Two-Phase Algorithm to Multiple Depots Vehicle Routing Problem with Soft Time Windows

Ailing Chen¹, Xinxin Gu²*, Zhao Gao³

¹School of Management Science and Engineering Shandong University of Finance and Economics Jinan, China
²School of Management Science and Engineering Shandong University of Finance and Economics Jinan, China
³School of Management Science and Engineering Shandong University of Finance and Economics Jinan, China
* gxx9697@163.com

Abstract—With the rapid growth of social economy, many companies often have multiple distribution centers. In this paper, multiple depots vehicle routing problem with soft time windows (MDVRPSTW) is researched. A mathematical model, which is aiming to minimize the transport cost, is established. Two-phase algorithm is designed to solve this problem. In the first phase, customers’ orders are distributed to the distribution center by exact algorithm. In the second phase, all the route of each distribution center is calculated. Because this problem is an NP-hard problem, it is difficult to get optimal solution fast when calculating large scale examples, improved genetic algorithm is adopted to solve this problem. In this paper, traditional roulette selection method is abandoned in the stage of designing the selection operator, so as to avoid that the early high fitness individuals occupy the population rapidly and the later population stops evolution due to the small difference of individual fitness. Finally, a practice example of a certain company is calculated by the above method to verify the efficiency of algorithm in this paper.

1. INTRODUCTION

Vehicle routing problem (VRP) was first raised by Dantzig in 1959[1]. With the development of modern commerce, many enterprises have established more than one distribution center in the region, so multi-depot vehicle routing problem (MDVRP) is very important. Multi-depot vehicle routing problem is a variant of single depot VRP, when the time window restriction is added to MDVRP, MDVRP becomes a multi-depot vehicle routing problem with time windows (MDVRPTW). It has become the goal of modern logistics industry to reduce costs and improve customer service satisfaction by scientifically planning distribution and transportation routes. The objective function includes not only the driving cost but also the delivery time.

Some researches focus on the multi-depot vehicle routing problem. Du et al. (2017) [2] studied a MDVRP for hazardous materials transportation and developed a fuzzy bilevel programming model and designed four algorithms. Li at al. (2019) [3] applied an ant colony algorithm to solve the MDVRP with multiple objectives. Singh et al. (2019) [4] presented a hybrid clustering based genetic algorithm to solve multi depot vehicle routing problem.
Some researches focus on the vehicle routing problem with time windows. Hashemi et al. (2020) [5] studied a VRP which combines multi-trip and time windows, and a node-based formula and a scene-based formula are provided. Wang et al. (2020) [6] considered customer satisfaction and proposed an algorithm to solve the low-carbon vehicle routing problem (LC-VRP).

With the complexity and development of the problem, time windows and multiple-depot have been combined to study by scholars. Li et al. (2016) [7] considered the following problem: the depot where the vehicle ends and the depot that starts may not be the same. Shen et al. (2018) [8] proposed a two-phase algorithm to solve a low-carbon multi-depot open vehicle routing problem with time windows. Dridi et al. (2019) [9] dealt a VRP that combines multiple depots, pickup and delivery and time windows, and developed a new algorithm to solve it. Khachay et al. (2019) [10] used a polynomial time approximation schemes to solve the vehicle routing problem with time windows and multiple depots.

The heuristic algorithm is adopted in most of the above literatures. However, it is well known that the use of heuristic algorithms will produce calculation errors. To reduce errors, there are two methods. The first, exact algorithm could be use, but it will take a lot of time when calculating large-scale examples. The second method is to optimize the heuristic algorithm and improve the calculation efficiency by redesigning the operator. In this paper, a new method, that is, the solution of MDVRPTW is divided into two parts, is adopted. In the first part, the exact algorithm is used to assign customers to the depot based on the principle of minimum transportation distance. Then, planning the traveling trips to vehicles is figured out by improved genetic algorithm.

2. Model Formulation

2.1. Problem description

The proposed model assumes that: (a) there are distribution centers in this problem. (b) each depot has vehicles with the same fixed cost and unit transportation cost. (c) there are M customers to be served and each is served only once. (d) each customer has a demand, an earliest and a latest service time $[E_i, L_i]$. (e) every route starts and ends at a distribution center.

2.2. Notations

Parameters

$V = \{1, 2, \ldots, N, \ldots, N + M\}$: set of nodes;

$i \& j$: node index, $i \& j = 1, 2, \ldots, N, \ldots, M+N$, in which 1 to N represent distribution centers;

$n$: depot index;

$k$: vehicle index.

$N$: total number of depots;

$M$: total number of customers;

$Q$: maximum capacity of vehicles;

$k_n$: the number of vehicles of the nth depot;

$q_i$: the demand of customer $i$;

$d_{ij}$: the distance from $i$ to $j$;

$s_i$: the server time of customer $i$;

$E_i$: the earliest time to service customer $i$;

$L_i$: the latest time to serve customer $i$;

$T_i$: arrival time of customer $i$;

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

$c_1$: fixed cost of vehicle;

$c_2$: unit transportation cost of vehicle;

$v$: average speed of vehicle;
Variables

\[ x_{ij}^{nk} = \begin{cases} 1 & \text{if vehicle } k \text{ in depot } n \text{ travels from } i \text{ to } j \\ 0 & \text{otherwise} \end{cases} \]

\[ y_{ij}^{nk} = \begin{cases} 1 & \text{if } i \text{ is served by vehicle } k \text{ in depot } n \\ 0 & \text{otherwise} \end{cases} \]

2.3. Penalty function

In real life, customers often have specific requirements for service time. One of the important ways to improve customer satisfaction is to meet customer service time requirements. The customer's delivery service time generally refers to a time range: the earliest service time and the latest service time, which is the time window \([E_i, L_i]\). Soft time window means that if the delivery vehicle does not deliver the goods within the time period requested by the customer, the corresponding punishment will be based on the length of time the vehicle arrived early or late. The penalty function is as follows:

\[ P_i = \begin{cases} p_1 (E_i - T_i) & T_i < E_i \\ 0 & E_i \leq T_i \leq L_i \\ p_2 (T_i - L_i) & T_i > L_i \end{cases} \]

\(p_1\) represents the unit waiting cost of early arrival of vehicles, and \(p_2\) represents the unit penalty cost of late arrival of vehicles.

![Penalty function of customer soft time window](image)

As is shown in figure 1, \(E_i\) represents the earliest service time, \(L_i\) represents the latest service time, Vehicle arrival time should be earlier than \(L_i\) and later than \(E_i\), otherwise the penalty cost will be occurred.

2.4. Formulation

Objective function

The total cost includes vehicle fixed cost, transportation cost and time penalty cost.

\[
\min \left( \sum_{j=1}^{N} \sum_{n=1}^{K_n} c_{ij} x_{ij}^{nk} + \sum_{j=1}^{N} \sum_{m=1}^{K_m} c_{ij} d_{ij} \right) + \sum_{j=1}^{N} \sum_{k=1}^{K_k} y_{ij}^{nk} \\
\left( p_1 \max (E_i - T_i, 0) + p_2 \max (T_i - L_i, 0) \right) \]

(1)

Constraints

\[
\sum_{j=1}^{N} x_{ij}^{nk} q_i \leq Q \quad (\forall n \in N, k \in K_n) \\
(2)
\]

\[
\sum_{j=1}^{N} \sum_{k=1}^{K_k} x_{ij}^{nk} \leq K_n \quad (\forall i = n \in N) \\
(3)
\]
\[ \sum_{n=1}^{N} \sum_{k=1}^{K_n} y_{nk}^i = 1 (\forall i \in M) \quad (4) \]

\[ \sum_{j=N+1}^{M+N} x_{ij}^n = \sum_{j=1}^{N} x_{ij}^n (\forall i \in M + N, n \in N, k \in K_n) \quad (5) \]

\[ \sum_{j=N+1}^{M+N} x_{ij}^n = \sum_{j=1}^{N} x_{ij}^n (\forall i = n \in N, k \in K_n) \quad (6) \]

\[ y_{ij}^n \leq x_{ij}^n (\forall i \in M, j \in M, n \in N, k \in K_n) \quad (7) \]

\[ x_{ij}^n, y_{ij}^n \in \{0, 1\} (\forall i \in N + M, j \in N + M, n \in N, k \in K_n) \quad (8) \]

Equation (1) means to minimize the total cost. Equation (2) constraints the vehicle capacity. Equation (3) guarantees the number of vehicles departing from the depot is less than that of all vehicles in the depot. Equation (4) guarantees that each customer is served only once. Equation (5) is a flow conservation constraint, which ensures that the vehicle arrives and leaves the same customer. Equation (6) guarantees that vehicles depart from the depot and eventually return to the original depot. Equation (7) is a constraint on the relationship of variables. Equation (8) defines variables in the model.

3. Algorithm Design

Because MDVRPSTW is a NP-hard problem, the optimal solution cannot be obtained with an exact algorithm in a short time. In this paper, the algorithm based on genetic algorithm and the exact algorithm is designed to get the optimal vehicle routes in two steps. That is, customers are first assigned to the depot, and then the distribution route of each depot is planned. At the same time, this paper uses the method of "exact algorithm first, then heuristic algorithm" to solve the problem. Firstly, through the calculation of accurate algorithms, the customers assigned to each depot are obtained based on the distance, and then the genetic algorithm is designed to plan the route for each depot.

3.1. Exact algorithm

A mathematical model is established to assign customers to each depot. Here are the formulas.

\[ \min \sum_{i=1}^{N} \sum_{j=N+1}^{M+N} x_{ij}^n d_{ij} \quad (9) \]

\[ \sum_{j=1}^{N} x_{ij} = 1 (\forall j \in M) \quad (10) \]

\[ \sum_{j=N+1}^{M+N} x_{ij} a_j \leq c_i (\forall i \in N) \quad (11) \]

\[ c_i = Q K_n (\forall i = n) \quad (12) \]

\[ x_{ij} \in \{0, 1\} (\forall i \in N, j \in M) \quad (13) \]

c_i represents the capacity of distribution center i. Equation (9) is to minimize the total distance. Equation (10) ensures that each customer is assigned to only one distribution center. Equation (11) guarantees the total demand of all customers served by each depot does not outstrip the capacity of the depot. Equation (12) indicates that the capacity of the depot is equal to the sum of the vehicle capacity of all vehicles in it. Equation (13) defines variables in the model.

3.2. Genetic algorithm

Genetic algorithm has a wide range of applications. In vehicle routing problem, it mainly goes through the following six steps:

a) Chromosome coding

Gene is the smallest unit of genetic algorithm, a natural number represents a gene, chromosome is the carrier of gene, a chromosome has multiple genes. Different chromosomes can cross each other at random. In genetic algorithm, the results of feasible solution and optimal solution are usually expressed by chromosomes. In this article, 1 represents the distribution center. The first gene of each chromosome
is 1, and then randomly generates 2–n natural numbers to represent all customers, forming an initial chromosome, such as (138265749) as shown in the figure 2.

![Initial Chromosome](image)

The chromosome string (138126517491) of eight customers served by three vehicles means that the first vehicle starts from the distribution center and serves customers 3 and 8 in turn, the second vehicle serves customers 2, 6 and 5, the third vehicle serves customers 7, 4 and 9 and then returns to the distribution center, forming three routes in total.

b) Population
The initial population is the initial set of chromosomes, and the number of chromosomes in the population is preferably 20 to 200. In this paper, the initial population is a randomly generated 50 groups of numbers.

c) Fitness
The fitness function of this paper is the total distance and time penalty cost of the route, which is completed in three steps.

Firstly, according to the vehicle capacity, the initial chromosome is processed into a chromosome that can represent several lines, i.e. line segmentation, such as (13826517491) as shown in the figure 3.

![Processed Chromosome](image)

Secondly, the distribution cost and time penalty cost of each chromosome are calculated as follows:

$$fitness(z) = \sum_{r=1}^{R} \left( d_{r,r+1} + p_1 \max(E_r - T_r, 0) + p_2 \max(T_r - L_r, 0) \right)$$  \quad (13)$$

R represents the number of all genes of the processed chromosome, and $d_{r,r+1}$ represents the distance from the gene r to the gene r+1 on the chromosome.

Finally, the fitness of all chromosomes in the population is calculated by the above formula.

d) Selection
The purpose of this step is to select the better and eliminate the worse chromosomes, and ensure that the best chromosomes can survive to the next generation. By calculating the fitness of chromosomes in the population, the chromosomes are arranged from small to large, and the chromosomes with poor fitness are eliminated according to a certain elimination rate. At the same time, the same number of chromosomes with excellent fitness are copied to make up for the eliminated chromosomes and keep the number of chromatids in the population unchanged. This step improves the selection method, abandons the roulette selection, avoids all the individuals in the group falling into the same extreme value and thus stop the evolution. Meanwhile, the offspring always keeps the best individuals in the parent generation, and searches for better samples in the direction of "high fitness mode as ancestor", so as to ensure that the global optimal solution can be searched finally.

e) Crossover
Crossover is an important condition for the optimization of the genetic algorithm. It can increase the diversity of chromosomes. The steps are as follows:

1) Randomly generated range of crossing.

![Crossing Scope](image)
(2) Exchange genes from the crossing scope to generate two new chromosomes.

\[
\begin{align*}
A'&: 1 \ 3 \ 2 \ 6 \ 3 \ 5 \ 4 \\
B'&: 1 \ 5 \ 7 \ 0 \ 2 \ 4 \ 7 \\
\end{align*}
\]

Figure 5. Chromosome crossover

(3) Modify the chromosome to ensure that there are no duplicate genes in the new chromosome. If there is a duplicate gene, replace the duplicate gene with the missing one.

\[
\begin{align*}
A''&: 1 \ 3 \ 2 \ 6 \ 7 \ 5 \ 4 \\
B''&: 1 \ 5 \ 3 \ 6 \ 2 \ 4 \ 7 \\
\end{align*}
\]

Figure 6. Chromosome correction

f) **Mutation**

In order to prevent genetic operation from falling into local optimum, mutation is introduced to increase chromosome diversity.

First, set a mutation rate and determine the customer point of the mutation, and then randomly generate a customer code as the replacement location. Here is an example to illustrate the mutation operation:

\[
\begin{align*}
\text{Mutation Location} &: 1 \ 3 \ 8 \ 2 \ 6 \ 5 \ 7 \ 4 \ 9 \\
\text{After Mutation} &: 1 \ 3 \ 5 \ 2 \ 6 \ 8 \ 7 \ 4 \ 2 \\
\end{align*}
\]

Figure 7. Mutation Processing

g) **Terminating conditions**

The evolution Terminates when the set number of iterations is reached.

h) **Steps of genetic algorithm**

First, a group of initial population is generated by constructing chromosomes, then the fitness of each chromosome in the population is calculated, and the genetic operations such as selection, crossover, mutation are performed in turn, until the termination condition is satisfied.

![The flow chart of genetic algorithm](image-url)
### 3.3. Efficiency verification

In order to verify the effectiveness of genetic algorithm, several standard examples on “https://www.sintef.no/projectweb/top/vrptw/solomon-benchmark” are calculated. The information of these examples is shown in Table 1.

| No. | Name  | Scales | Capacity |
|-----|-------|--------|----------|
| 1   | R101.25 | 26     | 200      |
| 2   | R101.50 | 51     | 200      |
| 3   | R101.100 | 101    | 200      |
| 4   | C101.25 | 26     | 200      |
| 5   | C101.50 | 51     | 200      |
| 6   | C101.100 | 101    | 200      |

The experimental environment is Inter Core i5, the computer memory is 16 G, whose operating system is Windows 10. All examples are calculated by MATLAB R2018a. Meanwhile, the genetic algorithm parameters are set as follow: population size = 100; iteration = 500; mutation probability = 0.01; selection probability = 0.4. The results of each example are shown in Table 2.

| Name  | Scales | Distance | Exact Distance | Error Rate |
|-------|--------|----------|----------------|------------|
| R101.25 | 26     | 650.99   | 617.1          | 0.05       |
| R101.50 | 51     | 1255.697 | 1044           | 0.20       |
| R101.100 | 101   | 2230.562 | 1637.7         | 0.36       |
| C101.25 | 26     | 213.0888 | 191.3          | 0.11       |
| C101.50 | 51     | 424.3031 | 362.4          | 0.17       |
| C101.100 | 101   | 1116.866 | 827.3          | 0.35       |

Through the calculation of examples, it can be found that the error rate increases gradually with the increase of the sample size. However, when the sample size is appropriate, the error rate of the genetic algorithm is acceptable, which proves the effectiveness of the genetic algorithm.

### 4. Numerical example

In order to verify the feasibility of the model and algorithm and solve practical problems, the algorithm was applied to the express delivery of Company J.

#### 4.1. Basic data

J company is an e-commerce enterprise which has 4 distribution centers in city B. Customers place orders online through the application, and can choose the arrival time of orders independently. The distribution centers carry out daily distribution to the post stations according to the customers' orders every day, and the order is generally fulfilled within 24 hours. The maximum load of the distribution vehicles is 760kg, so the capacity of each distribution center cannot exceed 2280 kg. and the number of vehicles in per distribution center is 3. The average speed of vehicles is 40km/h. Therefore, the actual distribution problem of company J is abstracted as the multi depot vehicle routing problem with time window.

By analyzing the order of J company on a certain day, 111 customer nodes are selected. 4 depots and part of customer information is shown in Table 3.
TABLE III. Depots and Part of the Customer Information

| Node ID | Location          | Soft time window | Demand |
|---------|-------------------|------------------|--------|
| distribution centers |
| 1       | 116.6486,40.11379 | [-∞,300]         | 0      |
| 2       | 116.5511,39.72694 | [-∞,300]         | 0      |
| 3       | 116.3651,39.6814  | [-∞,300]         | 0      |
| 4       | 116.8557,40.1238  | [-∞,300]         | 0      |
| customers |
| 1       | 116.3503,40.00051 | [79.00,209.00]   | 33.00576 |
| 2       | 116.2092,39.84573 | [29.00,210.00]   | 28.98765 |
| 3       | 116.3679,39.96732 | [3.00,151.00]    | 42.10459 |
| 4       | 116.3679,39.96732 | [63.00,228.00]   | 99.83581 |
| 5       | 116.8257,40.163   | [40.00,218.00]   | 96.94102 |
| 6       | 115.9019,40.37661 | [65.00,190.00]   | 61.87934 |
| 7       | 116.288,40.03371  | [84.00,208.00]   | 75.83544 |
| 8       | 116.3555,40.00991 | [67.00,205.00]   | 45.89193 |
| 9       | 116.3476,39.96406 | [26.00,165.00]   | 45.89193 |

4.2. Calculation result
First, LINGO is used to assign customers to each distribution center through exact algorithm. The assignment results are shown in Table 4.

TABLE IV. Assignment results

| Depot | Customer nodes |
|-------|----------------|
| 1     | 1, 6, 7, 8, 11, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 29, 30, 31, 44, 45, 46, 60, 61, 62, 64, 65, 66, 80, 93, 94, 97, 98, 100 |
| 2     | 3, 9, 28, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 47, 48, 51, 67, 68, 69, 70, 73, 74, 75, 76, 77, 78, 99, 102 |
| 3     | 2, 4, 10, 49, 50, 52, 53, 54, 55, 56, 57, 58, 71, 79, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 101, 103, 104, 105, 106, 107, 108, 109, 110, 111 |
| 4     | 5, 12, 13, 27, 59, 63, 72, 95, 96 |

Second, in order to get the final plan, genetic algorithm is used to calculate the optimal route of each distribution center as shown in table 5. The algorithm parameters are set as follows: Population size=100; mutation rate=0.01; elimination rate=0.4; maximum number of iterations=500.
### Table V. Vehicle Trips by the Genetic Algorithm

| Depot | Vehicle | Service sequence | Early and late arrivals (min) | Distance |
|-------|---------|------------------|-------------------------------|----------|
| 1     | 1       | 1->18->22->31->11->6->36->29->21->30->20->34->17->28->19->1 | 35.73 | 499.02 |
| 1     | 2       | 1->9->7->32->5->14->24->23->4->25->2->35->33->8->1 | | |
| 1     | 3       | 1->10->37->15->13->12->26->16->27->3->1 | | |
| 2     | 1       | 1->13->22->14->29->23->17->3->2->5->12->10->1 | 3.38 | 272.69 |
| 2     | 2       | 1->21->6->30->9->8->7->19->18->31->15->27->1 | | |
| 2     | 3       | 1->16->26->20->25->24->4->11->28->1 | | |
| 3     | 1       | 1->9->15->7->6->31->14->3->16->5->29->28->18->10->20->1 | 13.36 | 386.90 |
| 3     | 2       | 1->4->30->8->17->25->22->21->2->19->32->11->1 | | |
| 3     | 3       | 1->12->26->34->27->37->13->33->24->23->35->36->1 | | |
| 4     | 1       | 1->7->9->10->8->5->6->2->4->3->1 | 0 | 115.08 |

Finally, RMB 5 or 10 will be fined for every 10 minutes of vehicle arriving ahead or behind of required time, that is, p1 is 0.5 and p2 is 1. The total costs (including fixed costs, traveling costs and time penalty costs) are shown in Table 6.

### Table VI. Costs

| Depot | Vehicle number | Fixed cost (¥) | Traveling cost (¥) | Penalty cost (¥) | Total cost (¥) |
|-------|----------------|----------------|-------------------|-----------------|----------------|
| 1     | 3              | 615.07         | 329.35            | 27.87           | 972.29         |
| 2     | 3              | 615.07         | 179.98            | 2.69            | 797.74         |
| 3     | 3              | 615.07         | 255.35            | 10.68           | 881.10         |
| 4     | 1              | 205.02         | 75.95             | 0               | 280.97         |

#### 4.3. Comparison with the original distribution strategy

The comparisons before and after optimization are shown in Table 7.

### Table VII. Comparison of costs

| Current distribution strategy | Strategy after optimization |
|-------------------------------|-----------------------------|
| Fixed cost | Traveling cost | Penalty cost | Total cost | Fixed cost | Traveling cost | Penalty cost | Total cost |
| 2460          | 1064             | 50           | 3575        | 2050        | 841             | 41           | 2932        |
As can be seen from table 7, the number of vehicles after optimization is saved, and all costs have been reduced, in which fixed cost is reduced 16.67%, traveling cost is reduced 21.02%, penalty cost is reduced 17.78% and total cost is reduced 17.98%. It can be concluded the effective of genetic algorithm.

5. CONCLUSIONS
A multi-depot vehicle routing problem with soft time windows is studied, and a mathematical model for minimizing the total cost is established, then the exact algorithm and improved genetic algorithm are used to solve the problem. Finally, an actual case of a company is optimized. This paper uses the method of "first exact algorithm, then heuristic algorithm" to solve the problem. First of this, through the calculation of the exact algorithm, we get the customers assigned to each depot according to the distance, and then use the genetic algorithm to plan the route of each depot.

REFERENCES
[1] Dantzig, George B., and John H. Ramser. "The truck dispatching problem." Management science 6.1 (1959): 80-91.
[2] Du, Jiaoman, et al. "Multi-depot vehicle routing problem for hazardous materials transportation: a fuzzy bilevel programming." Information Sciences 399 (2017): 201-218.
[3] Li, Yongbo, Hamed Soleimani, and Mostafa Zohal. "An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives." Journal of cleaner production 227 (2019): 1161-1172.
[4] Singh, Varimna, L. Ganapathy, and Ashok K. Pundir. "An Improved Genetic Algorithm for Solving Multi Depot Vehicle Routing Problems." International Journal of Information Systems and Supply Chain Management (IJISSCM) 12.4 (2019): 1-26.
[5] Hashemi, Saeedeh, Majid Salari, and Mohammad Ranjbar. "Multi-Trip Open Vehicle Routing Problem with Time Windows: A Case Study." International Journal of Industrial Engineering 27.1 (2020).
[6] Li, Jian, Yang Li, and Panos M. Pardalos. "Multi-depot vehicle routing problem with time windows under shared depot resources." Journal of Combinatorial Optimization 31.2 (2016): 515-532.
[7] Shen, Ling, Fengming Tao, and Songyi Wang. "Multi-depot open vehicle routing problem with time windows based on carbon trading." International journal of environmental research and public health 15.9 (2018): 2025.
[8] Wang, Ziqi, and Peihan Wen. "Optimization of A Low-Carbon Two-Echelon Heterogeneous-Fleet Vehicle Routing for Cold Chain Logistics Under Mixed Time Window." Sustainability 12.5 (2020): 1967.
[9] Harbaoui Dridi, Imen, et al. "Optimisation of the multi-depots pick-up and delivery problems with time windows and multi-vehicles using PSO algorithm." International Journal of Production Research (2019): 1-14.
[10] Khachay, Michael, and Yuri Ogorodnikov. "Towards an efficient approximability for the Euclidean Capacitated Vehicle Routing Problem with Time Windows and multiple depots." IFAC-PapersOnLine 52.13 (2019): 2644-2649.