A fast dynamic programming algorithm to a varied capacity problem in vehicle routing

Pengle Zhang*

College of System Engineering,
Academy of Military Sciences,
AnXiang North Road No. 10,
Beijing, China
Email: paulplayergg@gmail.com
*Corresponding author

Yajie Dou

College of Information System and Management,
National University of Defense Technology,
Changsha, Hunan, China
Email: yajiedou_nudt@163.com

Abstract: As a typical combinational optimisation problem, the researches on vehicle routing problem (VRP) mostly focus on the case when vehicle loading capability is certain. In fact, the vehicle loading capability often changes with multi-type vehicles, varying goods size and customer's requirement alter. Thus, a varied capacity vehicle routing problem (VCVRP) is introduced and a fast dynamic programming algorithm is presented based on the K-step best fit deceasing and minimum spanning tree methods, the algorithms can serve as an approximate decoupling between the goods packing problem and router selection problem in VCVRP. Besides, the theoretical analysis on the upper bound of vehicle trip and local minimisation based on short-path priority principle are carried out theoretically. Finally, an example about varied capacity vehicle logistics transportation task is given with quality and performance analysis on different parameters and scales, to illustrate the feasibilities and advantages of the proposed algorithm.

Keywords: logistic systems; dynamic programming; varied capacity vehicle routing problem; VCVRP; fast algorithm; heuristic algorithm.

Reference to this paper should be made as follows: Zhang, P. and Dou, Y. (2018) 'A fast dynamic programming algorithm to a varied capacity problem in vehicle routing', *Int. J. Applied Decision Sciences*, Vol. 11, No. 2, pp.146–167.

Biographical notes: Pengle Zhang received his MSc in Engineering from the National University of Defense Technology, Changsha, China, in 2015. Currently, he is an Associate Research Assistants at College of System Engineering, Academy of Military Sciences. His research interests include issues related to management science and operational research, multiple criteria decision making and risk analysis, machine learning and data visualisation.

Yajie Dou received his MSc in Engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2011. He received his Dr. degree in Engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2016. Currently, he is a Lecturer at College of Information System and Management, National University of Defense Technology. His main research interests include weapon alternative decision and effectiveness evaluation.

1 Introduction

The vehicle routing problem (VRP) was first introduced by Dantzig and Ramser (1959) as a typical coupling optimisation problem. According to the different kinds of factors taken in consideration, it can be sorted as different types (Laporte, 2009; Toth and Vigo, 2016). In particular, VRP with loading capability constraints is defined as capacitated vehicle routing problem (CVRP) (Potvin, 2009). As a common topic in recent researchers, the fundamental CVRP problem can be described as follows: Suppose that the nodes of a given network and several goal nodes with certain requirement have been specified, and the supplement is carried out by several vehicles with fixed loading capability, then the optimisation goal will be minimising the sum transportation consumption (Labbé et al., 2016; Perboli et al., 2016). In general, loading weight limit is often considered as the fixed loading capability for each type of vehicles, each type of vehicles is considered to have the same loading capability for all goods, though different type may have different loading capability. In fact, it varies with different length, width, height, fragility, weight and some other characteristics, which make the problem more complex (Louveaux and Salazar-Gonzãlez , 2016; Zhang et al., 2017). In this study, these problems are treated as varied capabilities vehicle routing problems (VCVRP).

In the past researches, VRP often was divided into two NP-hard problems, goods packing problem and router selection problem (Clarke and Wright, 1964; Laporte and Nobert, 1983; Chen and Ting, 2010; Lai et al., 2017). These two problems are combined so tensely when they influence the decision goal, in other words, different router selection plan will lead to different goods loading plan, which in doubt increases the computational scale of the coupling problem. Therefore, the main methods for CVRV problem are Heuristic algorithms, which commonly contain classical heuristic algorithm and improvement heuristic algorithm (Wei et al., 2015). There exists a rich literatures on these two kinds algorithm (Kucukoglu and Ozturk, 2015; Absi et al., 2017). For classical heuristic algorithm, it begin with a null solution set and add new solution into it to make a full set, such as savings algorithm (Li et al., 2016), inserting algorithm (Labbé et al., 1991), spanning tree algorithm (Lin, 2014) and so on. Improvement algorithm begins with an initial solution and improves the solution through many times iteration, such as k-opt algorithm (Han, 2011), tabu searching Algorithm (Gendreau et al., 2008; Cordeau et al., 2015; Silvestrin and Ritt, 2017), genetic algorithm (Filipec et al., 2015; Wang et al., 2017), deterministic annealing (Moura and Oliveira, 2009; Lai and Tong, 2017), ant colony algorithm (Fuellerer et al., 2009; Androutsopoulos and Zografos, 2017), particle swarm algorithm (Sun et al., 2015; Jia et al., 2017) and so on. However, there seldom exists efficient algorithm with complexity under $O(n^2)$ until recent research.

Moreover, in the VCVRP problem, good loading plan has not only weight limits but also some other constraints, such as size and volume of the goods, which increase the computational complexity of coupling problem. The traditional heuristics algorithms cannot sever to this problem directly and some improved algorithms are also complex in deciding the anti-linear loading constraints with a long searching time.

In this study, a fast dynamic programming algorithm is proposed to solve the large-scale VCVRP problem. In the first part, we use K step back and forth strategy to solve good packing problem with changing loading capability limit based on traditional best fit deceasing (BFD) algorithm, called K-BFD algorithm, and it has fast convergence effect with local greedy performance and can avoid local optimal to some extent. In the second part, minimum spanning tree (MST) router selection strategy is employed to reduce and optimise the scale of original router network. Then, short path priority principal is applied to formulate a dynamic programming model based on MST algorithm (DPM-MST). So it achieves an approximate decoupling solution to solve the goods packing problem and router selection problem. Finally, the upper bound of vehicle path is given and a theoretical analysis on local minimisation based on Short-path priority principle is carried out, which assure the validity of the proposed algorithm.

2 VCVRP problem analysis and modelling

2.1 VCVRP statement and analysis

CVRP problem has been widely studied, and there are many models and algorithms for solving CVRP problem. Most of the researchers assume a strong restriction, making the vehicle load capacity fixed, and the customer can be only served once by each vehicle, to cut down the number of the available transport programs and the scale of calculation. But they have ignored that a customer may need to be served by multiple vehicles, and the load capacity of vehicles is varied with different customers due to many customer-related restrictions, then the problem is in fact a VCVRP problem.

Generally, a VCVRP problem can be simply described as follows: Suppose that the nodes of a given network and several goal nodes with certain requirement, if the delivery is carried out by several vehicles with fixed loading capability and the requirement of every goal node is satisfied, how to construct a transportation plan to minimise the sum transportation consumption.

VCVRP problems are much closer to practical constraints, and can be seen as an extension of the problem CVRP problem. The loading capacity of the vehicle regarded as limited changes mainly in the following reasons:

- In the actual cargo loading problems, there are often many types of vehicles, and the limited loading capacity of different types of vehicles is different.
- In the actual cargo loading problems, there are often many types of goods (customer). Because of its different length, width, height, vulnerability, weight or other special features, the same type of vehicles have different limited loading capacity for different goods.
- Also, changes in customer demand, traffic congestion and other external information changes may cause the limited loading capacity changes in an actual cargo distribution process.

Traditional CVRP problems assume ignore these features and you cannot consider the impact of the limited capacity changes, so although there are many optimisation researches for CVRP problem, it is difficult to apply to practical problems. The reason above pointed out the limited capacity changes, and 'changes in customer demand, traffic congestion and other external information changes' also implies that the change is dynamic, VCVRP problems mentioned herein, considering only limited capacity may vary, focus on the first two insufficient points for CVRP to make the improvement.

2.2 VCVRP modelling

As the VCVRP problem defined above, a class of simple CVRP model is formulated as: Let $G = (V, E)$ be a graph where $V = \{0, 1, 2, \cdots, n\}$ is the vertex set representing one depot (Original node) with n nodes (Goal node) corresponding to customers. Edge set $E = \{(i, j)|0 \le i \ne j \le n\}$ represents all the side, and each two vertices denoted as d_{ij}. Suppose that there are p kinds of vehicles Q, and the loading capacity of weight noted as $Q_1, Q_2, \cdots, Q_p$. Respectively, the length and width of the h^{th} vehicle can be denoted as L_h and W_h. Let there be r kinds of goods g, where the length and width of g^{th} good can be expressed as l_g and w_g. Besides, the quantity of each kind of vehicles and goods is sufficient. Let y_{hki}^{g} be the amount of g^{th} good delivered to i^{th} customer by k^{th} vehicle of the h^{th} type, and let q_i^g be the need of i^{th} customer for g^{th} good. Suppose all the edges d_{ij} is known, the problem is to designing a shortest-path delivery routes using m_k vehicles of h^{th} kinds, where $0 \le m_h$, and $1 \le h \le p$, to delivery r kinds of goods to meet n customers' needs. Meanwhile, these constraints given as follows should be satisfied:

1. the loaded goods in a vehicle is not allowed to exceed its own loading capability limit Q_i
2. each vehicle should meet the space constraints F and the goods cannot be overlapped and exceed to the vehicle size
3. using vehicle should meet the quantity constraints C and there are using quantity matching among different vehicles
4. each vehicle should be loaded in the depot rather than customer node, where only unloading carry out.

Thus, the programming model of this problem can be built as follows:

Objective function:

$$min \sum_{h=1}^{p} \sum_{k=1}^{m_h} \sum_{i=0}^{n} \sum_{j=0}^{n} d_{ij} x_{ijk}^{h} \tag{1}$$

Subject to:

$$\sum_{i=1}^{n} \sum_{g=1}^{r} y_{hki}^{g} \le Q_h \ (1 \le h \le p, 1 \le k \le m_h) \tag{2}$$

$$\sum_{h=1}^{p}\sum_{k=1}^{m_h} y_{hki}^{g} \geq q_i^g \ (0 \leq i, j \leq n, 1 \leq g \leq r) \tag{3}$$

$$F(y_{khi}^g, l_g, w_g, L_h, W_h) = 1 \ (0 \leq i \leq n, 1 \leq h \leq p, 1 \leq k \leq m_h, 1 \leq g \leq r) \tag{4}$$

$$C\,(m_1, m_2, \cdots, m_p) = 1 \ (1 \leq p) \tag{5}$$

$$x_{ijk}^h > 0, \text{ if } y_{khj}^g > 0 \ (0 \leq i, j \leq n, 1 \leq h \leq p, 1 \leq k \leq m_h, 1 \leq g \leq r) \tag{6}$$

$$y_{khj}^g = 0, \text{ when } x_{ijk}^h = 0 \ (0 \leq i, j \leq n, 1 \leq h \leq p, 1 \leq k \leq m_h, 1 \leq g \leq r) \tag{7}$$

$$\sum_{j=1}^{n}\sum_{k=1}^{m_h} x_{0jk}^h = m_h \ (1 \leq j \leq n, 1 \leq h \leq p) \tag{8}$$

$$\sum_{i,j \in S \times S} x_{ijk}^h \leq |S| - 1, S \subset V_k^h, S \neq \emptyset \ (1 \leq h \leq p, 1 \leq k \leq m_h) \tag{9}$$

$$x_{ijk}^h = \begin{cases} 1, & \text{if } h^{\text{th}} \text{ kind of } k^{\text{th}} \text{ vehicle through arc}(i,j) \\ 0, & \text{else} \end{cases} \\ (0 \leq i, j \leq n, 1 \leq h \leq p, 1 \leq k \leq m_h) \tag{10}$$

$$F(y_{khi}^g, l_g, w_g, L_h, W_h) = \begin{cases} 1, & \text{if constraints } F \text{ are met} \\ 0, & \text{else} \end{cases} \\ (0 \leq i \leq n, 1 \leq h \leq p, 1 \leq k \leq m_h, 1 \leq g \leq r) \tag{11}$$

$$C\,(m_1, m_2, \cdots, m_p) = \begin{cases} 1, & \text{if quantity constraints are met} \\ 0, & \text{else} \end{cases} \quad (1 \leq p) \tag{12}$$

where V_k^h represents the customers set served by k^{th} vehicle of h^{th} kind. Equation (1) is the objective function minimising the total length of delivery route. Equation (2) are constraints for each vehicle loading capability and equation (3) are constraints each customer needs. Equation (4) are loading space constraints and equation (5) are constraints for available vehicle quantity. Equations (6) and (7) indicate that the vehicle only can serve the customer along its route. Equations (8) and (9) represent that the delivery route of vehicle is a simple chain starting at depot. In particular, if and only if $p = 1$, $r = 1$, and without regard for space constraints and quantity constraints, the problem is simplified as the classical CVRP problem.

With the consideration of the complexity O space,constraints F and quantity constraints C, VCVRP programming model is defined as nonlinear mixed integer programming model with variables x_{ijk}^h , y_{hkj}^g, m_1, m_2, $\cdots$, m_k. For a VCVRP problem with n customers, p types of vehicles and r kinds of goods,the amount of variables is $h + \sum\limits_{h=1}^{p} (nm_h r + n^2 m_h)$. Moreover, when the amount of customer, vehicle and the type of goods increases to k times, the computing scale will increase to k^3 times. Thus, the problem is a very complex nonlinear mixed integer programming problem and actual branch-and-bound algorithm cannot acquire the results in expected time. Besides, traditional heuristic algorithm aiming at solving nonlinear constraints also becomes to be complex. So a fast dynamic programming algorithm is proposed in this study to solve the large-scale VCVRB problem.

3 Preparation knowledge for algorithm design and analysis

VCVRP problem is a class of coupling optimisation problem essentially, which is consisted of two typical problems: goods packing problem and router selection problem (Fuellerer et al., 2010). The objective, minimising the total length of the entire vehicles' route, can be constructed by two sub-objectives:

1 maximising the vehicle utilisation ratio

2 minimising the delivery length of a vehicle.

In fact, it is easy to see that the interactions on results of the two sub-objectives are coupled. The problem is not a convex programming on convex set and may not have global consistency convergence. However, the decomposed sub-objectives can be regard to have the same convergence trend. So it is possible to approximately decouple by dynamic programming algorithm to solve VCVRP problem.

3.1 BFD algorithm

BFD algorithm is a kind of classical algorithm for solving one-dimensional goods packing problem. It ranks the goods in descending order and put each good into the most suitable bin. If there is not a suitable one for the goods, a null bin will be starting to be used. It can be seen that the thought is similar to greedy algorithm (GA) and based on this thought, the possible solution is gradually constructed as less space wasting as possible (Wang and Wu, 2014).

From the above description, it is known that the complexity of solving one-dimensional can be divided into ordering complexity and judging complexity. The algorithmic complexity of the fastest ordering algorithm currently for a sequence whose length is n is $O(n\log(n))$ (Nilsson, 2000), for goods packing problem in VCVRP problem, the length of goods sequence is $\sum\limits_{i=1}^{n}\sum\limits_{g=1}^{r} q_i^g$, So the ordering complexity is $O((\sum\limits_{i=1}^{n}\sum\limits_{g=1}^{r} q_i^g)\log(\sum\limits_{i=1}^{n}\sum\limits_{g=1}^{r} q_i^g))$. For the goods sequence whose length is n and the kinds containing boxes are m, if it is packed in the most suitable box at the worst case, the judging time is $2nm$, so the judging complexity is approximate $O(2p\sum\limits_{i=1}^{n}\sum\limits_{g=1}^{r} q_i^g)$. Thus, the whole complexity of BFD algorithm in solving one-dimensional packing problem is approximate:

$$O\left(\left(\sum_{i=1}^{n}\sum_{g=1}^{r} q_i^g\right)\log\left(\sum_{i=1}^{n}\sum_{g=1}^{r} q_i^g\right)+2p\sum_{i=1}^{n}\sum_{g=1}^{r} q_i^g\right)\approx O(nr\log(nr)+pnr) \tag{13}$$

3.2 MST algorithm

MST is minimal connected subgraph in connected graph serial, while weighted undirected connected graph is a tree that contains all the vertices in original graph and the sides whose total weights sum is minimum (Graham and Hell, 2007). Generally, the algorithms contain Kruskal algorithm, Boruvka algorithm and Prim algorithm, etc. To find a MST for connected graph with n vertices and m sides, the time complexity of Kruskal algorithm is $\mathrm{O}(\log_2^{\mathrm{m}} + 2\mathrm{m}\log_2^{\mathrm{n}} + \mathrm{n})$, which depends on the number of sides and is more suitable for sparse graph. Similarly, the time complexity of Boruvka algorithm is $\mathrm{O}(m\log_2^n)$. However, the time complexity of Prim algorithm is $\mathrm{O}(n^2)$, which is suitable for dense graph (Guttmann, 2016).

MST algorithm is often used to solve router selection and routing optimisation problems. In solving router selection problem, MST is often used to optimise and reduce the scale of transportation network. Based on this, the connected graph of the links between depots and all the customers can be transported into a MST with path optimisation strategy (Felipe et al., 2009). In this study, MST algorithm is mainly used to optimise and reduce the scale of transportation in solving VCVRP problem. Therefore, the complexity is mainly about constructing MST in this study while the complexity is $\mathrm{O}(n^2)$ for VCVRP problem solved by Prim algorithm.

3.3 DPM model

Dynamic programming model (DPM) is a programming method for solving multi-stage decision optimisation and many problems are handled by DPM which is more efficient than linear programming and nonlinear programming (Gao and Jiang, 2016). In 1951, on the basis of the characteristics of a class of multi-stage decision problems, it is transformed into a series of interconnected single-stage decision problem by R. Bellman and solved one by one, proposing the optimality principle for this kind problem (Lusby et al., 2010).

In solving the CVRP problem, there is a rich literature for DPM applied in goods packing problem and router selection problem. Some researchers proposed DPM model to carry out fast solving in goods packing problem with same size. The core thought of this model is in the basis of packing every time, joining the smallest goods parallel to a certain coordinate plane (Petersen et al., 2015). In solving router selection problem, DPM is employed by some researcher in searching dynamically the shortest path as the optimal solution, the core recursive relationship thought of which are dividing the vertices into different phases orderly and adding neighbour node with shorted path to acquire the solution gradually and recursively (Lusby and Larsen, 2011). Form the DPM thoughts, we can see that its complexity is mainly about stages dividing and recursive relationship solving for every stage. For VCVRP problem, the stages of DPM is divided by solving router selection problem and the dynamic programming recursion is realised by solving goods packing problem. The stage dividing complexity of router selection problem based on MST algorithm is $O(n^2)$, and the stage is n with its goods packing complexity $O((\sum_{g=1}^{r} q_i^g)\log(\sum_{g=1}^{r} q_i^g) + 2p\sum_{g=1}^{r} q_i^g)$ based on BFD algorithm, therefore, the approximate complexity of VCVRP problem solving by DPM:

$$O\left(n^2 + n\left(\sum_{g=1}^{r} q_i^g\right)\log\left(\sum_{g=1}^{r} q_i^g\right) + 2np\sum_{g=1}^{r} q_i^g\right) \approx O(n^2 + nrlog(r) + npr) \quad (14)$$

4 The principle and realisation of algorithm

From the above analysis, VCVRP problem can be divided into two sub-problems with approximate convergence: goods packing problem and router selection problem. For goods packing problem, in the basis of BFD algorithm, K-step back-and-forth strategy is introduced and K-BFD algorithm is proposed. With the consideration of space and quantity constraints, it has fast approximate effect with local greedy and can avoid local optimal in some extent. Based on these, the scale of original router network is optimised and reduced by the router selection strategy based on MST. Besides, DPM-MST model is proposed based on short-distance-priority principle, realising the approximate decoupling of goods packing problem and router selection problem.

4.1 Goods packing problem solving based on K-BFD algorithm

In VCVRP problem, the vehicle loading capability is varying due to the different types and sizes of the vehicle. On the basis of the VCVRP problem described in Section 2.2, the quantity constraints C of different vehicles and the space constraints F of vehicles and goods should be taken in consideration. Therefore, the problem is a nonlinear mixed integer programming problem and difficult to searching an exact solution. Firstly, the assessing index U is introduced, also written as $U(i, g, h, k)$, representing the satisfaction level of g^{th} goods from i^{th} customer for k^{th} vehicles of h^{th} type and greater is better. Its value is determined by two factors, the one is the bigger vehicle is better and the other one is the bigger space used by selected goods is better, so it can be formulated as the follows (Gao and Jiang, 2016).

$$U(i, g, h, k) = \frac{1}{\frac{L_h W_h}{1-C} + \frac{l_g w_g}{(1-F)\left(L_h W_h - \sum_{i=1}^{n}\sum_{g=1}^{r} y_{khi}^g l_g w_g\right)}} \quad (15)$$
$$(0 \le i \le n, 1 \le h \le p, 1 \le k \le m_h, 1 \le g \le r)$$

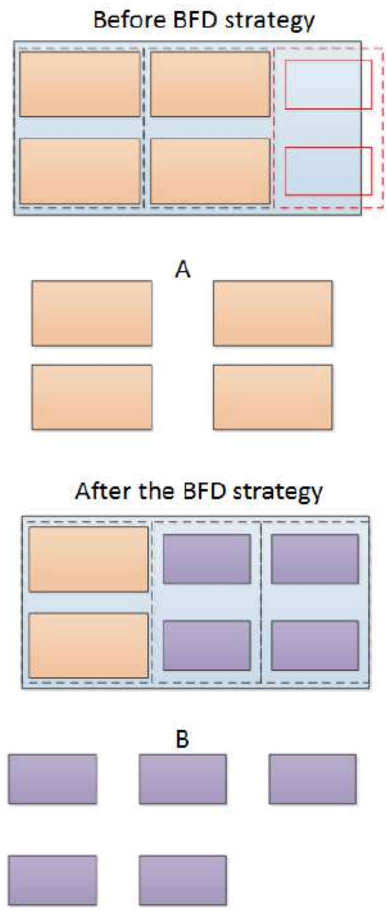
It is easy for BFD algorithm falling into a local optimal solution in computation, so the K-step back-and-forth strategy is introduced and K-BFD algorithm is proposed, which is given as the following rules:

Rule 1 According to the value of assessing index, the traditional BFD algorithm is used for loading.

Rule 2 The k-step back-and-forth strategy is introduced for BFD improvement.

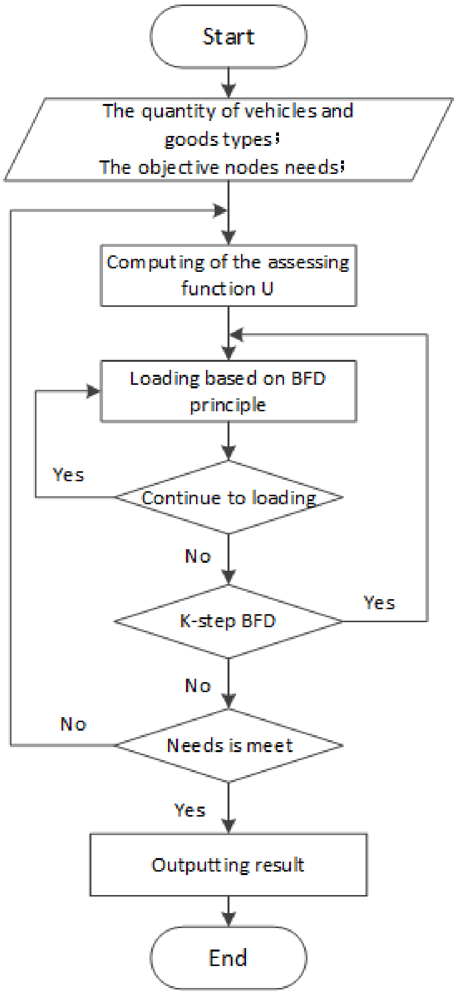
In the loading process with descending order, if there are goods unable to be loaded, the strategy will be not to start using next new vehicle but to adjust the loaded unit in the former vehicle and carry out a heuristic searching constructing smaller unit. The searching steps number is K and the objective is to reduce the space of current vehicle for loading, which is called as K-step back-and forth strategy. It can improve the usage rate of vehicle space further and the algorithm will be traditional BFD when K is equal to 0.

Figure 1 The illustration of K-step back-and-forth strategy (see online version for colours)



When BFD algorithm is used to load, the condition described in Figure 1 will come out, i.e., if the loaded goods hold more space, but it cannot hold any one more, the K-step back-and-forth strategy can be applied to replace the former goods with the k latter goods and the former one will be replaced with the new one which has the most space usage rate, realizing a higher space rate of the vehicle.

Figure 2 The K-BFD algorithm process



According to the rules above, the process of K-BFD algorithm can be expressed as Figure 2.

4.2 VCVRP problem solving based on DPM-MST model

The K-BFD algorithm is mainly for goods packing problem solving, but for VCVRP problem, there still has another problem-router selection problem. Moreover, the two problems are not separate but coupling for the entire optimisation objective. In other word, the loading plan will influence the router selection strategy directly and the router optimisation will lead to the adjustment of loading plans.

In solving VCVRP problem, DPM-MST model uses dynamic programming theory to decouple the goods packing problem and router selection problem approximately and consider the nodes transition of router at different stages, realising to load goods dynamically during every stage. For DPM-MST model, the core parts are MST generation, router selection and goods packing dynamically.

For the generation algorithm for MST, it is described in detail in Section 3.2. The Prim algorithm is employed in this study and the time complexity is $O(n^2)$. It is worthy to point out that in DPM-MST model, in basis of MST, even if router selection and dynamical goods packing cannot guarantee that the total delivery path shortest, but they avoid the longer side in original network and reduce the scale of the network greatly. Besides, they provide the basis for decoupling the goods packing problem and router selection problem in VCVRP problem.

The basic thoughts of router selection and dynamical packing can be described as follows: according to the tree structure, the nodes on the branches are supplied one by one, i.e., carry out the packing problem optimisation for some node, and for the lower-level node, the packing can use the space left by the high-level node. Therefore, the entire space usage rate is improved and the amounts of vehicle used in delivery are reduced. Meanwhile, when there is a low-level branch, in the basis of short-path-priority principle, the rest loading space should be supply for the low-level node with short-path first to reduce the whole length of the path.

Definition 1 (Dynamical packing): According to the tree structure, it starts at the root node based on the requirements of each branch and leaf node to load goods for them one by one using depth-first principle. For each node, the goods packing algorithm is K-BFD, if the node is not a leaf node and there exist not full boxes, their rest space will be used by next low-level nodes. Similarly, goods packing for each node can be finished until the last node is leaf node. And goods packing for each branch will be finished until all the leaf nodes are satisfied.

Definition 2 (Short path priority): For a given MST, if the current node has branches and there are available boxes in this node, the rest space will be provided for the low-level node with shortest path first from the current node.

In the basis of the definitions given above, the total path length is analysed and the final solution type is discussed, we can obtain the following propositions:

Proposition 1: For a given network $G(V, E)$, because of the dynamic packing and router selection strategy of MST $T(V', E')$, the total delivery length has an upper bound, i.e., $L \leq B_{\max} \sum_{h=1}^{p} m_h$, where L represents the total delivery path and m_k represents the vehicle number of k^{th} type, B_max is the length of the longest branch of the MST.

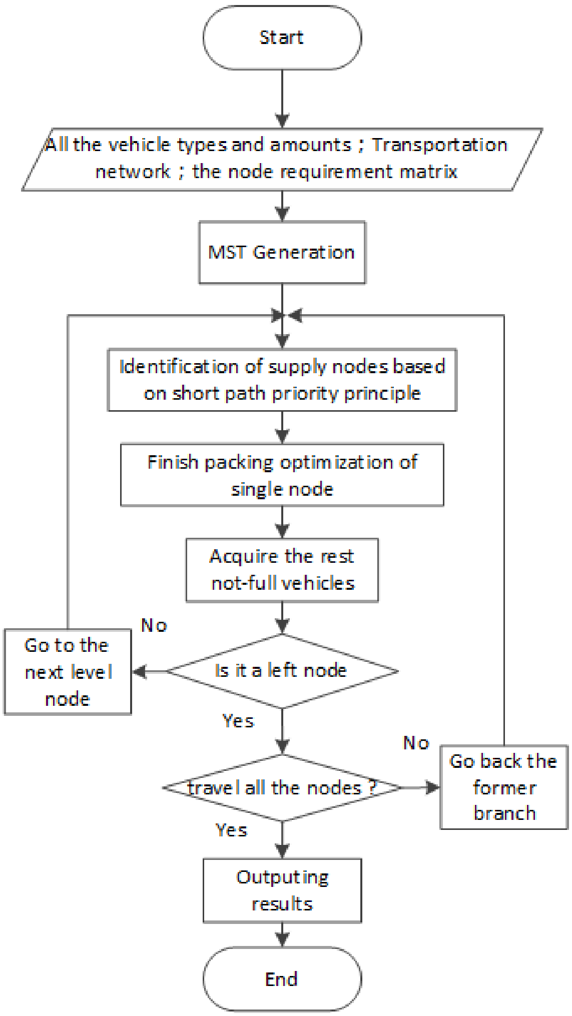
Proof: The length of total delivery path is the sum of the routing paths for all the vehicles, which can be obtained by the following formulation: $L = \sum_{h=1}^{p} \sum_{k=1}^{m_h} b_{hk}$, where b_hk indicates the delivery path of i^{th} vehicle of h^{th} type. Based on the above dynamical packing and router selection strategy, the delivery path of each vehicle must contain the branch of a certain tree. Thus, the delivery path of the vehicle $b_{hk} \leq B_j$, where

B_i represents the total length of the branch. So that $L = \sum_{h=1}^{p} \sum_{k=1}^{m_h} b_{hk} \leq \sum_{h=1}^{p} B_j m_h$, and because of $B_j \leq B_{\max}$, the equation $L \leq B_{\max} \sum_{h=1}^{p} m_h$ is right.

Proposition 2: For a given MST $T(V', E')$, each branch/leaf node V_s can reach its total delivery path local optimal result based on short path priority principle.

Proof: As the dynamical packing process given by Definition 1, the not full boxes from the former level node will be used to be loaded in the next low-level nodes. All the vehicles ending at this node must stop at this time and the route and delivery path are all fixed. Therefore, for the node, minimising the delivery path length of the not full vehicles is equal to minimising the total delivery path of the node, i.e., realising the optimal of the delivery path of this node.

Figure 3 Illustration of DPM-MST model procedure



Suppose that the router lies in stage s when node V_s transfer to node $V_s + 1$, V_s represents all the nodes set in the first s stages. Thus, the recursive relation for dynamical programming from s stage to $s + 1$ stage:

$$\begin{cases} v_{s+1} = \{j| \underset{j \neq v_s, v_j \notin V_s}{\arg \min} d_{sj}\} \\ v_1 = 0 \end{cases} \quad (1 \leq s \leq n) \tag{16}$$

The realisation process of DPM-MST model can be given as Figure 3.

4.3 The complexity of DPM-MST model

According to Section 4.2, the computation assumption of DPM-MST model is mainly about solving MST, router selection and goods packing of each node. For a VCVRP problem with p types of vehicles, r types of goods and n nodes, suppose that the complexity of its space constraints F and quantity constraints C are $O(F)$ and $O(C)$, respectively. As introduced in Section 3.1, the complexity of K-BFD is comprised of BFD complexity, the computation assumption of K-step back-and-forth, and the complexity of space constraints F and quantity constraints C. So the goods dynamical packing complexity is approximate as the follows:

$$\begin{aligned} &O\left(\left(\sum_{i=1}^{n}\sum_{g=1}^{r} q_i^g\right)\log\left(\sum_{i=1}^{n}\sum_{g=1}^{r} q_i^g\right)\right) \\ &+ O\left(2p\sum_{i=1}^{n}\sum_{g=1}^{r} q_i^g + kp\sum_{i=1}^{n}\sum_{g=1}^{r} q_i^g\right)\mathrm{O}(F)\mathrm{O}(C) \\ &\qquad \approx O(nr\log(nr)) + O(kpnr)\mathrm{O}(F)\mathrm{O}(C) \end{aligned} \tag{17}$$

We can see from equation (22) in Section 3.3 that the complexity of MST generation of DPM-MST based on MST algorithm is $O(n^2)$, and the complexity of router selection strategy based on short path priority principle is $O(n)$, so the computing time complexity is about to be:

$$\begin{aligned} &O(n^2) + \mathrm{O}(n) + \mathrm{O}\left(n\left(\sum_{i=1}^{n}\sum_{g=1}^{r} q_i^g\right)\log\left(\sum_{i=1}^{n}\sum_{g=1}^{r} q_i^g\right)\right) \\ &+ O\left(2np\sum_{i=1}^{n}\sum_{g=1}^{r} q_i^g + knp\sum_{i=1}^{n}\sum_{g=1}^{r} q_i^g\right)\mathrm{O}(F)\mathrm{O}(C) \\ &\qquad \approx O(n^2) + O(n) + O(n^2 r\log(nr)) + O(kpn^2 r)\mathrm{O}(F)\mathrm{O}(C) \\ &\qquad \approx O(n^2 r\log(nr)) + O(kpn^2 r)\mathrm{O}(F)\mathrm{O}(C) \end{aligned} \tag{18}$$

When $p = 1$ and $r = 1$, it means that the space constraints and quantity constraints are regardless, so the problem is simplified as a classical CVRP problem, so the computing time complexity of DPM-MST model is $O(n^2 log(n))$.

5 Case study and analysis

5.1 Illustrative example

We select vehicle logistics of sedan car as the background of the case study and vehicle logistics (VL) is the entire process of vehicle fast delivery according to customers' needs. The task, transporting various types of sedan cars to the whole country, is allocated by the producer of sedan car to logistics companies, which make transportation plans and allocate these sedan cars in the basis of tasks. A simple numerical example is given first to illustrate the DPM-MST model proposed in this study. Then, an example of solving large-scale VCVRP problem is given to explain the computing performance of DPM-MST model. These results can do much for managerial implication.

Table 1 Mode and size

Mode	*The length of all levels (m)*	*The high level width (m)*	*The low-level width (m)*
1-1	19	2.7	2.7
1-2	24.3	3.5	2.7
1	4.61	-	1.7
2	3.615	-	1.605

Figure 4 Node connection relationship graph

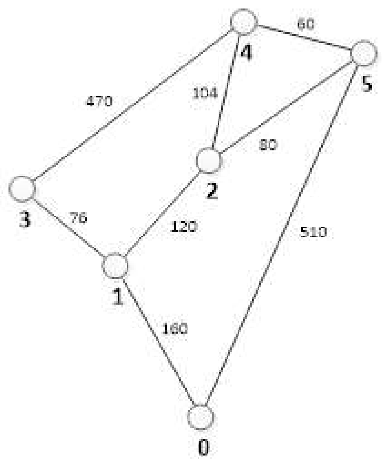


Table 2 Goods requirements of each node

Node	*1*	*2*	*3*	*4*	*5*
1	42	50	33	41	0
2	31	0	47	0	0

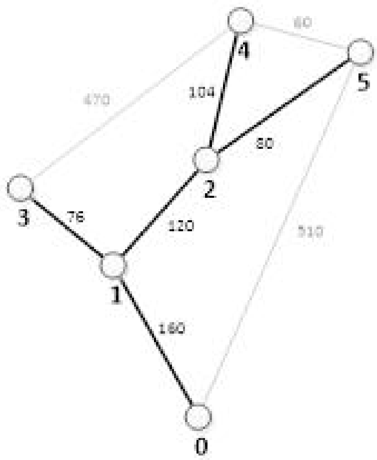
Numerical example of VCVRP problems can be described as follows: Assume that there are $p = 2$ vehicle (box) types, all are bunk truck with enough storage height limit, their height limit can be ignored, 1-1 and 1-2 type respectively. Assume that there are $r = 2$

cars (goods) types, type 1 and type 2 respectively, the exactly size are shown in Table 1. Set the number of nodes $n = 5$, $V = \{0, 1, 2, 3, 4, 5\}$, the relationships of each node connected and mutual distances are shown in Figure 4, the requirement of each node is shown in Table 2. Where space constraints F is the shape of the constraint length and width, the quantity constraints is the ratio for the constraint, wherein the number of no more than 1-1 1-2 Type 20.

5.2 Results

To solve according to DPM-MST, the MST in transport network should be conducted, the result is shown as Figure 5.

Figure 5 An illustration of MST



Then, the stages are divided by dynamic programming. Each stage is loaded dynamically and back-and-forth parameter $k = 5$, details are shown as below:

- Initialisation: $v_1 = 0$
- The first stage: $s = 1$, $v_2 = \{j| \underset{j \neq v_1, v_j \notin V_1}{\arg \min} d_{1j}\} = 1$, use K-BFD algorithm to conduct dynamic loading, there into the number of 1-1 type is 5 and the number of 1-2 type is 1. The last vehicle is not filled up and the node is not leaf node, then it will be considered in the next dynamic loading period.
- The second stage: $s = 2$, $v_3 = \{j| \underset{j \neq v_2, v_j \notin V_2}{\arg \min} d_{2j}\} = 3$, dynamic loading, the 1-2 type vehicle which is not filled up is considered firstly, thereinto the last 1-1 type vehicle is still not filled up. This vehicle will not be considered in next loading because it is a leaf node.

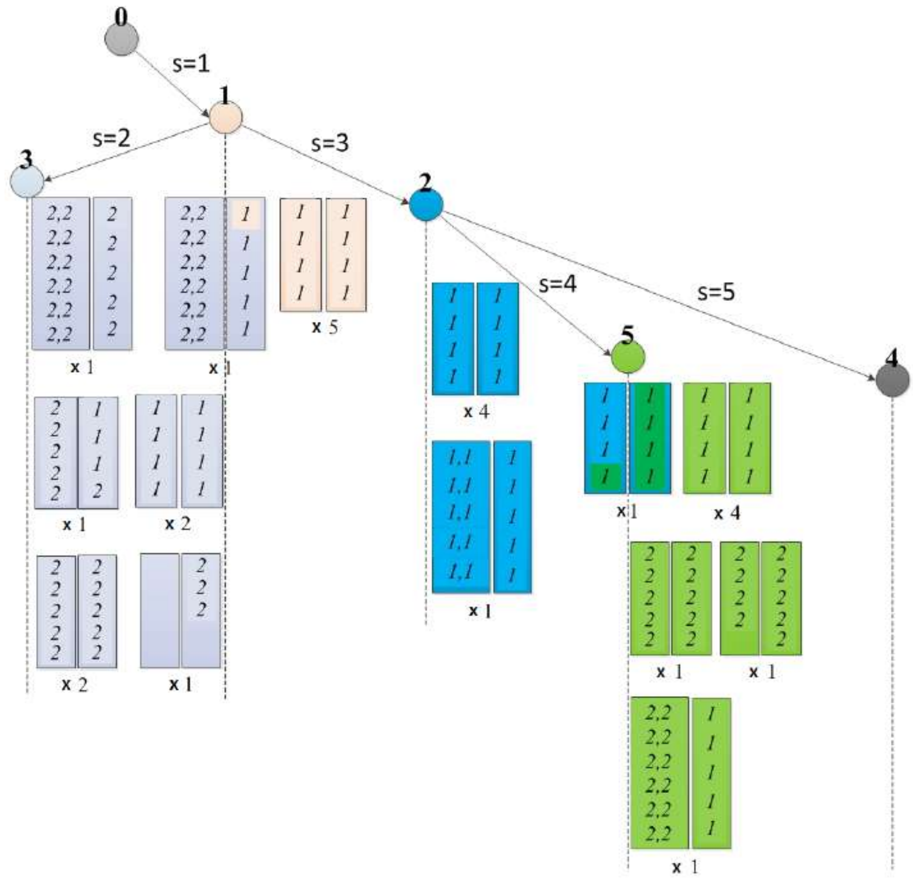
Similarly, the results of each stage will be obtained, they are shown in Table 3.

Table 3 Results of VCVRP

Back and forth parameter	*1-1 type*	*1-2 type*	*Total number of vehicle*	*Total length of the path*	*Average usage rate of vehicle*
$K = 5$	22	4	26	6,968	94.63%

The detail arrangement is shown as Figure 6. There into it means a vehicle is transported to a node when they have the same colour and the vehicle which has two colours will stop at one node and then be transported to the other node. Compare with traditional CVRP methods shown in Potvin (2009), we have promoted the efficiency from less than 70% to 94.63% to show the results can be validated and verified.

Figure 6 Detail arrangement (see online version for colours)



From these results, the manager can easily make an arrangement. The number of each type vehicles, the vehicle packing solution for the goods and the vehicle routing way for the customers can be easily set. Even when the customer changes their demand of requirement, the dynamic programming algorithm can adjust the solution in each dynamic step. That was quite valuable for managerial implication.

5.3 The computational performance analysis of DPM-MST

In DPM-MST model the VCVRP problem is solved by dynamical programming method and its computing time complexity is approximate: $O(n^2 r \log(nr)) + O(kpn^2 r)\mathrm{O}(F)\mathrm{O}(C)$, thus, it is a fast algorithm according to the computation time. In spite of the computing time, the solving quality and parameter sensitivity are also important. In this study, the performance of DPM-MST model is analysed form total length of path, the average usage rate of vehicle and computation time cost.

5.3.1 Parameter K influence analysis

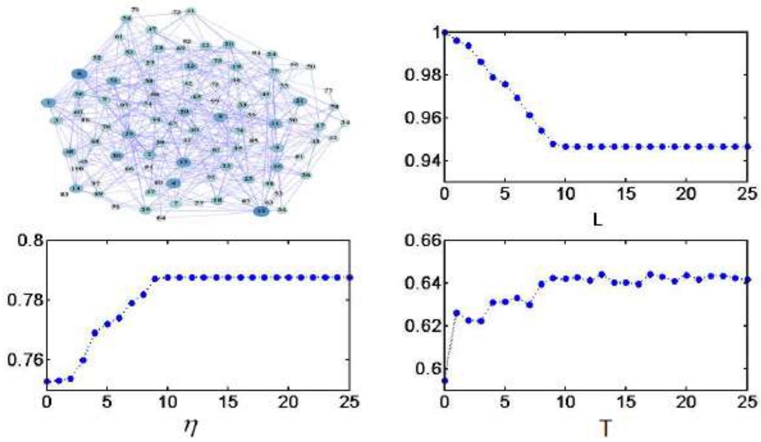
In DPM-MST, there is only one parameter k need to be identified, DPM-MST. The increasing of K will lead to the increasing chances of back-and-forth adjustment, bringing out much more better solutions. However, it can be seen from equation (26) that the increasing K will increase the time assumption of computation, so in order to analyse the influence of parameter K enough, it is necessary to reconstruct the illustration example. Let be the number of nodes is $n = 100$, and he relationship and distances are generated in random, $p = 5$, $r = 10$, similarly, the sizes of vehicle and goods are also generated in random. Let the value of K to be a number between 0 to 100, and the value increase is 5. The parameter K influence analysis is carried out from the change of total path L length ($L_max = 1$), the average usage rate η, and computation time T. The results are shown as Table 4 and Figure 7.

Table 4 The influence analysis result of parameter K

VCVRP: $n = 100, p = 5, r = 10$					
K	*Performance* (L, η, T)	K	*Performance* (L, η, T)	K	*Performance* (L, η, T)
0	1.0000, 0.7528, 0.5941	1	0.9961, 0.7530, 0.6259	2	0.9936, 0.7537, 0.6228
3	0.9862, 0.7601, 0.6222	4	0.9787, 0.7691, 0.6308	5	0.9758, 0.7720, 0.6313
6	0.9693, 0.7740, 0.6330	7	0.9612, 0.7790, 0.6298	8	0.9540, 0.7820, 0.6394
9	0.9477, 0.7870, 0.6421	10	0.9464, 0.7876, 0.6418	11	0.9464, 0.7876, 0.6424
12	0.9464, 0.7876, 0.6412	13	0.9464, 0.7876, 0.6441	14	0.9464, 0.7876, 0.6401
15	0.9464, 0.7876, 0.6401	16	0.9464, 0.7876, 0.6394	17	0.9464, 0.7876, 0.6441
18	0.9464, 0.7876, 0.6431	19	0.9464, 0.7876, 0.6409	20	0.9464, 0.7876, 0.6437
21	0.9464, 0.7876, 0.6415	22	0.9464, 0.7876, 0.6433	23	0.9464, 0.7876, 0.6434
24	0.9464, 0.7876, 0.6423	25	0.9464, 0.7876, 0.6414		

We can see from the above figure, with the back-and-forth parameter K increasing, the quality of computation results are improved obviously in the aspects of total path length and average usage rate. When $K > 10$, if the parameter increase, the average usage rate of vehicle cannot be increased anymore and the total length of path also cannot be reduced anymore.

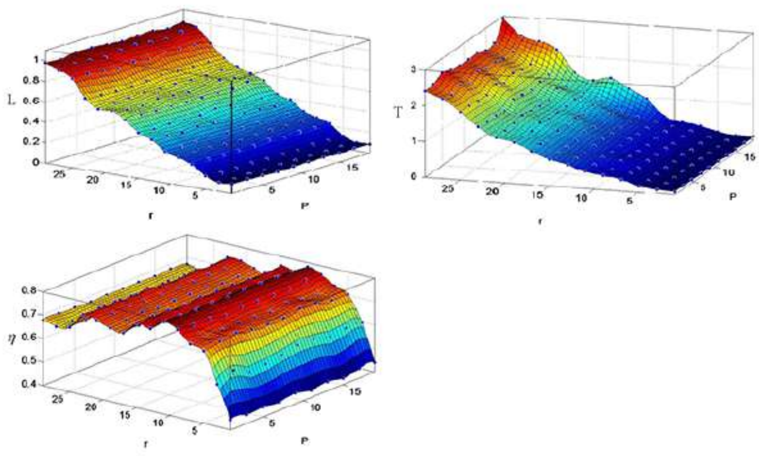
Figure 7 The influence analysis result of parameter K (see online version for colours)



5.3.2 *Limited capability changing influence analysis*

In contrast with CVRP problem, the big difference of VCVRP problem lies in the changing loading capability limits. As described in Section 2.2, it is necessary to further explore the computation performance of DPM-MST model for changing capability limits. Let be $n = 100$, $k = 10$, $p = 1, 3, 5, \cdots, 17, 19$, and $r = 1, 3, 5, \cdots, 27, 29$. Assume that the parameters are mixed pairwise and the solving results can be shown as Figure 8.

Figure 8 The influence analysis with capability limits changing (see online version for colours)



If the scale of nodes and back-and-forth parameter are fixed, with regard to the total length of delivery path, the capability limit will increase with parameters p, r increasing, leading to the increase of L. In DPM-MST, the increase of p has a little influence on L but r has great influence on L. It indicate that the total length of delivery path is influenced by the types of goods more obviously, which is consistent with the result of complexity analysis.

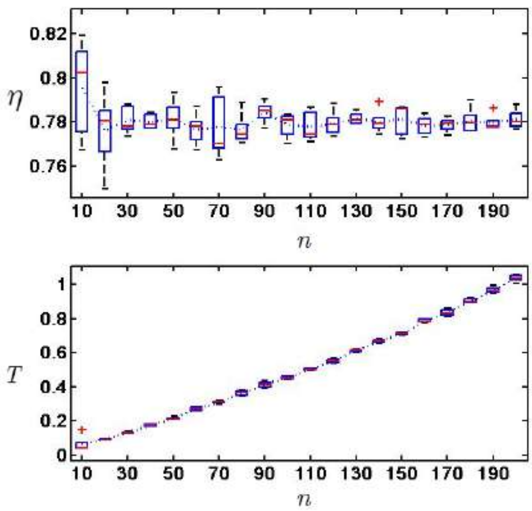
With regard to the average usage rate of vehicle, the increasing p, r, and types will improve the average usage of vehicle at some extent. However, the increase of p is small in contrast with L, when the value of r increase more, the capability limit changing greatly, but due to the approximate effect of DPM-MST fast algorithm, the average usage rate is unchanging.

With regard to the computing time, the increasing p, r leads to the changing capability limits and solving time assumption. Generally, the approximate linearity increase with increasing.

5.3.3 Influence analysis of node scale

DPM-MST model solving is a kind of efficient dynamic algorithm. To perform the simulating efficiency, the influence of node scale is analysed. Let be $k = 10$, $p = 5$, $r = 10$, and $n = 10, 20, 30, \cdots, 190, 200$, then conduct the solving. When the node scale is increased, the solving quality cannot be represented by the total length of the route, so it is mainly analysed by the average use ratio of the vehicle and consume time of solving. The results are shown in Figure 9.

Figure 9 Node scale influence (see online version for colours)



When the node scale is increased, the average use ratio of vehicle keeps around 78%, it declares that DPM-MST is appropriate to solve large-scale VCVRP problems and the fluctuation of solving quality is small for problems of different scales. Consuming time of solving is increased along with the increase of node scale and the increase of simulation time is much less than exponential order trend. The increase curve is smooth and it means the approximate computation complexity $O(n^2 log(n))$ analysed before it is correct. Considering VCVRP is the NP-hard problem, which means an acceptable time consuming with node scale increase. And we also calculated the average usage rate towards different scale which influenced a little. That is to say our method is robust with node scale increase without much accuracy decline.

6 Conclusions

An efficient dynamic DPM-MST model, which can be used to solve large-scale VCVRP, is proposed based on dynamic programming theory and combined with heuristics rules. VCVRP problem is divided into two NP-Hard problems, goods packing problem and router selection problem. These two problems are decoupling by the DPM-MST model. Besides, the algorithm of vehicle haul distance is proposed and the validity of this algorithm is insured by the theoretical analysis of short distance priority principle. And the model is analysed both in the fields of theoretical analysis and case study. It is shown in the results of the examples that when handling with a very big scale VCVRP, our methods could calculate out pretty quick with a satisfied result. Also the time consuming fluctuation of quality is nearly liner influenced by the range of nodes. It is very efficient for VCVRP problems. DPM-MST is appropriate to solve VCVRP problems which have a large scale and need high efficiency.

Considering the capability limit in the VCVRP problems is influenced by vehicle type, goods feature, and the customer demand change in logistics distribution. The first two characteristic have been discussed deep and clear, while the treatment for customer demand change is weak. We have settled the customer change only by adjustment during each dynamic programming step. Also, the capability change of VCVRP has its stage characters, and it is supposed to consider the stage characters in the dynamic programming model.

It is the key point and direction to improve the performance of this algorithm for customer demand change. Also how can the results be generalised to the other related fields, such as resource allocation, schedule arrangement, multi-object programming and so on, should be considered and analysed.

References

Absi, N., Cattaruzza, D., Feillet, D. and Housseman, S. (2017) 'A relax-and-repair heuristic for the swap-body vehicle routing problem', *Annals of Operations Research*, June, Vol. 253, No. 2, pp.957–978.

Androutsopoulos, K.N. and Zografos, K.G. (2017) 'An integrated modelling approach for the bicriterion vehicle routing and scheduling problem with environmental considerations', *Transportation Research Part C: Emerging Technologies*, September, Vol. 82, pp.180–209.

Chen, C.H. and Ting, C.J. (2010) 'Applying two-stage ant colony optimization to solve the large scale vehicle routing problem', *Journal of the Eastern Asia Society for Transportation Studies*, Vol. 8, pp.761–776.

Clarke, G.U. and Wright, J.W. (1964) 'Scheduling of vehicles from a central depot to a number of delivery points', *Operations Research*, Vol. 12, No. 4, pp.568–581

Cordeau, J., Gendreau, M. and Laporte, G. (2015) 'A tabu search heuristic for periodic and multi-depot vehicle routing problems', *Networks*, Vol. 30, No. 2, pp.105–119.

Dantzig, G.B. and Ramser, J.H. (1959) 'The truck dispatching problem', *Management Science*, Vol. 6, No. 1, pp.80–91

Felipe, A., Ortuño, M.T. and Tirado, G. (2009) 'New neighborhood structures for the double traveling salesman problem with multiple stacks', *TOP*, Vol. 17, No. 1, pp.190–213

Filipec, M., Skrlec, D. and Krajcar, S. (2015) 'Genetic algorithm approach for multiple depot capacitated vehicle routing problem solving with heuristic improvements', *International Journal of Modelling and Simulation*, Vol. 522, No. 2, pp.520–526.

Fuellerer, G., Doerner, K.F., Hartl, R.F. and Iori, M. (2009) 'Ant colony optimization for the two-dimensional loading vehicle routing problem', *Computers & Operations Research*, Vol. 36, No. 3, pp.655–673.

Fuellerer, G., Doerner, K.F., Hartl, R.F. and Iori, M. (2010) 'Metaheuristics for vehicle routing problems with three-dimensional loading constraints', *European Journal of Operational Research*, Vol. 201, No. 3, pp.751–759.

Gao, W. and Jiang, Z.P. (2016) 'Adaptive dynamic programming and adaptive optimal output regulation of linear systems', *IEEE Transactions on Automatic Control*, Vol. 61, No. 12, pp.4164–4169.

Gendreau, M., Iori, M., Laporte, G. and Martello, S. (2008) 'A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints', *Networks*, Vol. 51, No. 1, pp.4–18.

Graham, R.L. and Hell, P. (2007) 'On the history of the minimum spanning tree problem', *Annals of the History of Computing*, Vol. 7, No. 1, pp.43–57.

Guttmann, W. (2016) 'Relation-algebraic verification of Prim's minimum spanning tree algorithm', in Sampaio, A. and Wang, F. (Eds.): *Theoretical Aspects of Computing – ICTAC 2016. Lecture Notes in Computer Science*, Vol. 9965, pp.51–68, Springer, Cham.

Han, S.H. (2011) 'A comparison between genetic algorithm and k-opt local search method for the vehicle routing problem', *NUCB Journal of Economics & Information Science*, Vol. 56, No. 1, pp.97–105.

Jia, Y.H., Chen, W.N., Gu, T. et al. (2017) 'A dynamic logistic dispatching system with set-based particle swarm optimization', *IEEE Transactions on Systems Man & Cybernetics Systems*, No. 99, pp.1–15.

Kucukoglu, I. and Ozturk, N. (2015) 'An advanced hybrid meta-heuristic algorithm for the vehicle routing problem with backhauls and time windows', *Computers & Industrial Engineering*, Vol. 86, pp.60–68.

Labbé, M., Laporte, G. and Mercure, H. (1991) 'Capacitated vehicle routing on trees', *Operations Research*, Vol. 39, No. 4, pp.616–622.

Labbé, M., Laporte, G. and Mercure, H. (2016) 'Capacitated vehicle routing on trees', *Operations Research*, Vol. 39, No. 2, pp.616–622.

Lai, M. and Tong, X. (2017) 'A metaheuristic method for vehicle routing problem based on improved ant colony optimization and tabu search', *Journal of Industrial & Management Optimization*, Vol. 8, No. 2, pp.469–484.

Lai, M.Y., Liu, C.S. and Tong, X.J. (2017) 'A two-stage hybrid meta-heuristic for pickup and delivery vehicle routing problem with time windows', *Journal of Industrial & Management Optimization*, Vol. 6, No. 2, pp.435–451.

Laporte, G. (2009) 'Fifty years of vehicle routing', *Transportation Science*, Vol. 43, No. 4, pp.408–416.

Laporte, G. and Nobert, Y. (1983) 'A branch and bound algorithm for the capacitated vehicle routing problem', *Operations-Research-Spektrum*, Vol. 5, No. 2, pp.77–85.

Li, H., Chang, X., Zhao, W. et al. (2016) 'The vehicle flow formulation and savings-based algorithm for the rollon-rolloff vehicle routing problem', *European Journal of Operational Research*, Vol. 257, No. 3, pp.859–869.

Lin, S. (2014) 'Computer solutions of the traveling salesman problem', *The Bell System Technical Journal*, Vol. 44, No. 10, pp.2245–2269.

Louveaux, F.V. and Salazar-Gonzãlez, J.J. (2016) 'Solving the single vehicle routing problem with variable capacity', *Transportation Science*, Vol. 50, DOI: 141223041352002.

Lusby, R.M. and Larsen, J. (2011) 'Improved exact method for the double TSP with multiple stacks', *Networks*, Vol. 58, No. 4, pp.290–300.

Lusby, R.M., Larsen, J., Ehrgott, M. and Ryan, D. (2010) 'An exact method for the double TSP with multiple stacks', *International Transactions in Operational Research*, Vol. 17, No. 5, pp.637–652

Moura, A. and Oliveira, J.F. (2009) 'An integrated approach to the vehicle routing and container loading problems', *OR Spectrum*, Vol. 31, No. 4, pp.775–800.

Nilsson, S. (2000) 'The fastest sorting algorithm', *Dr Dobbs Journa: Software Tools for the Professional Programmer*, Vol. 25, No. 4, pp.38–42.

Perboli, G., Tadei, R. and Vigo, D. (2016) 'The two-echelon capacitated vehicle routing problem: models and math-based heuristics', *Transportation Science*, Vol. 45, No. 3, pp.364–380.

Petersen, H.L., Archetti, C. and Speranza, M.G. (2015) 'Exact solutions to the double travelling salesman problem with multiple stacks', *Networks*, Vol. 56, No. 4, pp.229–243.

Potvin, J-Y. (2009) 'State-of-the art review-evolutionary algorithms for vehicle routing', *INFORMS Journal on Computing*, Vol. 21, No. 4, pp.518–548 .

Silvestrin, P.V. and Ritt, M. (2017) 'An iterated tabu search for the multi-compartment vehicle routing problem', *Computers & Operations Research*, May, Vol. 81, pp.192–202.

Sun, H.F., Dong, K.T., Zhang, Q. et al. (2015) 'An improved particle swarm algorithm for heterogeneous fleet vehicle routing problem with two-dimensional loading constraints', *International Conference on Materials Engineering and Information Technology Applications*.

Toth, P. and Vigo, D. (2016) 'Vehicle routing problem', *Encyclopedia of GIS*, Vol. 49, No. 2, pp.799–805.

Wang, W.R. and Wu, Y.H. (2014) 'Modeling and solving vehicle routing problem with changing cost', *Computer Integrated Manufacturing Systems*, Vol. 20, No. 4, pp.979–987.

Wang, K., Lan, S. and Zhao, Y. (2017) 'A genetic-algorithm-based approach to the two-echelon capacitated vehicle routing problem with stochastic demands in logistics service', *Journal of the Operational Research Society*, pp.1–13.

Wei, L., Zhang, Z., Zhang, D. et al. (2015) 'A variable neighborhood search for the capacitated vehicle routing problem with two-dimensional loading constraints', *European Journal of Operational Research*, Vol. 243, No. 3, pp.798–814.

Zhang, D., Cai, S., Ye, F., Si, Y.W. and Nguyen, T.T. (2017) 'A hybrid algorithm for a vehicle routing problem with realistic constraints', *Information Sciences*, Vol. 394, No. C, pp.167–182.