Marinaki, and G. Dounias, A hybrid particle swarm optimization algorithm for the vehicle routing problem. Engineering Applications of Artificial Intelligence. vol. 23,no. 4, pp. 463-472, 2010.

[12] B. Yu, Z.Z. Yang, and B. Yao, An improved ant colony optimization for vehicle routing problem. European journal of operational research. vol. 196,no. 1, pp. 171-176, 2009.

[13] S. Zeng, R. Jiao, C. Li, and J.S. Alkasassbeh, A general framework of dynamic constrained multiobjective evolutionary algorithms for constrained optimization. IEEE transactions on Cybernetics. vol. 47,no. 9, pp. 2678-2688, 2017.