ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)

https://doi.org/10.17559/TV-20200809211109
Original scientific paper

Low Carbon Logistics Optimization for Multi-depot CVRP with Backhauls - Model and Solution

Xiaoning ZHU*, Ziqian ZHAO, Rui YAN

Abstract: CVRP (Capacitated Vehicle Routing Problems) is the integrated optimization of VRP and Bin Packing Problem (BPP), which has far-reaching practical significance, because only by taking both loading and routing into consideration can we make sure the delivery route is the most economic and the items are completely and reasonably loaded into the vehicles. In this paper, the CVRP with backhauls from multiple depots is addressed from the low carbon perspective. The problem calls for the minimization of the carbon emissions of a fleet of vehicles needed for the delivery of the items demanded by the clients. The overall problem, denoted as 2L-MDCVRPB, is NP-hard and it is very difficult to get a good performance solution in practice. We propose a quantum-behaved particle swarm optimization (QPSO) and exploration heuristic local search algorithm (EHLSA) in order to solve this model. In addition, three groups of computational experiments based on well-known benchmark instances are carried out to test the efficiency and effectiveness of the proposed model and algorithm, thereby demonstrating that the proposed method takes a short computing time to generate high quality solutions. For some instances, our algorithm can obtain new better solutions.

Keywords: capacitated vehicle routing problem; hybrid algorithm; low carbon

1 INTRODUCTION

It is an important, typical optimization problem of Vehicle Routing Problem (VRP), in which a fleet of vehicles will deliver the goods demanded by the customers at the minimized total carbon emissions. Considering the attributes of the fleet, traditional VRP has developed several variants, including CVRP (Capacitated Vehicle Routing Problems) which is the combination and optimization of VRP and Bin Packing Problem (BPP), HVRP (heterogeneous VRP) where customers are served by different types of vehicles [1-4]. Both VRP and bin BPP are important, typical combination optimization problems, and are frequently and independently studied in the past few decades. To resolve the VRP and BPP, numerous methodologies based on operations research and mathematical programming techniques or heuristic-based algorithms have been proposed in extant literature [5-7]. It mainly includes particle swarm optimization (PSO) algorithm, ant colony system (ACS) algorithm, tabu search (TS) algorithm and genetic algorithm (GA).

PSO algorithm is an intelligent optimization algorithm, proposed by Kennedy in 1995, which imitates the behavior of birds [8]. The algorithm initializes a group of random particles and uses the iterative method to make each particle find its own solution. The best position is close to the best particle in the group, so as to search the optimal solution. Ai and Kachitvichyanukul [9] proposed a PSO algorithm with multi-social structure, which is the first time to use PSO algorithm to solve vehicle routing problem with simultaneous delivery and pickup. Zhang *et al.*[10] also proposed a modified PSO algorithm, which used (m+1) dimensional vectors to represent a particle for the problem with m customers. In decoding, the particle is transformed into vehicle assignment matrix by scanning algorithm, and then the customer priority matrix of the same vehicle service is determined. The construction of vehicle path is based on these two matrices. This method helps to reduce the number of vehicles used. Goksal et al. [11] mentioned a hybrid algorithm based on PSO and VND for VRP. Although the calculation speed of PSO is fast, it is easy to precocious. In order to solve VRP, Montane and Galvao [12] proposed a TS algorithm with neighborhood structure composed of relocation, interchange, cross operation between paths and 2-opt operation in path. At the same time, a set of examples for solving VRP was given. Vural mentioned two GA algorithms, which are the first time to use GA to solve VRP [13]. Wade and Reimann used ACS to solve the VRP with backhauls (VRPB) [14, 15]. Recently, quantum-behaved particle swarm optimization (QPSO) is improved on the basis of the PSO. It not only inherits the advantages of the example algorithm, but also has much better global search performance than particle algorithm and avoids premature convergence to the local optimal solution [16].

In addition, local search algorithm is also widely used in VRP. This algorithm is usually combined with other algorithms to solve VRP problems. Bianchessi and Righini [17] proposed to use variable neighborhood descent (VND) algorithm to solve VRP. They also combined TS algorithm with VND algorithm VRP. In the process of constructing neighborhood structure, both node exchange based and edge swapping based operations are adopted. Crispimand Brandão [18] explored a hybrid algorithm based on TS and VND to solve VRP. This is the first time that modern heuristic algorithm is used to solve VRP problem. Ropkeand Pisinger [19] represented the variants of vehicle routing problem with backhaul in a unified model, and used a large neighborhood search algorithm to solve them.

In recent years, some attention has been devoted to their combined optimization, called two-dimensional Capacitated Vehicle Routing Problem (2L-CVRP). In the 2L-CVRP clients demand sets of rectangular weighted items, while vehicles have a weight capacity and a rectangular two-dimensional loading surface. The aim is to load the items into the vehicles and deliver them to the clients, through a road network, with minimum total cost. In transportation it is often necessary to handle rectangular-shaped items that cannot be stacked one on the top of the other because of their fragility, weight or large dimensions such as household appliances, delicate pieces of furniture, etc. Since Iori and Vigo proposed the 2L-CVRP, it has been explored extensively and solved by many heuristic algorithms [20-29]. For larger scale problems, Gendreau et al. [21] proposed a Tabu Search algorithm, in which the loading component of the problem was solved through heuristics, lower bounds and a truncated branch-and-bound procedure. Zachariadis et al. [22] proposed a metaheuristic algorithm which incorporated the rationale of Tabu Search and Guided Local Search based on paper, employing a guiding mechanism which could drastically diversify the search conducted by trying to eliminate low-quality features from the final solution to control the objective of the problem. For a population-based algorithm, the Ant Colony Optimization (ACO) was employed by Fuellerer, Doerner, Hartl, and Iori, and the performance was proven to be quite satisfactory [23]. The Greedy Randomized Adaptive Search Procedure combined with Evolutionary Local Search (GRASP×ELS) algorithm was proposed by Duhamel, Lacomme, Quilliot, and Toussaint, and this algorithm outperformed all previous methods and obtained new better solutions for several instances. However, only the Unrestricted 2L-CVRP problems were solved [24]. Recently, Zachariadis, Tarantilis, and Kiranoudis proposed an innovative compact meta-heuristic, named Promise Routing-Memory Packing (PRMP), which obtained excellent performance and improved Best Known Solutions for many instances [25]. Wei et al. [26] proposed a variable neighborhood search to address the routing aspect, and adopted a skyline heuristic to examine the loading constraints. The effectiveness of their approach was verified through experiments on widely used benchmark instances involving two distinct versions of loading constraints (unrestricted and sequential 2L-CVRP problem). According to the combination of the loading constraints, the 2L-CVRP mainly includes four variants, namely, 2L-Sequential Oriented Loading (2|SO|L), 2L-Sequential non-oriented (Rotated) Loading (2|SR|L), 2L-Unrestricted Oriented Loading (2|UO|L), and 2L-Unrestricted non-oriented (Rotated) Loading (2|UR|L) [22]. As far as we know, only Fuellerer et al. have studied all four variants [23].

In our paper, the 2L-CVRP is extended into a multi-depot 2L-CVRP with backhauls (2L-MDCVRPB). Some of the depots both deliver and pickup all items, while some offer either delivery or pickup service. Hence, two situations were discussed: S1, all depots both deliver and pickup items; S2, all the items for each customer are delivered or picked up at only one depot. The items in this model need to satisfy the sequential constraint, which belongs to the 2|SR|L.

The remainder of this paper is organized as follows. The model development in the next section introduces the formulation notations, hypotheses and complete mathematical model of 2L-MDCVRPB. Section 3 introduces the QPSO algorithm for the solution of the MDCVRPB, and EHLSA algorithm is proposed to present the solution of the 2L loading problem. Section 4 introduces three groups of experiments based on benchmark instances including 2L-CVRP, 2L-CVRPB and 2L-MDCVRPB instances. In the last section, the key contributions of this research are discussed along with the related application suggestions.

2 PROBLEM DESCRIPTION

2.1 Definition

The 2L-MDCVRPB problem can be illustrated through a complete undirected graph $G = (V, E)$, where $V = A \cup B \cup D$ is a set of n linehaul customer vertices ($A = \{1, \dots, n\}$), m backhaul customer vertices ($B = \{n + 1, ..., n + m\}$) and d depot vertices ($D = \{n + m + 1, ..., n + m + d\}$), $E = \{(i, j) | i, j \in V; i \neq j\}$ is the set edges between the customer vertices and depot vertices. For simplicity, the set of all customers vertices V/D is denoted as V_c. The length of edge ij equals the distance c_{ij} between nodes i and j. The distance is a positive and has no directionality: $c_{ij} > 0$, $c_{ij} = c_{ji}$ and $c_{ii} = 0$.

There are $m_i > 0$ items to be delivered or picked up for the customer i ($i \in V_c$), and no demand from the depot, i.e. $m_i = 0$, $i \in D$. The weight, length and width of item k for customer i are denoted as q_{ik}, l_{ik} and w_{ik} ($1 \leq k \leq m_i$, $i \in V_c$), respectively. Therefore, we can obtain the total weight of all items $q_i = \sum_{k=1}^{m_i} q_{ik}$ and all items' area $s_i = \sum_{k=1}^{m_i} l_{ik} \cdot w_{ik}$ for customer i.

Each depot l has a fleet of vehicles P_l $(l \in D)$ to provide the distribution services. The vehicle p in depot l is denoted as $p \in P_l$. For each vehicle, the carbon emissions factor per mileage is denoted as ρ; the maximum capacity is denoted as Q; the loading surface is denoted as $S = L \cdot W$, with L being the length and W being the width. With indices $x \in \{1, \dots, L\}$ and $y \in \{1, \dots, W\}$, $\left(x_{ik}^{pl}, y_{ik}^{pl}\right)$ represents the site of the bottom-left corner of item k to (from) customer i in vehicle p belongs to depot l.

Drawing on Dominguez et al. [30], the feasible routes can be divided into linehaul routes and backhaul routes. Hence, a feasible route R^{pl} can be viewed as the subset of clients $\left(R^{pl} \subseteq V_c\right)$ visited by the vehicle p of depot l. Under the clustered visits assumptions, the route R^{pl} was divided into two independent subsets, including linehaul R_L^{pl} and backhaul R_B^{pl}, i.e., $R^{pl} = R_L^{pl} \cup R_B^{pl}$ and $R_L^{pl} \cap R_B^{pl} = \phi$.

On this basis, three decision variables were put forward. The first decision variable was defined as $z_{ij}^{pl} \in \{0,1\}$, $i, j \in V$, $p \in P_l$, $l \in D$. If $z_{ij}^{pl} = 1$, then vehicle p at depot l travels starting at node i and ending at node j; if $z_{ij}^{pl} = 0$, then vehicle p at depot l travels starting at node j and ending at node i. Here, it is assumed that $z_{ii}^{pl} = 0$, $i \in V$, $p \in P_l$ and $l \in D$. The second decision variable was defined as $b_i^{pl} \in \{0,1\}$, $i \in V_c$, $p \in P_l$, $l \in D$. If $b_i^{pl} = 1$, then the vehicle p at depot l can visit customer i. Obviously, $\sum_{j \in V} z_{ij}^{pl} = b_i^{pl}$, $i \in V_c$, $p \in P_l$, $l \in D$. The third decision variable θ_{ik} was defined to show item k belonging to client i has been rotated $(\theta_{ik} = 1)$ or has not been rotated $(\theta_{ik} = 0)$.

2.2 Hypotheses

The 2L-MDCVRPB calls for a set of routes and the feasible packing solutions for all vehicles to operate with the minimum total carbon emissions under the following hypotheses:

(1) Each route has the identified starting and ending depot, which can be the original depot or any other depot. In S1, every route must have the same starting and ending depot.

(2) To avoid split deliveries, each single vehicle can only serve to one customer.

(3) Each route should visit at least one linehaul customer.

(4) Each customer can only be visited once, i.e. all items of every customer must be placed in a single vehicle.

(5) Each vehicle should not be overloaded and the items loaded on it should not surpass the loading surface.

(6) All items belonging to every customer should totally be loaded on the vehicle's loading surface.

(7) When loading and unloading all items, they should be carried out from the rear of the vehicle, and only straight movement (one per item) is allowed.

(8) Each item has a fixed direction and can be rotated up to 90° at most.

(9) The sides of each item being loaded must be parallel to those of the loading surface.

(10) Stacking or stowing items is not allowed in the vehicle.

(11) The items cannot be rearranged at the site of any customer.

2.3 Modelling

This paper presents the 2L-MDCVRPB model which includes problem definition, hypothesis and model construction. The graph theory description of the model is given in the problem definition, and some assumptions are put forward in order to make the model boundary clearer. Finally, the integer programming model of the problem is given and the meaning of each formula in the model is explained. The 2L-MDCVRPB problem can be formulated as:

$$\min \sum_{l\in D}\sum_{p\in P_l}\sum_{j\in V}\sum_{i\in V} \rho\cdot c_{ij}\cdot z_{ij}^{pl} \tag{1}$$

s.t.

$$\sum_{j\in A} z_{lj}^{pl} = \sum_{i\in V_c} z_{il}^{pl}\ ,\ \forall p\in P_l\ ,\ \forall l\in D \tag{2}$$

$$z_{lj}^{pl} = 0\ ,\ \forall j\in B\ ,\ \forall p\in P_l\ ,\ \forall l\in D \tag{3}$$

$$\sum_{l\in D}\sum_{p\in P_d}\sum_{i\in V} z_{ij}^{pl} = 1\ ,\ \forall j\in V_c \tag{4}$$

$$\sum_{i\in V} z_{iu}^{pl} = \sum_{j\in V} z_{ju}^{pl}\ ,\ \forall u\in V_c\ ,\ \forall p\in P_l\ ,\ \forall l\in D \tag{5}$$

$$\sum_{i\in B, j\in A} z_{ij}^{pl} = 0\ ,\ \forall p\in P_l\ ,\ \forall l\in D \tag{6}$$

$$\sum_{i\in A, j\in B} z_{ij}^{pl} \le 1\ ,\ \forall p\in P_l\ ,\ \forall l\in D \tag{7}$$

$$\sum_{p\in P_l}\sum_{j\in V_c} z_{lj}^{pl} \le \left|P_l\right|\ ,\ \forall l\in D \tag{8}$$

$$\sum_{i\in A, j\in V} q_i\cdot z_{ij}^{pl} \le Q\ ,\ \forall p\in P_l\ ,\ \forall l\in D \tag{9}$$

$$\sum_{i\in A, j\in V} s_i\cdot z_{ij}^{pl} \le S\ ,\ \forall p\in P_l\ ,\ \forall l\in D \tag{10}$$

$$\sum_{i\in B, j\in V} q_i\cdot z_{ij}^{pl} \le Q\ ,\ \forall p\in P_l\ ,\ \forall l\in D \tag{11}$$

$$\sum_{i\in B, j\in V} s_i\cdot z_{ij}^{pl} \le S\ ,\ \forall p\in P_l\ ,\ \forall l\in D \tag{12}$$

$$\sum_{j\in V} z_{ij}^{pl} = b_i^{pl}\ ,\ \forall i\in V_c\ ,\ \forall p\in P_l\ ,\ \forall l\in D \tag{13}$$

$$\begin{gathered} 0\le x_{ik}^{pl} \le \left(L-l_{ik}\right)\left(1-\theta_{ik}\right)+\left(L-w_{ik}\right)\theta_{ik}, \\ \forall k\in\left\{1,\dots,m_i\right\}, \forall i\in V_c \end{gathered} \tag{14}$$

$$\begin{gathered} 0\le y_{ik}^{pl} \le \left(W-w_{ik}\right)\left(1-\theta_{ik}\right)+\left(W-l_{ik}\right)\theta_{ik}, \\ \forall k\in\left\{1,\dots,m_i\right\}, \forall i\in V_c \end{gathered} \tag{15}$$

$$\begin{aligned} & ll_{ii'kk'}^{pl} = \max\left\{\begin{matrix}\left(x_{ik}^{pl}+l_{ik}\right)\cdot\left(1-\theta_{ik}\right),\left(x_{ik}^{pl}+w_{ik}\right)\cdot\theta_{ik}, \\ \left(x_{i'k'}^{pl}+l_{i'k'}\right)\cdot\left(1-\theta_{i'k'}\right),\left(x_{i'k'}^{pl}+w_{i'k'}\right)\cdot\theta_{i'k'}\end{matrix}\right\} \\ & -\min\left(b_i^{pl}\cdot x_{ik}^{pl}, b_{i'}^{pl}\cdot x_{i'k'}^{pl}\right)- \\ & -\left(l_{ik}\cdot\left(1-\theta_{ik}\right)+w_{ik}\cdot\theta_{ik}+l_{i'k'}\cdot\left(1-\theta_{i'k'}\right)+w_{i'k'}\cdot\theta_{i'k'}\right), \\ & \forall k\in\left\{1,\dots,m_i\right\}, \forall k'\in\left\{1,\dots,m_{i'}\right\}, \forall i,i'\in V_c, \forall p\in P_l, \\ & \forall l\in D \end{aligned} \tag{16}$$

$$\begin{aligned} & ww_{ii'kk'}^{pl} = \max\left\{\begin{matrix}\left(y_{ik}^{pl}+w_{ik}\right)\cdot\left(1-\theta_{ik}\right),\left(y_{ik}^{pl}+l_{ik}\right)\cdot\theta_{ik}, \\ \left(y_{i'k'}^{pl}+w_{i'k'}\right)\cdot\left(1-\theta_{i'k'}\right),\left(y_{i'k'}^{pl}+l_{i'k'}\right)\cdot\theta_{i'k'}\end{matrix}\right\}- \\ & -\min\left\{b_i^{pl}\cdot y_{ik}^{pl}, b_{i'}^{pl}\cdot y_{i'k'}^{pl}\right\}- \\ & -\left(w_{ik}\cdot\left(1-\theta_{ik}\right)+l_{ik}\cdot\theta_{ik}+w_{i'k'}\cdot\left(1-\theta_{i'k'}\right)+l_{i'k'}\cdot\theta_{i'k'}\right), \\ & \forall k\in\left\{1,\dots,m_i\right\}, \forall k'\in\left\{1,\dots,m_{i'}\right\}, \forall i,i'\in V_c, \forall p\in P_l, \\ & \forall l\in D \end{aligned} \tag{17}$$

$$\begin{aligned} & \max\left\{b_i^{pl}\cdot b_{i'}^{pl}\cdot ll_{ii'kk'}^{pl}, b_i^{pl}\cdot b_{i'}^{pl}\cdot ww_{ii'kk'}^{pl}\right\}\ge 0 \\ & \forall k\in\left\{1,\dots,m_i\right\}, \forall k'\in\left\{1,\dots,m_{i'}\right\}, \forall i,i'\in A, \\ & \forall i,i'\in B, \forall p\in P_l, \forall l\in D \end{aligned} \tag{18}$$

$$\begin{aligned} & \max\left\{z_{ii'}^{pl}\cdot ww_{ii'kk'}^{pl}, z_{ii'}^{pl}\cdot\left(x_{ik}^{pl}-x_{i'k'}^{pl}\right)\right\}\ge 0, \\ & \forall k\in\left\{1,\dots,m_i\right\}, \forall k'\in\left\{1,\dots,m_{i'}\right\}, \\ & \forall i,i'\in A, i\ne i', \forall p\in P_l, \forall l\in D \end{aligned} \tag{19}$$

$$\begin{aligned} & \max\left\{z_{i'i}^{pl}\cdot ww_{ii'kk'}^{pl}, z_{ii'}^{pl}\cdot\left(x_{ik}^{pl}-x_{i'k'}^{pl}\right)\right\}\ge 0, \\ & \forall k\in\left\{1,\dots,m_i\right\}, \forall k'\in\left\{1,\dots,m_{i'}\right\}, \\ & \forall i,i'\in B, i\ne i', \forall p\in P_l, \forall l\in D \end{aligned} \tag{20}$$

$$z_{ij}^{pl}\in\left\{0,1\right\}, \forall i,j\in V, \forall p\in P_l, \forall l\in D \tag{21}$$

$$\begin{gathered} x_{ik}^{pl}\in\left\{1,\dots,L\right\}, y_{ik}^{pl}\in\left\{1,\dots,W\right\}, \forall k\in\left\{1,\dots,m_i\right\}, \\ \forall i\in V_c, \forall p\in P_l, \forall l\in D \end{gathered} \tag{22}$$

The objective function is (1), which means to minimize the total carbon emissions. Eq. (2) ensures that each vehicle leaves and returns to the same depot. Eq. (3) ensures that a linehaul customer is placed at the start of the route. Eq. (4) and Eq. (5) confirm that every customer is only visited once, and a vehicle leaves from the customer after the visit. Eq. (6) and Eq. (7) ensure that all linehaul customers are served before backhaul customers. Eq. (8) sets the upper limit of the number of vehicles. Eq. (9) and Eq. (10) ensure that all items loaded on every vehicle are within the capacity or loading surface of that vehicle. Eq. (11) and Eq. (12) are the equivalent restrictions with respect to backhaul locations. Eq. (13) is the calculation function of z_{ij}^{pl} and b_i^{pl} . Eq. (14) and Eq. (15) ensure that the items are loaded within the loading surface. Eq. (14) ensures that the horizontal ordinate of the item plus the length (with no rotation) or width (with rotation) of the item does not exceed the length of the carriage, and the Eq. (15) indicates that the ordinate of the item plus the width (with no rotation) or length (with rotation) of the item does not exceed the width of the carriage. Eq. (16) and Eq. (17) are the calculation functions of *ll* ans *ww*, respectively. Eq. (18) ensures that the items are not stacked on the vehicle surface, because no linehaul customer is visited after backhaul customers. Eq. (19) and Eq. (20) ensure the first in, last out principle. Eq. (21) and Eq. (22) specify the ranges of the decision variables.

3 DESIGN OF HYBRID HEURISTIC ALGORITHM

The proposed QPSO (Quantum-behaved Particle Swarm Optimization) and EHLSA (Exploration Heuristic Local Search Algorithm) both are hybrid algorithms for solving the 2L-MDCVRPB, in which QPSO algorithm is for routing vehicles and EHLSA algorithm is for loading all the items of the clients into the vehicle. Section 3.1 proposes the QPSO algorithm for the solution of the MDCVRPB, and it improves the solution in new decoding method and initial solution generation. In section 3.2, EHLSA algorithm is proposed to present the solution of the 2L loading problem.

3.1 Quantum-Behaved Particle Swarm Optimization

In the typical PSO algorithm, particle swarm is composed of particles, in which each particle is an independent individual. In the multidimensional search space at a certain speed, each particle's position is considered to be a better position of each problem particle. When the first iteration is carried out, the position of a particle is randomly generated and the particle's velocity is initialized to zero. In other iterations, the particle's position and speed are updated according to the following formulas:

$$V_{i,j}(t+1) = w \cdot V_{i,j}(t) + c_1 \cdot r_{1,i}(t) \cdot \left[P_{i,j}(t) - X_{i,j}(t)\right] + \\ + c_2 \cdot r_{2,i}(t) \cdot \left[P_{g,j}(t) - X_{i,j}(t)\right] \tag{23}$$

$$X_{i,j}(t+1) = X_{i,j}(t) + V_{i,j}(t+1) \tag{24}$$

After studying the PSO algorithm convergence, Liu et al. mentioned that each particle must converge to the absorber C_i of the particle, where the C_i was expressed as follows [31]:

$$C_{i,j}(t) = \frac{\left(c_1 \cdot r_1 \cdot P_{i,j}(t) + c_2 \cdot r_2 \cdot P_{g,j}(t)\right)}{\left(c_1 \cdot r_1 + c_2 \cdot r_2\right)} \\ j = 1,2,\cdots,d, \;\; c_1, c_2 \sim U(0,1) \tag{25}$$

$$C_{i,j}(t) = \alpha \cdot P_{i,j}(t) + (1-\alpha) \cdot P_{g,j}(t), \\ \alpha = \frac{(c_1 \cdot r_1)}{\left(c_1 \cdot r_1 + c_2 \cdot r_2\right)}, \;\; \alpha \sim U(0,1) \tag{26}$$

where P_i is the best particle between individuals, P_g is the global best particle. According to the calculation formula of α, it shows that the role of local attractors is to attract particles i randomly, and it is situated in a super rectangle with P_i and P_g at the two ends of the diagonal.

Many scholars have studied the convergence characteristics of PSO. On the basis of the basic convergence characteristics of Particle Swarm Optimization, Jin et al. [2] mentioned a Quantum Particle Swarm Optimization algorithm (QPSO). Since the evolution equation does not require a velocity vector, QPSO algorithm is improved for search strategies with fewer parameters.

According to Tang et al., in the quantum space-time framework, a wave function $\psi(\vec{x},t)$ represented the quantum state of the particle [27]. After that a global point was added into QPSO, which also represented Mainstream Thought or mean best position of the population [28]. The global point is the average of the *Pbest* positions, which is denoted as *mbest*. That is:

$$mbest(t) = \left(mbest_1(t), mbest_2(t), \dots mbest_d(t)\right) \\ = \left(\frac{1}{n}\sum_{i=1}^{n} P_{i,1}(t), \frac{1}{n}\sum_{i=1}^{n} P_{i,2}(t), \dots, \frac{1}{n}\sum_{i=1}^{n} P_{i,d}(t)\right) \tag{27}$$

The following formula can calculate the new position of a particle:

$$X_{i,j}(t+1) = C_{i,j}(t) \pm \beta \cdot \left|mbest_j - X_{i,j}(t)\right| \cdot \ln(1/u) \\ u = rand(0,1) \tag{28}$$

where contraction-expansion coefficient β is used to control the convergence rate of the algorithm.

There are four steps for the QPSO algorithm running. The first step is to initialize the position of particles randomly, and the P_i is the initial position of particle *i*. In addition, the initial velocity of the particles is all zero. The second step is to get the value of the particles objective function through the decoding method. After that we can obtain the value of P_i and P_g of every particle using its objective function. P_g is set to the best P_i of all population. The last is to apply (26) and (28) to calculate the velocity and position of each particle respectively.

(1) Decoding method

The existing QPSO model cannot solve the problem of 2L-MDCVRPB proposed in this paper. Therefore, this paper makes the following changes: (1) making sure the QPSO can handle the problem of picking up items with backhauls through 3D coding; (2) combining QPSO with EHLSA to solve the two dimensional loading problem. The final QPSO-EHLSA algorithm can solve the 2L-MDCVRPB problem. Assuming that there are multiple depots in the model, we design a new particle coding method to represent the relationship between the T depots, the v vehicles and the n client. This method uses a $3\times n$ matrix, where the first row X_d shows the depots, the second row X_v represents the vehicles, and the third row X_t indicates the vehicles distances. For example, supposing there are two depots and ten customers. Depot1 has two vehicles and depot2 has three vehicles. Fig. 1 shows the results as follows.

Client	1	2	3	4	5	6	7	8	9	10
X_d	1	1	2	1	2	2	2	1	2	2
X_v	1	1	3	2	4	4	4	2	5	5
X_t	0.3	1.2	0.6	2.3	4.8	1.1	2.9	3.9	2.8	1.7

Figure 1 Decoding example

X_t uses its value to adjust the routes instead of the distance, so before that, all clients who can be cut off by each vehicle must be determined, and then sorted according to the value of X_t to determine every vehicle's route. This method can ensure that the client can only be severed once, thereby reducing the adjustment process of the feasible scheme. Fig. 2 shows the adjustment process.

Client	1	2	3	4	5	6	7	8	9	10
X_d	1	1	2	1	2	2	2	1	2	2
X_v	1	1	3	2	4	4	4	2	5	5
X_t	1	2	1	1	3	1	2	2	2	1

⇩

Depot1	Vechile1: 1 —> 2
	Vechile2: 4 —> 8
Depot2	Vechile3: 3
	Vechile4: 6 —> 7 —> 5
	Vechile5: 10 —> 9

Figure 2 Change process of location and route

(2) Local search heuristics

We will introduce the local search heuristics in this subsection. The 2-opt heuristic and inner-tour swap was an improved method, provided by Norouzi et al. [29].

The 2-opt heuristic is based on the idea of exchange, transforming one path into another, which is suitable for two trips; first, choosing a spot in each trip so that the two trips are divided into two parts. The next step is to connect the first part of the first trip to the second part of second trip. In the same way, the second part of the first trip add into the first part of second trip.

Inner-tour swap is different from 2-opt heuristic. It can be applied in a single trip. Above all, there are two clients assigned to the one trip, and then their sequences are replaced. We define two counts, including count 1 and count 2 in order to use this local search set. Count 1 is from 1 to trip length-1 that the length is the same as the number of clients in the selected tour, and count 2 is from count 1 to trip length. Next is to exchange the sequence of count 1st client and count 2nd client. If the result of the exchange reduces the cost, the change is accepted and will continue. Otherwise, the change is rejected and another pair of customers is selected for exchange through the count 1 and count 2.

3.2 Exploration Heuristic Search Algorithm

On the basis of QPSO algorithm in Section 3.1, the tours can be obtained. In addition, the clients visited by the vehicles and the vehicles in the routes can be determined. The vehicles with relevant constraints need to load all items required by the clients. Actually, there are two aspects considered in the packing algorithm. One is to identify the next loading project, the other is to identify the possible loading location. Therefore, the two methods will be provided.

(1) Determine the items order

Suppose that the order of all clients' visits in a certain tour is fixed. If a client's items are being unloaded, other items in this vehicle cannot be moved. As a result, OV presents that items are sorted in descending visit order in the vehicle. After that the items order of each customer is fixed, while the order of goods belonging to the same customer is not fixed. It should be noted that the unrestricted model does not consider visit order to sort items.

There are two orders, $O1$ and $O2$, set to identify the eventual orders of one client's items in the sequential model or all clients' items in the unrestricted model. $O1$ and $O2$ are arranged in descending order according to the area $l\cdot w$ and length l, respectively. If there are the same bottom areas in the two items, $O2$ is prior to $O1$. Otherwise, $O1$ is prior to $O2$. Through the rule above, the loading sequence of items can be determined as follows:

Depot 1 - Vehicle 1: I_{12}-I_{11}-I_{23}-I_{22}-I_{21},
Vehicle 2: I_{42}-I_{41}-I_{43}-I_{82}-I_{81}-I_{83}-I_{84},
Depot 2 - Vehicle 3: I_{32}-I_{31}-I_{33}-I_{34},
Vehicle 4: I_{61}-I_{62}-I_{63}-I_{72}-I_{71}-I_{73}-I_{51}-I_{52},
Vehicle 5: I_{102}-I_{101}-I_{103}-I_{92}-I_{91}.

(2) Determine the feasible loading position

For the sorting of the loading locations, we propose a new heuristic rule. The O (0, 0) in Fig. 3 at the front-left corner of the vehicle is the first selected item.

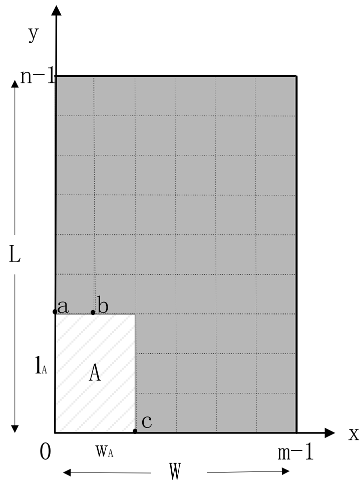


Figure 3 Feasible loading position

It should be noted that the coordinates at the front left represent the location of the item. For example, item A is located at point *O* with the length of 3 and the width of 2. The space occupied is a square with four points, which is point (0, 0), (0, 3), (2, 0) and (2, 3). After following the rule that items should be close to the edge of the vehicle or other items, new locations list *involves* $a\left(0,W,L-l_A\right)$, $b\left(1,W-1,L-l_A\right)$, $c\left(2,W,L\right)$ when *A* is loaded, as shown in Fig. 3.

After that, in order to reduce the waste of the space, the most ideal location of all current items loading position should be selected in all feasible positions. We list all rules about sorting the feasible loading positions.

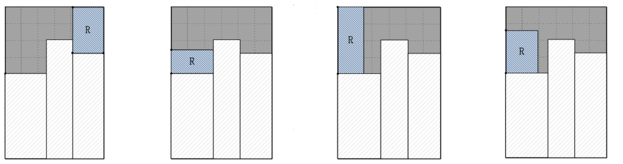
(1) when the fitness is 1, it means that the length and width of the item are just suitable for the feasible loading place of the vehicle provided, as shown in Fig. 4a.

(2) when the fitness is 2, it means that the width of the item is exactly equal to the feasible loading place, but for length, the item is smaller, as shown in Fig. 4b.

(3) when the fitness is 3, it shows that the length of the item is exactly equal to the feasible loading place, but for width, the item is smaller, as shown in Fig. 4c.

(4) when the fitness is 4, it shows both for width and length, the item is smaller, as shown in Fig. 4d.

(5) When the fitness is infinite, both for length and width, the item is greater than the feasible loading position.



(a) fitness is 1 (b) fitness is 2 (c) fitness is 3 (d) fitness is 4

Figure 4 Fitness of item and loading place

4 EXPERIMENTAL VERIFICATION

4.1 Experiments on 2L-CVRP and 2L-CVRPB Instances

Our EHLSA -QPSO algorithm was tested with the 2L-CVRP instance proposed by Iori et al. [20], and the 2L-CVRPB instance introduced by Dominguez et al. [30]. In fact, the 2L-MDCVRPB can be viewed as the extension of the 2L-CVRP and 2L-CVRPB. The 2L-CVRP instance is an especial case of 2L-CVRPB where there are only linehaul customers, and the 2L-CVRPB instance is a special case of 2L-MDCVRPB with a single depot. The experiment was conducted under the loading constraints of Sequential non-oriented (Rotated) Loading (2|SR|L) and Sequential Oriented Loading (2|SO|L).

According to the results on the sequential oriented variant of the 2L-CVRP (2|SO|L), our EHLSA -QPSO algorithm converged to the Best Known Solution (BKS) faster than the contrastive algorithms. The Variable Neighborhood Search (VNS) consumed 3 times the duration, the Ant Colony Optimization (ACO) took 2.2 times, and the Biased-Randomized Large Neighbourhood Search (BR-LNS) took 4.3 times.

Next, the 2L-VRP instances were extended into new instances for the 2L-CVRPB. The extension follows the way Toth and Vigo generated VRPB instances from the classic Euclidean VRP. From the 180 2L-CVRP instances in the original sets, 540 2L-CVRPB instances were created totally, and divided into three groups by the percentage of linehaul customers (50%, 60% and 80%). 60 instances (20 in each group) were selected for the numerical experiment.

Through the experimental results, we can see that our algorithm converged to lots of best solutions for many instances. For the 60 test instances, 53 new best solutions were identified correctly. The mean optimized gap was 1.22% for the first group, 2.20% for the second group, and 3.32% for the third group. In addition, the EHLSA -QPSO algorithm outperformed the BR-LNS, which has been applied to solve the 2L-CVRPB in only one paper.

4.2 Experiments on 2L-MDCVRPB Instances

By adding a depot, the 60 selected 2L-CVRPB instances were converted to two-depot instances (2L-MDCVRPB instances), and used to verify the effect of our algorithm. The second depot was placed at a random point between the first depot and the center of customers. The numerical experiments were divided into two groups, based on whether the items should be delivered to the designated depot. In the first group, a depot was added to the 2L-CVRPB instances; in the second group, the designated depot was selected randomly after adding a depot.

The results show that the total distance of the first group was shorter than that of the second group. This is because some customers are served by the new depot nearby, eliminating the need to travel from/to the single depot. However, when the item must be sent to the designated depot, the customer is served by the designated depot rather than the nearby depot. Therefore, the mean distance in *S*2 was longer than that in *S*1. The mean computing time in *S*2 was also relatively long, due to the need to check the fulfilment of the constraint on the designated depot.

5 CONCLUSIONS

This paper probes deep into the 2L-MDCVRPB and designs an improved hybrid algorithm EHLSA -QPSO to solve the problem. As far as we know, this is the first attempt to tackle the realistic extension of the VRP with two-dimensional capacity. Moreover, item rotation is allowed in vehicle loading, which is a realistic but rarely reported assumption.

After exploring various loading constraints and problem variants, the author concluded that our EHLSA -QPSO algorithm takes a short computing time to generate high quality solutions. It has been proved that particle swarm optimization is a very efficient approach for the 2L-MDCVRPB, and the quantum-behaved can help particle swarm optimization to escape effectively from local optimum, and it can be applied to the similar problem in other areas. For the selected instances, our algorithm can find the BKS with much shorter time than contrastive algorithms. The research findings provide new insights into the 2L-MDCVRPB, and effective tool to solve the 2L-CVRP or 2L-CVRPB.

According to the results of the experiments on 2L-CVRP, 2L-CVRPB and 2L-MDCVRPB instances, the proposed algorithm in this paper has achieved better performance in both the calculation results and the calculation speed. This paper solves a scientific problem extracted from the practice of logistics distribution optimization. The main contribution is to establish a mathematical model to solve this scientific problem.

Therefore, there are not many numerical experiments that have been carried out, which is the limitation of this research. In the future, we will improve and design a new algorithm with higher efficiency to solve this problem with a large number of numerical experiments.

Acknowledgement

The work has been supported by National Natural Science Foundation of China (No. 71602008, No. 71802021), Fundamental Research Funds for the Central Universities (FRF-BD-20-15A), Beijing Natural Science Foundation (No. 9184023), and Beijing Social Science Fund Research Project (No. 16JDGLC032, No. 17JDGLB011, No. 18GLB022).

6 REFERENCES

[1] Brandão, J. (2011). A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. *Computers & Operations Research, 38*(1), 140-151. https://doi.org/10.1016/j.cor.2010.04.008

[2] Jin, X., Liang, Y., Tian, D., & Zhuang, F. (2013). Particle swarm optimization using dimension selection methods. *Applied Mathematics and Computation, 219*(10), 5185-5197. https://doi.org/10.1016/j.amc.2012.11.020

[3] Gendreau, M., Laporte, G., Musaraganyi, C., & Taillard, É. D. (1999). A tabu search heuristic for the heterogeneous fleet vehicle routing problem. *Computers & Operations Research, 26*(12), 1153-1173. https://doi.org/10.1016/S0305-0548(98)00100-2

[4] Golden, B., Assad, A., Levy, L., & Gheysens, F. (1984). The fleet size and mix vehicle routing problem. *Computers & Operations Research, 11*(1), 49-66. https://doi.org/10.1016/0305-0548(84)90007-8

[5] Pan, J., Fu, Z., & Chen, H. W. (2019). Split delivery vehicle routing problem with minimum delivery amounts. *Journal Européen des Systèmes Automatisés, 52*(3), 257-265. *https://doi.org/10.18280/jesa.520306*

[6] Pan, J., Fu, Z., & Chen, H. W. (2019). A tabu search algorithm for the discrete split delivery vechicle routing problem. *Journal Europeen des Systemes Automatises, 52*(1), 97-105. https://doi.org/10.18280/jesa.520113

[7] Prins, C. (2009). Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Engineering Applications of Artificial Intelligence, 22*(6), 916-928. https://doi.org/10.1016/j.engappai.2008.10.006

[8] Kennedy, J. (2011). Particle swarm optimization. *Proc. of 1995 IEEE Int. Conf. Neural Networks, 4*(8), 1942-1948. https://doi.org/10.1007/978-0-387-30164-8_630

[9] Ai, T. J. & Kachitvichyanukul, V. (2009). A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research, 36*(5), 1693-1702. https://doi.org/10.1016/j.cor.2008.04.003

[10] Zhang, N. Z., Sun, G. H., Wu, Y. H., & Geng, F. H. (2009). A modified particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *2009 7th Asian Control Conference, Hong Kong*, 1679-1684.

[11] Goksal, F. P., Karaoglan, I., & Altiparmak, F. (2013). A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery. *Computers & Industrial Engineering, 65*(1), 39-53. https://doi.org/10.1016/j.cie.2012.01.005

[12] Montane, F. A. T. & Galvao, R. D. (2006). A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research, 33*(3), 595-619. https://doi.org/10.1016/j.cor.2004.07.009

[13] Vural, A. V. (2003). A GA based meta-heuristic for capacitated vehicle routing problem with simultaneous pick-up and deliveries (Doctoral dissertation).

[14] Wade, A. & Salhi, S. (2003). An ant system algorithm for the mixed vehicle routing problem with backhauls. *Metaheuristics: Computer Decision-Making*, 699-719. https://doi.org/10.1007/978-1-4757-4137-7_33

[15] Reimann, M. & Ulrich, H. (2006). Comparing backhauling strategies in vehicle routing using ant colony optimization. *Central European Journal of Operations Research, 14*(2), 105-123. https://doi.org/10.1007/s10100-006-0163-8

[16] Sun, J., Feng, B., & Xu, W. (2004). Particle swarm optimization with particles having quantum behavior. *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753),* Portland, OR, USA, 325-331. https://doi.org/10.1109/CEC.2004.1330875

[17] Bianchessi, N. & Righini, G. (2007). Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. *Computers & Operations Research, 34*(2), 578-594. https://doi.org/10.1016/j.cor.2005.03.014

[18] Crispim, J. & Brandão, J. (2005). Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls. *Journal of the Operational Research Society, 56(*11), 1296-1302. https://doi.org/10.1057/palgrave.jors.2601935

[19] Ropke, S. & Pisinger, D. (2006). A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research, 171*(3), 750-775. https://doi.org/10.1016/j.ejor.2004.09.004

[20] Iori, M., Salazar-González, J. J., & Vigo, D. (2007). An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Transportation science, 41*(2), 253-264. https://doi.org/10.1287/trsc.1060.0165

[21] Gendreau, M., Iori, M., Laporte, G., & Martello, S. (2008). A Tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks: An International Journal, 51*(1). https://doi.org/10.1002/net.20192

[22] Zachariadis, E. E., Tarantilis, C. D., & Kiranoudis, C. T. (2009). A guided tabu search for the vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research, 195*(3), 729-743. https://doi.org/10.1016/j.ejor.2007.05.058

[23] Fuellerer, G., Doerner, K. F., Hartl, R. F., & Iori, M. (2009). Ant colony optimization for the two-dimensional loading vehicle routing problem. *Computers & Operations Research, 36*(3), 655-673. https://doi.org/10.1016/j.cor.2007.10.021

[24] Duhamel, C., Lacomme, P., Quilliot, A., & Toussaint, H. (2011). A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem. *Computers & Operations Research, 38*(3), 617-640. https://doi.org/10.1016/j.cor.2010.08.017

[25] Zachariadis, E. E., Tarantilis, C. D., & Kiranoudis, C. T. (2013). Integrated distribution and loading planning via a compact metaheuristic algorithm. *European Journal of Operational Research, 228*(1), 56-71. https://doi.org/10.1016/j.ejor.2013.01.040

[26] Wei, L., Zhang, Z., Zhang, D., & Lim, A. (2015). A variable neighborhood search for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research, 243*(3), 798-814. https://doi.org/10.1016/j.ejor.2014.12.048

[27] Tang, D., Cai, Y., Zhao, J., & Xue, Y. (2014). A quantum-behaved particle swarm optimization with memetic algorithm and memory for continuous non-linear large scale problems. *Information Sciences, 289*, 162-189. https://doi.org/10.1016/j.ins.2014.08.030

[28] Sun, J., Xu, W., & Feng, B. (2004). A global search strategy of quantum-behaved particle swarm optimization. *IEEE Conference on Cybernetics and Intelligent Systems, Singapore*, 111-116. https://doi.org/10.1109/ICCIS.2004.1460396

[29] Norouzi, N., Sadegh-Amalnick, M., & Alinaghiyan, M. (2015). Evaluating of the particle swarm optimization in a periodic vehicle routing problem. *Measurement, 62*, 162-169. https://doi.org/10.1016/j.measurement.2014.10.024

[30] Dominguez, O., Guimarans, D., Juan, A. A., & Ignacio, D. L. N. (2016). A biased-randomised large neighbourhood search for the two-dimensional vehicle routing problem with backhauls. *European Journal of Operational Research, 255*(2), 442-462. https://doi.org/10.1016/j.ejor.2016.05.002

[31] Liu, J., Sun, J., & Xu, W. (2006). Quantum-behaved particle swarm optimization with adaptive mutation operator. *International Conference on Natural Computation*, Xi'an, 959-967. https://doi.org/10.1007/11881070_126

Contact information:

Xiaoning ZHU, associate professor
(Corresponding author)
Donlinks School of Economics and Management,
University of Science & Technology Beijing,
Beijing 100083, China
E-mail: zhuxiaoning@ustb.edu.cn

Ziqian ZHAO, master student
Donlinks School of Economics and Management,
University of Science & Technology Beijing,
Beijing 100083, China
E-mail: 15222706863@163.com

Rui YAN, associate professor
Donlinks School of Economics and Management,
University of Science & Technology Beijing,
Beijing 100083, China
School of Management and Economics,
Beijing Institute of Technology,
Beijing 100081, China
E-mail: yanrui@ustb.edu.cn