Study of variable neighborhood descent and tabu search algorithm in VRPSDP

Y Christopher, S Wahyuningsih* and D Satyananda
Department of Mathematics, Universitas Negeri Malang, Indonesia

*Corresponding author: sapti.wahyuningsih.fmipa@um.ac.id, E-mail: yuliochristo@gmail.com, darmawan.satyananda.fmipa@um.ac.id

Abstract. The application of graph theory can be used to solve distribution optimization problems, especially the Vehicle Routing Problems (VRP). One variant of VRP is the Vehicle Routing Problem with Simultaneous Delivery and Pickup (VRPSDP), which has particular constraints, namely requests, and returns, which are done simultaneously. The VRPSDP problems can be solved by a variety of algorithms, such as the tabu search algorithm and the Variable Neighborhood Descent (VND) algorithm. The basic principle of the VND algorithm is to determine the initial solution and repair the solution using the neighborhood operator. In this article, the initial solution is determined by the insertion heuristic algorithm and makes improvements using six neighborhood operators. The optimum condition of the VND algorithm is reached if there is the most optimal solution for all operators, and the solution has converged to a minimum value. The application of the VND algorithm on VRPSDP was made using Borland Delphi 7 software. The VND algorithm application program was tested using standardized data sets with 100 and 200 customers, then compared multiple routes, solutions, and gaps (%). For many customers, 100 and 200 on the two types of data sets, there are many more VND algorithm routes than the tabu search algorithm. The results of the distance solution for the four data sets show that the tabu search algorithm is smaller than the VND algorithm. The type c data set has a smaller gap compared to the type r data set for both 100 and 200 customers to test tabu search algorithms and VND algorithms.

1. Introduction
In everyday life, problems are often found regarding the distribution of goods. Distribution is an activity to market products from producers to consumers. Distribution activities are one of the important factors in a company to make a profit. One way to obtain these benefits is to minimize distribution costs. Distribution costs can be minimized by choosing the right distribution route. Distribution problems can be solved using one of the disciplines, namely mathematics. One of the branches of mathematics that can solve distribution problems is graph theory, especially in the Vehicle Routing Problem (VRP) material. Some studies that discuss VRP problems, for example, can be seen in [12], [17], and [23].

The VRP is a problem to find the best route for a vehicle that will leave one or more depots for delivery and return of goods, where customer locations are widely spread. The solution expected from the completion of VRP is that it can provide significant benefits by reducing costs. In its development, VRP has many variants with various obstacles, for example, VRPTW can be seen in [3], [5], [7], the CVRP problems can be seen in [7], the CVRPTW can be seen in [2], [11], [22], and the Vehicle...
Routing Problem with Simultaneous Delivery and Pickup (VRPSDP) [13]. The VRPSDP problem is a VRP variant that has special problems, namely that requests and returns are carried out simultaneously. This results in fluctuations in current vehicle loads resulting in increased difficulty in checking the feasibility of a solution. Therefore, the main aspect is checking the current vehicle load on each customer because the vehicle capacity cannot be overcrowded. The purpose of VRPSDP is to minimize the total distance from the route that vehicles traverse when distributing goods without violating capacity constraints [25].

The VRPSDP problem can be solved by several algorithms, both heuristic, and metaheuristic. Several studies related to VRPSDP and the algorithms used include the Artificial Bee Colony (ABC) algorithm discussed by [20], the Multiple Neighborhood Guided Local Search (MN_GLS) algorithm discussed by [25], the Ant Colony System-Variable Neighborhood Search (ACS-VNS) algorithm discussed by [12], the Parallel Simulated Annealing algorithm discussed by [15], and the Effective Ant Colony Optimization (EACO) algorithm discussed by [16]. The tabu search algorithm is efficient [19]. The tabu search algorithm is one type of metaheuristic algorithm that can be used to solve the VRP variant as seen in [14], [4], [6], and [21].

The Variable Neighborhood Descent (VND) algorithm is a metaheuristic algorithm that is applied to several combinatorial optimization problems [20]. The basic principle of the VND algorithm is to determine the initial solution and improve the solution using the neighborhood operator in the VND algorithm. The VND algorithm is based on the idea that the exploration of a given solution should start with a predetermined neighborhood. The neighborhood is a set of solutions from the implementation of a predefined operator. If no improvement is found, then the search should be extended to a different neighborhood. This continued until the search reached the largest available neighborhood. If at some point there is a better solution, then the process is resetting to the neighborhood first. On the other hand, the process stops when all the neighborhood has been explored and no improvement is found [9]. Several studies that use the VND algorithm to solve a problem include the Vehicle Routing Problem with Time Windows [8], Capacitated Vehicle Routing Problem [1], and CVRP, MDVRP, and VRPTW [18].

Based on the description above, in this article, a comparative study of the two algorithms is carried out, namely the VND algorithm and the tabu search algorithm to solve VRPSDP. The implementation of the VND algorithm to solve VRPSDP is designed with the Borland Delphi 7.0 programming language. The VND algorithm application program was tested using standardized data sets with 100 and 200 customers, then compared multiple routes, solutions, and gaps.

2. Research Method
The method used in this research is as follows:
a) Doing a literature study to examine the application of the VND algorithm to VRPSDP problems
b) Implementing the VND algorithm on VRPSDP into a computer program using Borland Delphi 7, by inputting data, processing the solution with the algorithm, and then obtaining the final solution of the algorithm calculation
c) Testing the program using the VRPSDP data set with 100 customers and 200 customers

3. Result and Discussion
3.1. VND Algorithm on VRPSDP
In its completion, the VND algorithm on VRPSDP has two steps are used, namely finding an initial solution and improving customer position by using the neighborhood operator in the VND algorithm. In the initial solution search stage, the Insertion Heuristic algorithm will be used, then the customer position will be corrected to find the minimum total distance without breaking capacity constraints. In improving customer position, six neighborhood operators will be used. The six neighborhood operators are used because they provide optimal results in the process. From the six neighborhood operators, several solutions will be obtained. The best solution will be selected and will be compared with the original solution. If all operators have been carried out and the corrective solution obtained
converges to a minimum value, then the optimum condition of the VND algorithm has been achieved. The steps for the VND algorithm on VRPSDP are:

I. Finding Initial Solutions

At this stage, the initial solution to the VRPSDP problem will be sought. The initial solution will be a temporary route. The initial solution generation in this study uses an Insertion Heuristic algorithm. The steps for the Insertion Heuristic algorithm [10] are:

Phase I: Form a route with \( d_i \) (delivery customer \( i \)) \( \geq p_i \) (pickup customer \( i \))

1. Establish \( PR_k = (0,0)_k \) initial routes for each vehicle \( k \) \( (f_k) \)
2. For each customer \( i \) with \( d_i \geq p_i \), sort the results from \( d_i - p_i \) and list d-customers from biggest to smallest
3. Starting from the top d-customer list, select customer \( i \). Find the \((h,j)\) side and paste \( i \) between \((h,j)\) on the existing \( PR_k \) route, so that you will get a new side, \((h_i)\) \( i \), and \((i,j)\). Calculate the value of the insertion distance in \((c_{hi} + c_{ij} - c_{nj})\), where \( c \) is the distance. Then, check the capacity constraints on the customer distance insertion \( i \). Select route \( PR_k \) which has a minimum insertion distance and does not violate capacity constraints. If it does not violate the capacity constraints and has the same distance insertion value, then any \( PR_k \) route can be chosen. If all distance insertions violate capacity constraints, then insert customer \( i \) on the new \( PR_k \) route.
4. If the customer \( i \) has entered the route, delete customer \( i \) from the d-customer list. Then repeat step 3 until there are no customers on the d-customer list.

Phase II: Insert Customer with \( d_i < p_i \)

For each customer, \( i \) with \( d_i < p_i \), insert customer \( i \) into all \( PR_k \) routes that have been formed in phase I. Calculate the value of insertion distance using \((c_{hi} + c_{ij} - c_{nj})\). Check capacity constraints on all insertions. Select minimum insertion distance and do not violate capacity constraints. If all distance insertions violate capacity constraints, then insert customer \( i \) on the new \( PR_k \) route. Customer \( i \) may only be inserted once. The process stops when all \( i \) customers have entered the route.

II. Improve Customer Position

In the previous stage, an initial solution was obtained. The initial solution is not yet certain as a result that has a minimum total distance. Therefore, the customer position will be improved by using several neighborhood operators that aim to find other possible outcomes that are better than the initial solution. Some neighborhood operators in the VND algorithm that will be used to improve customer positions are swap 1-1, swap 2-1, swap 2-2, insertion, exchange, and 2-opt. The total distance obtained from the improvement of the customer’s position will be compared with the total distance obtained from the initial solution.

Improvement of customer position is done by exchanging customers between routes (inter-route) and exchanging customers on their routes (intra-route). Based on [6], the exchange of customer positions is done by exchanging inter-route first, namely swap operators 1-1, swap 2-1, and swap 2-2 in order. Then an intra-route exchange is performed with the insertion, exchange, and 2-opt operators in order.

The process of improving the customer’s position begins by preparing a list of neighborhoods that will be used in an order that is \( DN_1 = swap \ 1-1, \ swap \ 2-1, \ swap \ 2-2, \ insertion, \ exchange, \ 2-opt \) and calculating the total distance of the initial solution \( f(b) \). The neighborhood list will be given \( DN_1 \) to \( DN_6 \) notation in sequence. Repair starts from \( DN_1 \). Repair with \( DN_1 \) using \( f(b) \) initial solution. The notation for the result of improvement is \( f(b^{'}) \). Then calculate the total distance after fixing \( f(b^{'}) \). If \( f(b^{'}) < f(b) \) is on \( DN_1 \), then \( b^{'}) \) will be the new \( b \) and will be fixed again with \( DN_1 \). If \( f(b^{'}) \geq f(b) \), then it will be fixed with the next \( DN_2 \), \( DN_2 \). If the improvement \( DN_2 \) produces \( f(b^{'}) < f(b) \), then \( b^{'}) \) will become a new \( b \) and will be repaired again with \( DN_1 \). If repair \( DN_2 \) produces \( f(b^{'}) \geq f(b) \), it will be fixed with \( DN_3 \). The workmanship is done in the same way.
up to $DN_k$. All improvements must pay attention to capacity constraints. The repair process stops when it has passed $DN_k$ and has reached its optimum condition.

Simply stated, the steps of the VND algorithm in VRPSDP are as follows:

1. Finding the initial solution ($b$) with the Insertion Heuristic algorithm
2. Prepare a list of neighborhood $DN_x$ and calculate $f(b)$ from the initial solution
3. Make improvements with $DN_x$ in order, $x$ starts from 1
4. The process below will be repeated until $x = 6$
   a. Fix $b$ with $DN_x$. Obtained improvements $b'$
   b. Look for the value ($f(b')$).
   c. If ($f(b') < f(b)$) then $b' = b$ so $f(b') = f(b)$ and $x = 1$.
   If ($f(b') \geq f(b)$) then $x = x + 1$.

3.2. Tabu Search Algorithm on VRPSDP

The Tabu Search algorithm is a metaheuristic algorithm that has better performance when compared to other approaches and has a very good solution [20]. Tabu Search Algorithm is an algorithm that uses the initial solution as the basis for finding a solution to be improved by looking for different neighborhoods. The following are the general steps for the Tabu Search algorithm on VRPSDP that have been researched by [14]:

Step 0 (Initialization)

Generating initial solutions using one of the following heuristic procedures, namely: IGR1 (Independent Grouping and Routing 1), IGR2 (Independent Grouping and Routing 2), SGR1 (Simultaneous Grouping and Routing 1), SGR2 (Simultaneous Grouping and Routing 2).

Step 1 (Initial phase)

The Tabu Search algorithm starts from the current initial solution and works on as many as initial iterations.

Step 2 (Intensification phase)

Perform the Tabu Search algorithm intensification procedure starting from the best solution found in Step 1. The original taboo tenure value is divided in half during intensification and done as many as later iterations.

Step 3 (Diversification phase)

Perform the Tabu Search algorithm diversification procedure starting from the solution obtained in Step 2. The taboo tenure values used in the previous phase are maintained during the diversification phase. The algorithm is done as many as $d_{iter}$ iterations.

Step 4 (Standard phase)

Repeat the Tabu Search algorithm starting from the solution obtained in Step 3. Deleting and inserting the sides of the original taboo tenure values will be reconstructed. The algorithm is done as many as $inter_{iter}$ iterations.

Step 5 (Interactive phase)

Repeat Steps 2-4 until one of the stopping rules is reached.

Step 6 (Re-initialization)

Generate new initial solutions by utilizing one of the heuristics (IGR1, IGR2, SGR1, SGR2) that have not been used so far. Repeat Steps 1–5 until all four heuristic procedures are used to generate the initial solution.

Stop Rules

- No displacement was fulfilled
- The maximum number of iterations for the appropriate phase has been reached (Steps 1-4). The maximum number of iterations is the same for each re-initialization (Steps 2-4).

3.3. Implementation of VND algorithm on VRPSDP

The implementation of the VND algorithm on VRPSDP has been made using a programming language Borland Delphi 7. The input contained in the program is the position of the point, the
distance between points, customer requests, and returns, and vehicle capacity. The output contained in the program is in the form of routes that have been completed using the VND algorithm on VRPSDP and the output in the form of images of the final solution that has been obtained. The following is an example display of VRPSDP with 6 customers who are completed using the program that has been created. The display of the VRPSDP sample program with 6 customers can be seen in Figure 1, Figure 2, Figure 3, Figure 4, and Figure 5 below.

![Graph model](image1.png)

**Figure 1.** Graph model

| Point Input | Distance Input | Parameter | Result | Resulted Graph |
|-------------|----------------|-----------|--------|----------------|
| 0           |                | 1         | 19     | 57             | 51           | 49          | 4          | 12          |
| 1           | 19             | 51        | 10     | 53             | 25           | 53          |            |             |
| 2           | 57             | 51        | 49     | 27             | 30           | 10          |            |             |
| 3           | 51             | 10        | 49     | 50             | 11           | 38          |            |             |
| 4           | 48             | 53        | 27     | 50             | 68           | 9           |            |             |
| 5           | 4              | 25        | 30     | 11             | 69           |             |            | 54          |
| 6           | 12             | 53        | 10     | 38             | 9            | 94          |            |             |

**Figure 2.** Distance Input
**Figure 3** Demand, Return, and Capacity Input

| Customer | Delivery |
|----------|----------|
| 1        | 495      |
| 2        | 540      |
| 3        | 325      |
| 4        | 290      |
| 5        | 300      |
| 6        | 245      |

**Capacity**

800

**Figure 4.** Demand, Return, and Capacity Input

| Customer | Pickup |
|----------|--------|
| 1        | 450    |
| 2        | 500    |
| 3        | 300    |
| 4        | 300    |
| 5        | 315    |
| 6        | 250    |

Customers with \(d = p\) are
- Customer 2 has a difference of 40
- Customer 1 has a difference of 35
- Customer 3 has a difference of 25

Customers with \(d < p\) are
- Customer 4 has a difference of -10
- Customer 5 has a difference of -15
- Customer 6 has a difference of -5

**Initial Solution**
- Route 1 is 0-2-6-0 with a distance of 79.00 km
- Route 2 is 0-1-5-0 with a distance of 48.00 km
- Route 3 is 0-3-4-0 with a distance of 153.00 km
  - The total distance is 277.00 km

**Repair results with swap 1-1**
- Route 1 is 0-2-6-0 with a distance of 79.00 km
- Route 2 is 0-3-5-0 with a distance of 66.00 km
- Route 3 is 0-1-4-0 with a distance of 121.00 km
  - Total distance 266.00 km

**End Result with VND**
- Route 1 is 0-2-6-0 with a distance of 79.00 km
- Route 2 is 0-3-5-0 with a distance of 66.00 km
- Route 3 is 0-1-4-0 with a distance of 121.00 km
  - Total distance 266.00 km
Based on the results of the completion of the program, the final result is obtained after experiencing repairs 1 time, where the final result is three routes. The first route is 0-2-6-0 with a distance of 79 km, the second route is 0-3-5-0 with a distance of 66 km, and the third route is 0-1-4-0 with a distance of 121 km. The three routes have a total distance of 266 km.

The program was tested using the VRPSDP data set which can be found at http://users.ntua.gr/ezach/ [24]. The VRPSDP data set used is the result of changing the data set on VRPTW. The data set used consists of 2 types, namely type c (grouped customer locations) and type r (customer locations are built arbitrarily). The trials were conducted with 4 different data sets, namely c101, r101, c1_2_1, and r1_2_1. Each data set has a vehicle capacity of 200 and the number of customers, respectively, 100, 100, 200, 200. Below are the results of the test results for the four data sets.

- Test data set c101 (100 customers)
  Obtained the final results of testing the c101 data set with 18 routes and a total distance of 1329.02 km. Figure 6 below is an example of displaying the final result of the data set after testing.
In the same way, the results are obtained:

- Test data set r101 (100 customers)
The final results of the r101 data set trial are 13 routes and a total distance of 1212.74 km.

- Test data set c1_2_1 (200 customers)
The final results of the c1_2_1 data set trial are 30 routes and a total distance of 4170.39 km.

- Test data set r1_2_1 (200 customers)
The final results of the r1_2_1 data set trial are 26 routes and a total distance of 4114.07 km.

Furthermore, the results of the trial data set were compared with the results of the study using the Tabu Search algorithm by [11], as shown in Table 1.

| Data Set  | Tabu Search Algorithm | The VND algorithm |
|-----------|-----------------------|-------------------|
|           | Number of Routes | Solution | Number of Routes | Solution | Gap (%) |
| c101      | 17                  | 1259.79   | 18              | 1329.02  | 5.5     |
| r101      | 12                  | 1042.62   | 13              | 1212.74  | 16.3    |
| c1_2_1    | 29                  | 3792.62   | 30              | 4170.39  | 9.9     |
| r1_2_1    | 23                  | 3447.20   | 26              | 4114.07  | 19.3    |

Based on the table above, it can be seen that the four data sets tested with the VND algorithm have a solution with a total distance greater than the tabu search algorithm solution. Besides, the VND algorithm solution has more total routes than the tabu search algorithm solution. The possible causes of the greater total distance and more total routes in the VND algorithm are variations of the neighborhood operator used and how to choose it. In the inter-route improvement used in the VND algorithm, the neighborhood operator has one type of improvement, namely the swap type. Meanwhile, the inter-route improvement used in the Tabu Search algorithm consists of three types, namely relocation, interchange, and crossover.

The advantage of the VND algorithm used in this study is that it does not require parameters in the calculation, while the tabu search algorithm uses several parameters in the calculation. Also, the VND algorithm always produces the same solution for every calculation, so it doesn't take a long time to solve the problem.

In Table 1 above, it can be seen that for the case of 100 customers, the c101 data set has a smaller gap value than the r101 data set. For the case of 200 customers, data set c1_2_1 has a smaller gap value than data set r1_2_1. Thus, in both cases, it can be said that the c type data set has a smaller gap value than the r type data set.

4. Conclusion
The VND algorithm in VRPSDP has two steps, namely finding the initial solution using the insertion heuristic algorithm and improving the customer's position using the neighborhood operator in the VND algorithm. Customer position improvement is done to find the minimum total distance without violating capacity constraints. In improving customer position, six neighborhood operators will be used. Of the six neighborhood operators, the best solution will be chosen and will be compared with the initial solution. If all operators have been carried out and the corrective solution obtained converges to a minimum value, then the optimum condition of the VND algorithm has been achieved. The implementation of the VND algorithm on VRPSDP has been made using the Borland Delphi 7 programming language. The input contained in the program is the position of the point, the distance between points, customer requests, and returns, and vehicle capacity. The output contained in the program is in the form of routes that have been completed using the VND algorithm on VRPSDP and the output in the form of images of the final solution that has been obtained. The program has been tested using 4 VRPSDP data sets, namely c101, r101, c1_2_1, and r1_2_1 with the number of

---

Table 1. Data Set Comparison

| Data Set  | Tabu Search Algorithm | The VND algorithm |
|-----------|-----------------------|-------------------|
|           | Number of Routes | Solution | Number of Routes | Solution | Gap (%) |
| c101      | 17                  | 1259.79   | 18              | 1329.02  | 5.5     |
| r101      | 12                  | 1042.62   | 13              | 1212.74  | 16.3    |
| c1_2_1    | 29                  | 3792.62   | 30              | 4170.39  | 9.9     |
| r1_2_1    | 23                  | 3447.20   | 26              | 4114.07  | 19.3    |
customers respectively 100, 100, 200, 200. Based on the test results of the four data sets, the solution generated by the Tabu Search algorithm has a smaller total distance compared to the VND algorithm. Besides, the VND algorithm solution has more total routes than the tabu search algorithm solution. The trial results also show that in both cases 100 and 200 customers, the c type data set has a smaller gap value than the r type data set.

Acknowledgments
This article is a part of research funded by PNBP, contract no: 4.3.474/UN32.14.1/LT/2020, entitled ”Variable Neighborhood Descent (VND) Algorithm on Vehicle Routing Problem With Simultaneous Delivery and Pickup (VRPSDP) and Its Implementation”. Thank you to Universitas Negeri Malang for funding the research.

References
[1] Amous M, Toumi S, Jarboui B, and Eddaly M 2017 A Variable Neighborhood Search Algorithm for the Capacitated Vehicle Routing Problem,” Electronic Notes in Discrete Mathematics 58 231–238
[2] Cardoso P J S, Schütz G, Mazayev A, Ey E, and Corrêa T 2015 A Solution for a Real-time Stochastic Capacitated Vehicle Routing Problem with Time Windows Procedia Computer Science 51 2227–2236
[3] Chiang W C and Cheng C Y 2017 Considering the Performance Bonus Balance in the Vehicle Routing Problem with Soft Time Windows Procedia Manufacturing 11 2156–2163
[4] Cordeau J F, Laporte G and Mercier A 2001 A unified tabu search heuristic for vehicle routing problems with time windows Journal of the Operational Research Society 52 (8) 928–936
[5] Dhahri A, Mjirda A, Zidi K, and Ghedira K 2016 A VNS-based Heuristic for Solving the Vehicle Routing Problem with Time Windows and Vehicle Preventive Maintenance Constraints Procedia Computer Science 80 1212–1222
[6] Escobar J W, Linfati R, Toth P and Baldoquin M G 2014 A hybrid Granular Tabu Search algorithm for the Multi-Depot Vehicle Routing Problem J Heuristics 20 (5) 483–509
[7] El-Sherbeny N A 2010 Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods Journal of King Saud University - Science 22 (3) 123–131
[8] Harzi M and Krichen S 2017 Variable Neighborhood Descent for Solving the Vehicle Routing Problem with Time Windows Electronic Notes in Discrete Mathematics 58 175–182
[9] Herrán A, Colmenar J M, and Duarte A 2019 A Variable Neighborhood Search Approach for the Hamiltonian p-median problem Applied Soft Computing 80 603–616
[10] Kalayci C B and Kaya C 2016 An Ant Colony System Empowered Variable Neighborhood Search Algorithm for the Vehicle Routing Problem with Simultaneous Pick-up and Delivery Expert Systems with Applications 66 163–175
[11] Khachai M Y and Ogorodnikov Y Y 2019 Polynomial-Time Approximation Scheme for the Capacitated Vehicle Routing Problem with Time Windows Proc. Steklov Inst. Math 307 51–63
[12] Korablev V, Makeev I, Kharitonov E, Tshukin B, and Romanov I 2016 Approaches to Solve the Vehicle Routing Problem in the Valuable Delivery Domain Procedia Computer Science 88 487–492
[13] Lai C, Chen C, and Ma Y 2008 Vehicle Routing Problem With Simultaneously Deliveries And Pickups Journal of Far East University 25 (3) 475–484
[14] Montané F A T and Galvão 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
[15] Mu D, Wang C, Zhao F, and Sutherland J W 2016 Solving Vehicle Routing Problem with Simultaneous Pick-up and Delivery using Parallel Simulated Annealing Algorithm IJSTL 8 (1) 81-106
[16] Sayyah M, Larki H, and Yousefikhoshbakht M 2016 Solving the Vehicle Routing Problem with Simultaneous Pickup and Delivery by an Effective Ant Colony Optimization *Journal of Industrial Engineering and Management Studies (JIEMS)* 3 (1) 15-38

[17] Sawik B, Faulin J, and Pérez-Bernabeu E 2017 A Multicriteria Analysis for the Green VRP: A Case Discussion for the Distribution Problem of a Spanish Retailer *Transportation Research Procedia* 22 305–313

[18] Satyananda D and Wahyuningsih S 2019 VND in CVRP, MDVRP, and VRPTW cases *J. Phys.: Conf. Ser.* 1320 012025

[19] Shi Y, Boudouh T, and Grunder O 2018 An Efficient Tabu Search based Procedure for Simultaneous Delivery and Pick-up Problem with Time Window *IFAC-Papers Online* 51 (11) 241–246

[20] Simsir F and Ekmekci D 2019 A Metaheuristic Solution approach to Capacitated Vehicle Routing and Network Optimization *Engineering Science and Technology, an International Journal* 22 (3) 727-735

[21] Tamashiro H, Nakamura M, Okazaki T, and Kang D 2010 A Tabu Search Approach combined with An Extended Saving Method for Multi-depot Vehicle Routing Problems with Time Windows *Biomedical Soft Computing and Human Sciences* 15 (1) 29-37

[22] Tebaldi L, Murino T, and Bottani E 2020 An Adapted Version of the Water Wave Optimization Algorithm for the Capacitated Vehicle Routing Problem with Time Windows with Application to a Real Case Using Probe Data *Sustainability* 12 (9) 3666

[23] Toffolo T A M, Christiaens J, Malderen S V, Wauters T, and Berghe G V 2018 Stochastic local search with learning automaton for the swap-body vehicle routing problem *Computers & Operations Research* 89 68–81

[24] Zachariadis E E and Kiranoudis C T 2011 A local search metaheuristic algorithm for the vehicle routing problem with simultaneous pick-ups and deliveries *Expert Systems with Applications* 38 (3) 2717–2726

[25] Zhu H, Feng J, and Li H 2017 MN_GLS for VRP with Simultaneous Delivery and Pickup *Expert Systems with Applications* 28 (6) 12