Drug delivery route optimization with a capacity based on the ALNS algorithm

Chuyao Ji

School of Applied Mathematics, Stony Brook University, New York, 11794, USA

jichuyao20030518@163.com

Abstract. This paper established a vehicle routing problem (VRP) model with capacity limitation to solve the drug delivery route optimisation problem using an adaptive large-scale neighbourhood search algorithm. The research aims to match orders to riders based on information such as merchant location, rider location, customer location, order remaining time, and rider load to minimise the total delivery distance while ensuring that each order is delivered on time. The model includes multiple objective functions and constraints, such as travel cost and performance cost, as well as time Windows and load capacity limits. By designing three kinds of damage operators (random damage, worst damage and correlation damage) and three kinds of repair operators (greedy repair, regret repair and random repair), and applying the acceptance criteria of the simulated annealing algorithm, the solution process is optimised. The experimental results show that the ALNS algorithm can effectively solve the optimal path scheme after several iterations, significantly reduce the total distribution distance and time, and improve distribution efficiency. The results of this study have important reference value to the actual drug delivery system, which helps improve the rate of patient treatment and the timeliness of drug delivery. In this paper, detailed experiments and data analysis verify the proposed algorithm's effectiveness and feasibility.

Keywords: ALNS, VRP problem, pharmacy distribution, NP-hard problem.

1 Introduction

Dantig et al [1]. proposed the VRP problem for the first time in 1959. They studied the mathematical model and algorithm used to solve the problem, and began to apply it to the enterprise practice. On the basis of Dantig, Clark et al [2]. improved the solution quality by proposing a saving algorithm. With the publication of these two articles, more and more scholars began to study different kinds of VRP problems, hoping to find more solving models and algorithms. Lenstra et al [3]. analyzed the complexity of VRP problems and pointed out that all VRP problems are NP-hard problems. When solving the Vehicle Routing Problem with Time Windows (VRPTW), in addition to satisfying the constraints of the basic VRP problem, it is necessary to consider the constraints of different time Windows of different customers. According to different time window constraints, it is necessary to construct the optimization objective function related to time window in the optimization process, which greatly increases the complexity of problem solving. Gayialis et al [4]. used the Large Neighborhood Search (LNS) algorithm to solve the VRPTW problem with the minimum number of vehicles and the total driving distance as the objective function. Corstiens et al [5]. added multivariate statistical

CONF-CIAP 2025, January 17-23, Eskisehir, Turkey

DOI 10.4108/eai.17-1-2025.2355342

analysis to the large-scale neighborhood search algorithm to solve the vehicle routing problem with time window. The results of this study have important reference value to the actual drug delivery system, which is helpful to improve the rate of patient treatment and the timeliness of drug delivery.

2 Mathematical Model and Algorithm

2.1 The Problem Described in the Mathematical Model

Problem Description. After the customer places the order, the order and the rider are matched according to the merchant location (i.e. pick-up point), rider location, customer location (i.e. delivery point), the remaining time of the customer order, and the load of the rider, etc., and the total delivery distance is shortened as much as possible on the basis of ensuring that each order can be delivered on time.

Three distribution centers (pharmacies), are Yuxin, Yifeng, Good medicine, 20 customer points, set up 12 riders but not necessarily all [6]. There is corresponding coordinate information in Excel, where the serial number behind the pharmacy is just convenient to correspond with the customer point, such as Yifeng 1 corresponds to the customer point 1, Good medicine 17 corresponds to the customer point 17. Because it is the first take and then send, the demand of the pharmacy is set to a positive number, and the demand of the corresponding customer point is set to a negative number. The coordinates of the rider are the initial position of the rider, and the rider does not need to return to the initial position after the completion of the delivery task. Set a constraint for delivery within 30 minutes (u is 30 minutes) so there is no time window [7].

Because the coordinates can only calculate the spherical distance, the actual driving will be detoured afterwards, and the distance is longer, so the distance needs to be multiplied by 2 based on the spherical distance.

Assumptions. Assume that the rider's electric vehicle is an electric vehicle of the same model, and all parameters such as electric vehicle capacity, maximum mileage, power consumption and other parameters are the same; It is assumed that electric vehicles all travel at the same speed and have the same speed. Assume that the electric vehicle power is sufficient, do not consider the case of power depletion; It is assumed that the electric vehicle has no maximum driving range constraint; Assume that there are no special goods, such as the volume of a single cargo exceeds the maximum capacity of the electric vehicle; Assume that the customer receives the goods no matter when the rider delivers them; Assume that the service time of the rider's delivery at the customer's point is the same; The dispensing time of the drugstore merchant and the pick-up time of the rider at the drugstore are not considered; Do not consider the pick-up point, that is, the drugstore is out of stock, etc.; Do not consider the weather, traffic and other uncontrollable factors; The coordinates of nodes are represented by latitude and longitude, and the distance between nodes can be calculated by formula (1) :

$$D = 2R \times \sin^{-1} \sqrt{\sin^2 \frac{W_1 - W_2}{2} + \cos W_1 \times \cos W_2 \times \sin^2 \frac{J_1 - J_2}{2}} \tag{1}$$

(Where R is the radius of the Earth, W is the latitude, and J is the longitude)

Parameters and Symbols. B: Drugstore collection, that is, collection of all pickup points,

$$B = \{1,2,3, \dots, i, \dots, n\}$$

C: Customer set, that is, all delivery points set, $C = \{n+1, n+2, \dots, n+i, \dots, 2n\}$;

$N = B \cup C$, indicating the collection of all pick-up and delivery points;

K: set of riders, $K = \{1,2,3, \dots, m\}$;

h_k: initial position node of rider k, $h_k = 2n + k$, $\forall k \in K$;

γ_k: virtual end distribution node of rider k, $\gamma_k = 2n + m + k$, $\forall k \in K$;

$M_k^1 = N \cup \{h_k\}$, $\forall k \in K$;

$M_k^2 = N \cup \{\gamma_k\}$, $\forall k \in K$;

$M_k = N \cup \{h_k, \gamma_k\}$, denotes the set of all nodes that rider k can access;

Parameters:

a_k: indicates the maximum load volume of rider k;

α_i: represents the service time spent by the rider to pick up the delivery at node i;

u: All orders need to be delivered within the set time u;

d_{ij}: the distance between nodes i and j;

t_{ij}: the time from node i to node j;

q_i: The volume of the drug-loaded and unloaded by the rider at node i is positive at the pick-up point [8] and negative at the delivery point;

Decision variables:

x_{ijk}: Rider k takes 1 from node i to node j, otherwise takes 0;

y_{lk}: Take 1 when the l order is delivered by rider k, otherwise take 0;

Other derived variables:

b_{ik}: the time when rider k arrives at node i and begins service;

Q_{ik}: the load of rider k when he reaches node I;

Model. Driving cost: The travel cost is related to the delivery distance, and the delivery distance is determined by both the order assigned to the rider and the order in which the rider serves the order. To some extent, the driving cost also represents the driving distance, and the smaller the driving cost, the shorter the total distribution path, that is, the shortest distribution time. The cost of travel is shown in formula (2), where c is the cost per unit distance traveled by the rider.

$$f_1 = \sum_{k\in K} \sum_{i,j\in M_k^1,\forall k\in K} c \times d_{ij} \times x_{ijk} \tag{2}$$

Performance cost: For human reasons, many platforms offer performance pay to encourage riders to fulfill as many orders as possible. Since the research purpose of this paper is only related to the delivery distance and has nothing to do with the number of orders completed by the rider, this paper only considers the performance cost based on the delivery distance. The smaller the performance cost, the shorter the total distribution path. The performance cost is shown in formula (3), where v is the rider's performance per unit distance traveled.

$$f_2 = \sum_{k\in K} \sum_{i,j\in M_k^1,\forall k\in K} v \times d_{ij} \times x_{ijk} \tag{3}$$

Objective function: The objective function of this paper is the minimum total platform cost, as shown in formula (4). The objective function consists of two parts, where f_1 represents the driving cost of all riders and f_2 represents the performance cost of all riders.

$$min\, f_1 + f_2 \tag{4}$$

Constraints:

$$\sum_{k\in K} y_{lk} = 1, \forall l \in B \tag{5}$$

$$x_{ilk} \le y_{lk}, \forall l \in B, \forall i \ne l \in M_k^1, \forall k \in K \tag{6}$$

$$x_{i,l+n,k} \le y_{lk}, \forall l \in B, \forall i \ne l+n \in M_k^1, \forall k \in K \tag{7}$$

$$\sum_{k\in K} \sum_{i\ne j\in M_k^1,\forall k\in K} x_{ijk} = 1, \forall j \in B \tag{8}$$

$$\sum_{i\ne j\in M_k^1} x_{ijk} = \sum_{i\ne j\in M_k^2} x_{jlk}, \forall j \in N, \forall k \in K \tag{9}$$

$$\sum_{i\ne j\in M_k^1} x_{ijk} = \sum_{i\ne j\in M_k^2} x_{j+n,l,k}, \forall j \in B, \forall k \in K \tag{10}$$

$$b_{ik} + \alpha_i + t_{ij} \le b_{jk} + \left(1 - x_{ijk}\right) \times M, \forall i \in M_k^1, \forall j \in M_k^2, \forall k \in K \tag{11}$$

$$\sum_{j\in M_k^2} x_{2n+k,j,k} = 1, \forall k \in K \tag{12}$$

$$\sum_{j\in M_k^1} x_{j,2n+m+k,k} = 1, \forall k \in K \tag{13}$$

$$b_{ik} \le b_{i+n,k}, \forall i \in B, \forall k \in K \tag{14}$$

$$b_{i+n,k} \le u, \forall i \in B, \forall k \in K \tag{15}$$

$$Q_{ik} + q_i \le Q_{ik} + M \times \left(1 - x_{ijk}\right), \forall i \in M_k^1, \forall j \in M_k^2, \forall k \in K \tag{16}$$

$$Q_{2n+m+k,k} = 0, \forall k \in K \quad (17)$$

$$Q_{ik} \leq a_k, \forall i \in N, \forall k \in K \quad (18)$$

Variable constraints:

$$x_{ijk} \in \{0,1\}, \forall k \in K, \forall i,j \in N \quad (19)$$

$$y_{lk} \in \{0,1\}, \forall k \in K \quad (20)$$

$$b_{ik} \geq 0, \forall i \in N, \forall k \in K \quad (21)$$

$$Q_{ik} \geq 0, \forall i \in N, \forall k \in K \quad (22)$$

The constraint (5) means that each order can only be delivered by one rider. The constraints (6) and (7) represent the satisfied relationship between two 0-1 variables. The constraint (8) is used to ensure that each order is assigned. The constraint (9) is used to ensure that nodes in the network are circulating. Constraint (10) means that the same order can only be picked up and delivered by the same rider. Constraint (11) represents the time relationship that needs to be satisfied between two adjacent nodes for the same rider, where M is the maximum constraint. The constraint (12) indicates that the rider must start from the initial node [9]. Constraint (13) indicates that the rider must return to the corresponding virtual destination after completing the delivery task. Constraint (14) indicates that the pick-up time must be met earlier than the delivery time for the same order. Constraint (15) means that for the same order, the delivery time cannot exceed the set time u. The constraints (16) and (17) represent the load change relationship of riders at two adjacent nodes. Constraint (18) means that the rider's cargo load must never exceed the maximum load. The constraints (19) and (20) are constraints on the 0-1 decision variable. The constraints (21) and (22) are constraints on other derived variables [10].

2.2 Algorithm

According to the constraints, the initial solution is generated.

Destruction operator. In this paper, three kinds of destruction operators are applied to remove the customer from the current solution, and after removing the customer, the customer is placed in the Destroy List (DL), waiting for subsequent repair operators to operate on it. The following describes the three destruction operators.

Random destroy randomly selects and removes F customer points from the current solution. This destruction operator has the advantage of fast computation and avoids local optimality through randomness. The operation of randomness removal can increase the randomness of ALNS algorithm, increase the diversity of population, and avoid the algorithm falling into the local optimal solution.

The core idea of Worst destroy is to remove the customer whose position is unsuitable in the current solution. It determines the suitability of the customer's position by calculating the cost savings after the customer's position is removed from the current solution. The higher the cost savings, the more unsuitable the customer's position in the current solution. TCj and TCi represent the current solution cost before and after the customer is removed, respectively. The greater the ΔTZ, the higher the cost savings after the customer is removed [11].

Calculate the ΔTZ of all customers and sort them from highest to lowest, removing customers in order. To increase the randomness of the operator, the number of removals is evenly extracted from F, and the operator can remove some customers that increase the cost, so as to achieve the purpose of reducing the cost.

Related destroy Removes customers in pairs according to the correlation between customers. The correlation is calculated as follows:

First, randomly select a customer i to remove, then calculate the degree of correlation between other customers and i and rank them in descending order, starting from the most relevant customers to remove customers until enough customers are removed.

The following three repair operators reinsert the customer from the Destroy List into the solution.

Greedy repair removes the customer and inserts it into the best position, calculates the insertion path v of customer j, and the insertion cost Δ = ΔI(i,r,q)- di - dj - di at the position, and inserts customer j into the optimal node best loc. If all insertion points are not feasible to insert, a new path is taken. Until all the removed customers are returned to the path, and the new solution is finally obtained.

The insertion position of the customer point is determined with Regret repair according to the insertion cost regret value of the customer. By recalculating the insertion cost of customer points in DL to the second waiting position of the new solution, and estimating the regret value of inserting the current position accordingly, the difference between the two is the largest, and the regret value is large, which means that if the current node is not inserted, the node will be selected to be inserted, and then the insertion is performed, and the regret value is inserted from the high point of the regret value. The formula for calculating the regret value is as follows: $reg_k = IZ' - IZ_j'$

On behalf of the regret value after the insertion of candidate node k by the customer point, the value with the highest regret value is selected for insertion, and the new solution is obtained by cyclic selection until the end of insertion.

Random repair randomly inserts customers from DL back into a path of the solution until all customer points in DL are reinserted into the path to produce a new solution.

If only the optimal solution is accepted in the optimization process, the algorithm is prone to fall into the local optimal. To avoid this situation, the simulated annealing algorithm is used to allow a certain different solution to be accepted. In the annealing process, a lower difference solution is accepted with a certain probability according to the Metropolis criterion, and the acceptance probability decreases with the decrease in temperature. If the total cost of the new solution is less than the optimal solution, the solution is accepted, and both the optimal solution and the current solution are updated. If the total cost of the new solution is greater than the optimal solution but less than the total cost of the current solution, the solution is accepted and the current solution is updated. If the total cost of the new solution is greater than the total cost of the current solution, the new solution is accepted with a certain probability, which is calculated as follows:

$$RE_e = e^{\frac{Z_{new} - Z_{now}}{TEM}}$$

Where: Z_{new}, Z_{now} are the total cost of the current solution and the new solution respectively; e is a natural constant; TEM Indicates the current temperature. The temperature decreases at the cooling rate ΔTEM. The initial temperature is TEM_0, and the end temperature is TEM_f.

Considering the difference in the ability to explore the solution between the damage operator and the repair operator, the score weight of the operator is updated according to the acceptance criteria, and the damage operator and the repair operator are selected by the roulette algorithm. At the beginning, the score weights of all operators are the same. In the iterative process, the adjustment sub-operators are set up, and the weight of the newly solved parts is determined by roulette. The purpose of the operator strategy is to increase the number of operators and to determine the probability of selecting the sub destruction in the iteration process according to the temperature and the weight of the sub operator. In the process of selection, ALNS can adjust the operator weight adaptively according to the performance of the operator and strengthen the search.

3 Analysis Of Experimental Results

3.1 Experimental Parameter Settings

The maximum load volume of the rider is 200, the specified time is 20 minutes, the drug collection time is 0.5 minutes, and the rider service time is 1 minute

3.2 Result Analysis

After 100 iterations of ALNS algorithm, the iteration diagram is obtained, as shown in Figure 1. The optimal solution of the model is reached in 60 iterations. Through the iteration diagram, it can be found that the optimal result is obtained at about the 55th iteration. It can also be seen from the convergence diagram that the designed algorithm can effectively jump out of the local optimal solution and improve the convergence of the algorithm because of the addition of the new solution acceptance criterion of simulated annealing. It is worth noting that in the distribution paths, there are cases where the paths delivered by electric vehicles (red line) cross the paths delivered by fuel vehicles (green line) because the algorithm finds that the total cost is lower when these points are delivered by electric vehicles during the iteration process.

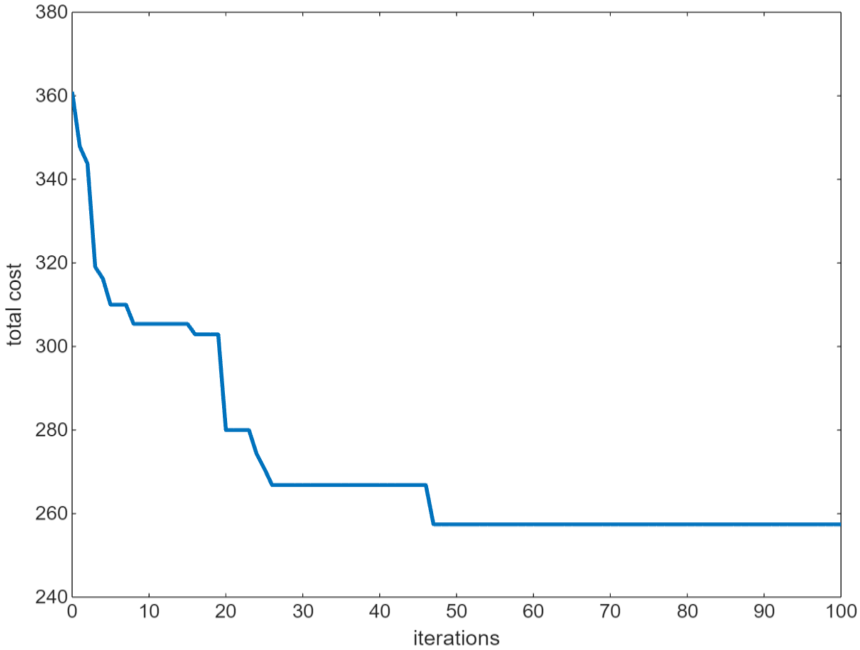

Fig. 1. Iteration diagram of ALNS algorithm

In the figure, there are a total of 8 delivery riders, of which 2 riders are idle and 6 riders are responsible for the delivery task of all demand points, each rider is responsible for the arrangement of the demand points as shown in the table, the 1st rider starts from the initial position, goes to pharmacy 2, pharmacy 1 picks up the medicine, and then is responsible for the delivery of the customer points 16, 3, 9, 20, 12, and 5, and then returns to the initial position; the time taken is 18.77 minutes, and the form cost of $52.14 and performance cost of $21.47, the second rider traveled from the initial position to Pharmacy 1 to pick up the medication, and subsequently delivered the medication to Client 8 and Client 10 before returning to the initial position; the time taken was 4.69 minutes, the cost of traveling was $9.30, and the cost of performance was $3.83, and the third rider went to Pharmacy 1 to pick up the medication, then satisfied Client Point 7, and then went to Pharmacy 2 to pick up the medication, then satisfy customer points 22, 11, 23, and 14, and then return to the initial position, with a time bit of 18.41 minutes, a driving cost bit of $52.72, and a performance cost bit of $21.71; Rider 4 goes from the initial position to Pharmacy 2 to pick up the medication, and returns to the starting point after visiting only customer point 17, with a time bit of 2.45 minutes, a driving cost bit of $4.04, and a performance cost bit of 1.66, Rider 6 goes to Pharmacy 1 and Pharmacy 2 to pick up medication from the initial position and subsequently delivers to customer points 21, 19, 4, 6, 15, 18, 29 in turn and then returns to the starting point, with time consumed bit 18.64 minutes, traveling cost bit $49.47, and performance cost bit $20,372, and Rider 8 goes to Pharmacy 1 to pick up medication from the initial position and visits only customer point 13.Fig. 2 illustrates the rider pickup and delivery path scenarios, and Figure 3 shows the time Gantt chart scenarios of each rider departing to the pharmacy to pick up the medication, delivering it to the customer, and finally returning to the starting point.

Table 1. Rider Assignment and Delivery Performance Summary

Rider number	Route	Time used/min	Running cost/yuan	Performance cost/yuan
1	24-2-1-16-3-9-20-12-5-24	18.7705	52.1497	21.4734
2	25-1-8-10-25	4.6904	9.3093	3.8333
3	26-1-7-2-22-11-23-14-26	18.406	52.7253	21.7104
4	27-2-17-27	2.4517	4.0446	1.6654
5				
6	29-1-2-21-19-4-6-15-18-29	18.6411	49.4749	20.372
7				
8	31-1-13-31	3.9458	10.3947	4.2802

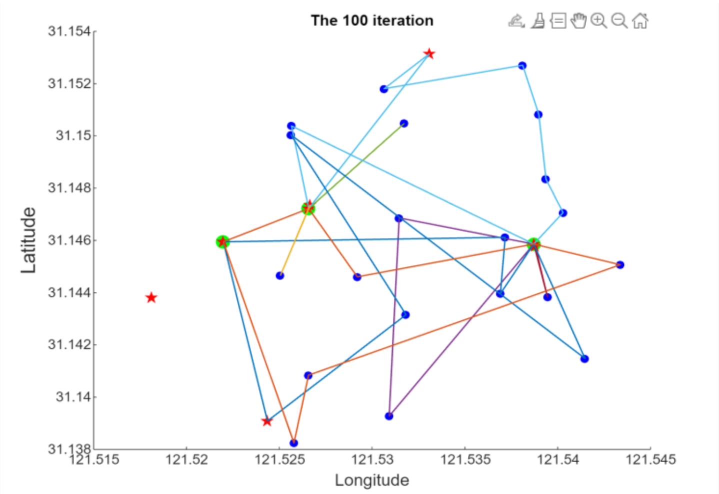

Fig. 2. Optimal distribution path diagram (The red five-pointed star indicates the initial location of the rider, the green circle indicates the location of the pharmacy, and the blue circle indicates the location of the demand point.)

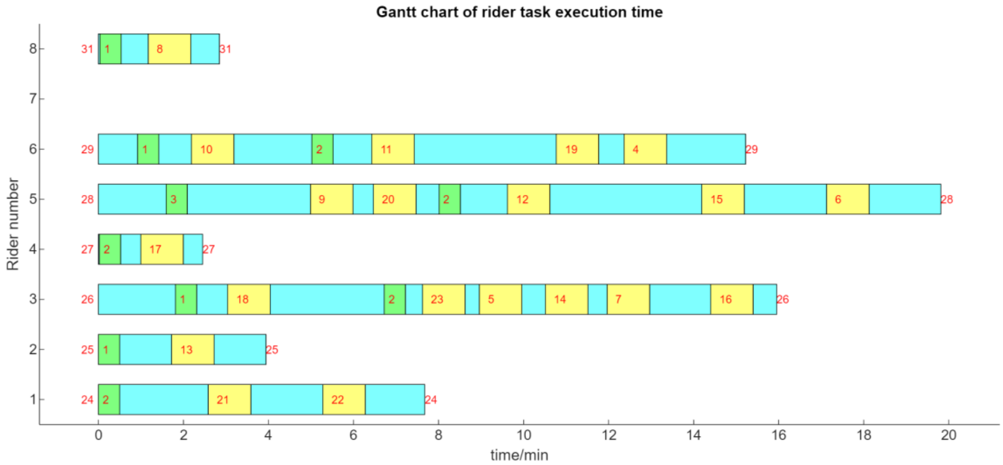

Fig. 3. Time arrangement diagram of the rider performing the task (Green is the pick-up period at the pharmacy, blue is the driving period, and yellow is the service period.)

4 Conclusion

This paper proposes and verifies the effectiveness of the adaptive large-scale neighborhood search algorithm (ALNS) for drug delivery route optimization with capacity constraints. By establishing the VRP model and combining it with the actual distribution demand, this paper studies how to minimize the total distribution distance based on ensuring the punctual delivery of each order. In the design of the model, multiple constraints such as time window, load limit and driving cost were comprehensively considered. Three kinds of damage operators (random damage, worst damage and correlated damage) and three kinds of repair operators (greedy repair, regret repair and random repair) were applied to further optimize the solution process of understanding.

The experimental results show that the ALNS algorithm can effectively find the optimal distribution path scheme after several iterations, significantly reduce the distribution distance and time, and improve distribution efficiency. The research results of this paper provide an important reference value for the actual drug delivery system, which can effectively improve the timeliness of drug delivery and the rate of patient treatment.

In this paper, the effectiveness and feasibility of the proposed algorithm are verified by detailed experimental data and analysis, which provides a new idea and method for solving the complicated drug delivery route optimization problem. Future studies can further optimize the algorithm to consider more practical constraints, such as traffic conditions and weather effects, in order to improve the applicability and robustness of the algorithm in practical applications.

References

[1] Dantzig G., Ramser H, The Truck Dispatching Problem [J]. Management Science,1959,6(1):80-91.

[2] Clarke G., Wright J. W, Scheduling of Vehicles from a Central Depot to a Number of Delivery Points [J]. Operations Research, 1964,12(4):568-581.

[3] Lenstra J. K., Kan A, Complexity of vehicle routing and scheduling problems [J]. Networks, 2010,11(2):221-227.

[4] Gayialis S.P, Kechagias E.P, A Multiobjective Large Neighborhood Search Metaheuristic for the Vehicle Routing Problem with Time Windows [J]. Algorithms, 2020,13(10):243.

[5] Corstjens J., Depaire, B., Caris A, A multilevel evaluation method for heuristics with an application to the VRPTW [J]. International Transactions in Operational Research,2020,27(1):168-196.

[6] Gutierrez A., DieulleL., LabadieN, A multi-population algorithm to solve the VRP with stochastic service and travel times [J]. Computers & Industrial Engineering,2018, 125:144-156.

[7] Lu Zhen, Chengle Ma, Kai Wang, Liyang Xiao, Wei Zhang. Multi-depot multi-trip vehicle routing problem with time windows and release dates [J]. Transportation Research Part E, 2020, 135(C):1-21.

[8] Song C. S., Sungs S. H, Integrated Service Network Design for a Cross-Docking Supply Chain Network [J]. The Journal of the Operational Research Society, 2003, 54(12):1283-1295.

[9] Musa R, Arnaout J P, Jung H. Ant colony optimization algorithm to solve for the transportation problem of cross-docking network [J]. Computers & Industrial Engineering, 2010,59(1):85-92.

[10] Mehmet G., James H., Bookbinder. CROSS-DOCKING AND ITS IMPLICATIONS IN LOCATION-DISTRIBUTION SYSTEMS [J]. Journal of Business Logistics, 2004, 25(2):199-228.

[11] Miao, Z., Yang, F., Fu, K. et al. Transshipment service through crossdocks with both soft and hard time windows. Ann Oper Res 192, 21–47 (2012). https://doi.org/10.1007/s10479-010-0780-4