AUT Journal of Electrical Engineering

AUT J. Elec. Eng., 55(3) (Special Issue) (2023) 323-332
DOI: 10.22060/eej.2022.21404.5474

Using a Tuning Parameter to Compromise Computation Time and Shipping Cost in an MDVRP

M. A. Zajkani, R. Rahimi Baghbadorani, M. Haeri*

Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran

ABSTRACT: Vehicle routing in last-mile delivery plays a decisive role in the new world of people's lifestyles. At present, a growing number of people order their needs online, and this forces companies to employ innovative delivery logistics to reduce their last-mile shipping costs. The goal is to minimize the cost of travel that depends on the Euclidean distance between customers. Companies require solving vehicle routing problems (VRP) in a reasonable time. In this paper, a new approach is introduced that solves the multi-depot vehicle routing problem (MDVRP) in real-time. We propose a new method by clustering and decomposing the main problem into smaller ones using a tuning parameter α. This approach could reduce the solution time noticeably (up to 95%) while the shipping cost is still reasonable.

Review History:

Received: May, 19, 2022
Revised: Nov. 17, 2022
Accepted: Dec. 20, 2022
Available Online: Dec. 01, 2023

Keywords:

Vehicle routing problem
Tuning parameter
Computation time
Complexity
Clustering

1- Introduction

Nowadays, most people prefer to save their time and energy by ordering their daily stuff online and are reluctant to shop in traditional ways. In conformity, many companies have been developed to respond to this tendency with on-site delivery, where shipping is the primary determinant [1]. Statistics show that the final cost in this business is mostly influenced by transportation and logistics processes [2]. Therefore, companies' attention to routing problems, can help them to become more cost-effective and increase their revenue. VRP is classified into multiple categories [3]. Fig. 1 shows these categories, their relation, and a short description. The type of VRP is determined by the considered circumstances, some of which are listed in Table 1. According to the scenarios considered, different types of VRP are defined (Table 2) [4, 5].

One of the most practical types of VRP is the MDVRP, where any customer is serviced only once with a set of vehicles that belongs to non-identical depots and eventually returns to their depots. Considering the demand of the customers, the number of vehicles in each depot, and their capacity, each customer is assigned a vehicle.

The main idea is to minimize the shipping cost of the vehicles during the servicing. So far, different algorithms have been suggested to solve an MDVRP [6]. With an increase in the number of customers and/or depots, the complexity of MDVRP, which itself is an NP-hard problem, increases as well. In many companies, the solution time is an essential factor for online servicing. Therefore, the employed algorithms should be able to solve the problem in real-time. Available solutions for high dimensional MDVRP are mostly obtained via heuristic or meta-heuristic approaches where many decisive parameters should be appropriately selected. Otherwise, the solution would not be reasonable, and the answer would be very different from the best answer so far, in addition to the answer error, it may take a long time to solve. Genetic algorithm [7], particle swarm optimization [8], bat algorithm [9], grey wolf optimization [10], ant colony optimization [11], flower pollination algorithm [12], and multi-verse optimization [13] can be mentioned as the main and widely used methods. On the other hand, exact solution-based approaches are inefficient solve problems with large dimensions.

To use the exact solution, the problem decomposes into sub-problems (first stage of decomposition) by a capacitated clustering approach [14]. In other words, each sub-problem considers a single depot that identifies its customers and serves them. According to this method, the main problem decomposes into many sub-problems with a single depot that serves its allocated customers. However, the number of vehicles and customers in each sub-problem may still cause more complexity than the exact solution method could

*Corresponding author's email: haeri@sharif.ir

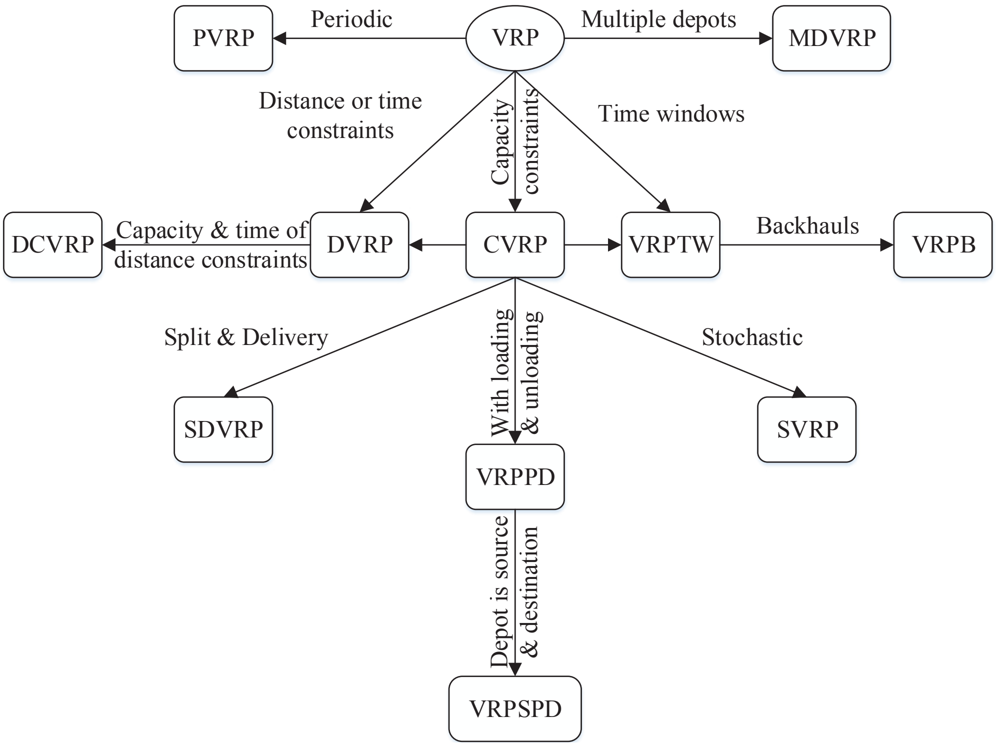

Fig. 1. The important types of VRP and their relation.

Table 1. A number of the taxonomy of VRP.

Services	Single depot or multi-depots.
Number of vehicles	Exactly v vehicles, up to v, or unlimited vehicles.
Time of Windows type	Restriction on customers, roads, etc.
Capacity consideration	Capacitated vehicles or incapacitated vehicles.
Vehicle homogeneity (capacity)	Similar vehicles, load-specific vehicles, heterogeneous vehicles, or customer-specific vehicles
Number of stops on route	Known or partially known.
Load splitting constraint	Splitting allowed or splitting not allowed
Customer service demands quantity	Deterministic, stochastic, or unknown
Request time of the new customer	Deterministic, stochastic, or unknown
On-site service/waiting times	Deterministic, time-dependent, vehicle time-dependent, stochastic, or unknown
Time window structure	Soft time window, strict time window, or a mix of both
Time horizon	Single-period or multi-period
Backhauls	Nodes request simultaneous pickup and deliveries, or nodes request either line haul or backhauls service, but not both.

Table 2. Important types of VRP.

VRP type	Abbreviation	Description
Capacitated VRP	CVRP	VRP with the additional constraint that every vehicle must have a uniform capacity of a single commodity.
Multiple Depot VRP	MDVRP	Several depots from which it can serve its customers.
Periodic VRP	PVRP	VRP is generalized by extending the planning period to M days.
Split delivery VRP	SDVRP	It is allowed that a customer can be served by different vehicles.
Stochastic VRP	SVRP	One or several components of the problem are random.
VRP with Backhauls	VRPB	Customers can demand or return some commodities.
VRP with Pickup and Delivery	VRPPD	It is a VRP in which the possibility that customers return some commodities is contemplated.
VRP with Satellite Facilities	VRPSF	Satellite replenishment allows the drivers to continue making deliveries until the close of their shift without necessarily returning to the central depot.
VRP with Time Window	VRPTW	Every customer must be serviced in a special time range.
Dynamic VRP	DVRP	One or several components of the problem are dynamic.
Distance-constrained Capacitated VRP	DCVRP	It's a kind of CVRP addition to distance-constrained.
VRP with Simultaneous Pickup and Delivery	VRPSPD	Both pickup and delivery tasks simultaneously occur at various customer locations, and the depot is the source and destination.

handle in real-time. In this case, the mentioned sub-problems are decomposed again into a subdivided problem (second phase of decomposition). These problems contain one or more vehicles with their allocated customers. At the end of the method, each subdivided problem is solved by the exact method [15] using binary linear programming [16]. This way, the proposed method could find a reasonable solution in admissible time; of course, it should be added that the answer may have a little more error than other methods, but the solution time will be much less.

Another contribution of this paper is to define a tuning parameter α in the second phase of decomposition to compromise between the computation time and shipping cost. This parameter takes values between 0 and 1. When the computation time is crucial, this parameter should be large, and when the shipping cost is more important, it should be small. Each depot has its α, and the operator has the freedom to decide according to the situation. For example, the problem will be less complicated when the number of customers in a subdivided problem is low. In this case, the solution time will not be as important as the total shipping cost.

The rest of this paper is arranged as follows. The description and formulation of the MDVRP are presented in Section 2. The proposed approach and its details are provided in Section 3. The optimization procedure performed to solve the defined subdivided problems is described in Section 4. Simulation results are presented in Section 5. Eventually, conclusions and future works are given in Section 6.

2- MDVRP

The problem statement and formulation are presented in the two following subsections.

2- 1- Problem statement

The main idea of the MDVRP is to minimize the total shipping costs due to fuel consumption, travel time, and covered distance. Every customer should be serviced by one vehicle considering the demand and capacity of vehicles. Each vehicle starts servicing from a depot and returns to the same depot after completing the servicing. To minimize the shipping cost, the MDVRP designs a set of vehicle routes serving all customers, considering the maximum number of vehicles per depot and vehicle capacity. The following notations are used to define an MDVRP [17].

Let $G=(V, E)$ be a graph where $V=\{v_1, v_2, \cdots, v_n\}$ is the set of vertices and $E=(v_i, v_j): \forall v_i, v_j \in V, i \neq j$ is the set of edges. The vertices' set V is the union of two sets, V^{DEP} and V^{CST}, where $V^{DEP}=\{v_1, v_2, \cdots, v_d\}$ represents the set of depots, and V^{CST} is the set of customers. Thus, $V=\{v_1, v_2, \cdots, v_d, v_{d+1}, \cdots, v_n\}$. For each depot $v_i \in V^{DEP}$ a subset of customers $v_i \in V^{CST}$ is assigned. Each vertex $v_j \in V^{CST}$ is characterized by a nonnegative demand c_j (demand of customer j), and each edge (v_i, v_j) is associated with a cost f_{ij}. In each depot $v_i \in V^{DEP}$, there are m vehicles with capacity C. By considering Euclidean distance between nodes i and j as f_{ij}, representing the real cost of air and traffic pollution, we formulate the

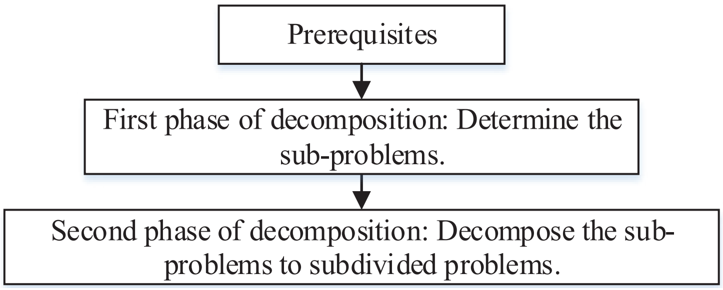

Fig. 2. The main steps of the proposed approach.

MDVRP in the next section.

2- 2- Integer-based model

We present an integer-based model for MDVRP. If b_{ij}^{k} equals 1, then vehicle k visits node j immediately after node i . J is the total cost, which should be minimized.

$$\min J = \sum_i \sum_j \sum_k b_{ij}^k f_{ij}, \text{ subject to} \tag{1}$$

$$\sum_i \sum_j b_{ij}^k c_j \le C, \forall k \in V \tag{2}$$

$$\sum_k \sum_j b_{ij}^k = 1, \forall i \in \{2,3,\cdots,N\} \tag{3}$$

$$\sum_k \sum_i b_{ij}^k = 1, \forall j \in \{2,3,\cdots,N\} \tag{4}$$

$$\sum_k \sum_i b_{ii}^k = 0 \tag{5}$$

$$\sum_k \sum_i b_{i1}^k = \sum_k \sum_j b_{1j}^k \tag{6}$$

$$\sum_k \sum_j b_{1j}^k \ge 1 \tag{7}$$

The cost function is given in (1). Constraint (2) explains that the total load of each vehicle should not exceed its capacity. Constraint (3) implies that every customer must be serviced by a vehicle. Constraint (4) states that there is only one path for each node, except $v_j \in V^{CST}$. Constraint (5) prevents one node return to itself (5). Constraint (6) states that each vehicle that exits any depot must finally return to the same depot and finally (7) expresses that at least one vehicle is used by each depot.

3- Decomposition methods

To reduce the complexity of MDVRP and, therefore, its solution time, two consecutive decompositions are implemented. In the first step, the problem is clustered into sub-problems, each including only one depot. In the second step, these sub-problems are further decomposed into subdivided problems using the tuning parameter to influence the complexity. The second step of decomposition causes a reduction in vehicle and customer numbers in the last routing problem. The subdivided problems are solved by the exact linear integer programming defined in Section 4. The proposed approach is shown in Fig. 2.

3- 1- Capacitated clustering algorithm (first step)

In the first step of decomposition, a capacitated clustering problem (CCP) is solved to allocate each customer to one depot by considering the demand of the customer, the number of vehicles, the capacity of the vehicle, and the shipping cost between nodes. In this problem, there are n customers with their known demands distributed in (x, y) coordinates. These customers are grouped to form k clusters. Each cluster has $n_1, n_2, \cdots, n_k$ number of customers such that $\sum_{j=1}^{k} n_j = n$. The problem is given with a set of customers $(r_1, r_2, r_3, \cdots, r_n)$, coordinates $((x_1, y_1), (x_2, y_2), (x_3, y_3), \cdots, (x_n, y_n))$, demands $(d_1, d_2, d_3, \cdots, d_n)$, and capacity (C) where $r_i \in R$ is the set of customers who are distributed in the Euclidean plane (x_i, y_i). Demand d_i and capacity (C) of the cluster are positive integers. Euclidean distance matrix $\text{cost} = [\text{cost}_{ij}]$ is determined based on the number of customers (i) and depots (j) as

$$\text{cost}_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}.$$

Cost is an $m \times n$ matrix, where m is the number of customers, and n is the number of depots. Let $Z = [z_{ij}]$

be a binary matrix such that

$$z_{ij} = \begin{cases} 1 & \text{if customer } i \text{ is assigned to cluster } j \\ 0 & \text{otherwise} \end{cases}.$$

As an example, consider 6 customers who require to be allocated to 2 depots based on the capacity constraint that, is the customer's demand should be less than or equal to the vehicle capacity. Let $c_1, c_2, c_3, \cdots, c_n$ be the customers distributed in the (x, y) plane. The rows of the binary matrix represent the customers, and its columns represent the clusters. A possible binary matrix could be as follows.

$$Z^T = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

The objective is to find Z that minimizes

$$\sum_{j=1}^{k} \sum_{i=1}^{n} \text{cost}_{ij} \, z_{ij}, \text{ subject to} \tag{8}$$

$$\sum_{j=1}^{k} z_{ij} = 1, \, i = 1,2,3, \cdots, n, \tag{9}$$

$$\sum_{i=1}^{n} d_i z_{ij} \leq C, \, j = 1,2,3, \cdots, k, \tag{10}$$

where cost_{ij} represents the cost of the closeness of customer i to cluster j (i.e., the cost or time of travel of customer i in cluster j). Minimizing the objective function in (8) reduces the customers' total assignment cost to the clusters. Constraint (9) assures that customer i is assigned only to a single cluster. Constraint (10) assures that the total demand of customers in a cluster does not exceed the cluster capacity C.

The standard deviation of each row of the cost matrix is calculated, and its value is assigned to the same row of a newly constructed vector P. Thus, P is an $m \times 1$ vector. By sorting the rows of P in descending values, each customer's priority will be determined. Customers with smaller standard deviations will have higher priority. By starting from the customer with the lowest priority, each customer will be allocated to the nearest depot based on constraint (10). If constraint (10) is not satisfied, the selected customer will be assigned to the next nearest depot based on constraint (10). This method has been applied to example P01 of [18], where $n = 50$ customers are going to be allocated to $k = 4$ depots. In this way, the original problem of MDVRP is divided into 4 sub-problems. Fig. 3 shows the results of this clustering.

3- 2- Second level of clustering by integrating the tuning parameter

If the resulting sub-problems are still complex to be solved in real-time, they can be further decomposed into more minor problems. For this step, we use the extended k-means approach [19].

Customers of Depot 1 are clustered using the k-means approach, and results are shown in Figs. 4-6 for three values of tuning parameter α (0, 0.33, 1). In this step, the number

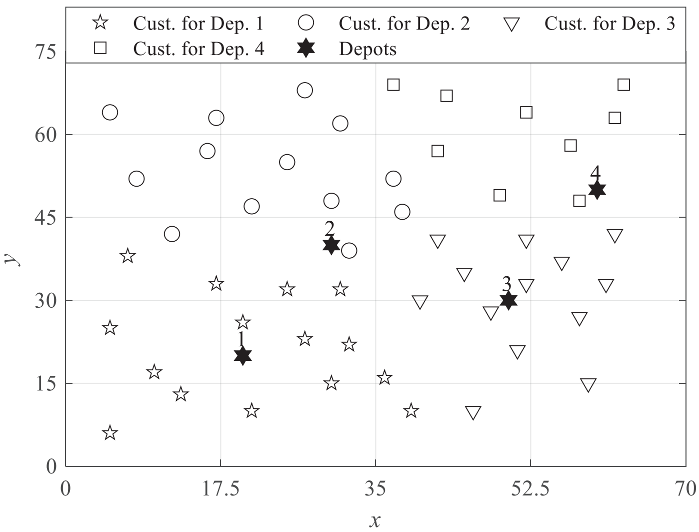

Fig. 3. The first clustering for every depot in P01.

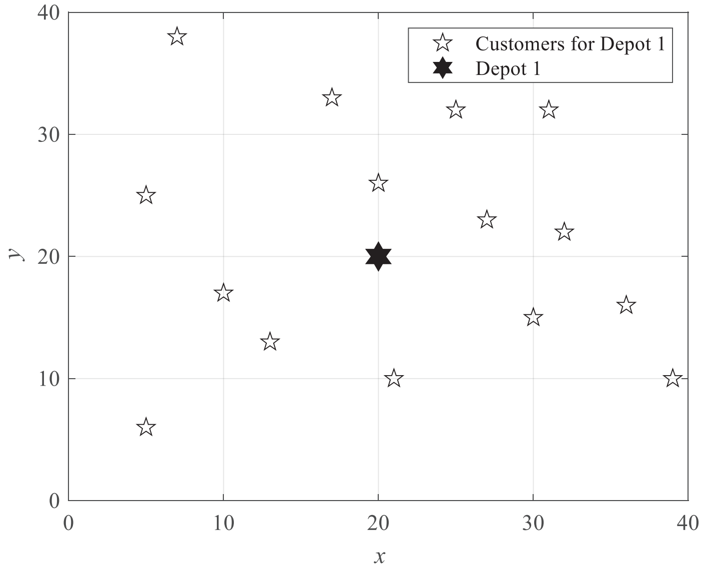

Fig. 4. Clustering with α=0 (single cluster and single optimization problem).

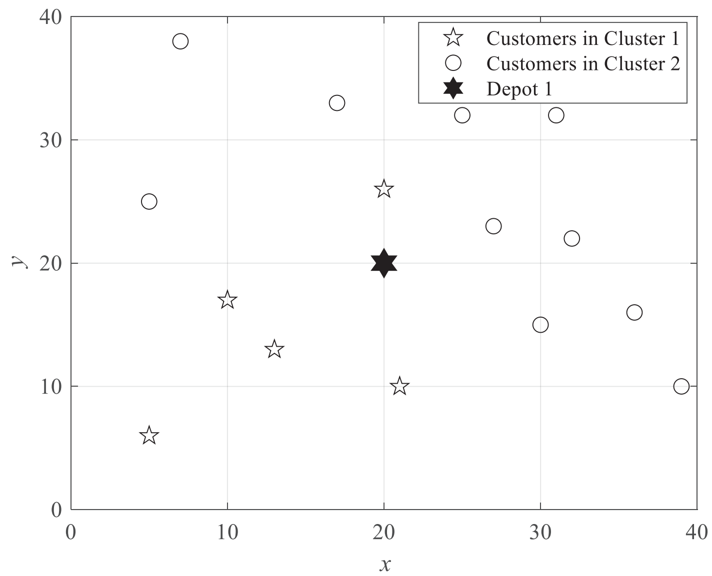

Fig. 5. Clustering with α=0.33 (2 clusters and 2 optimization problems).

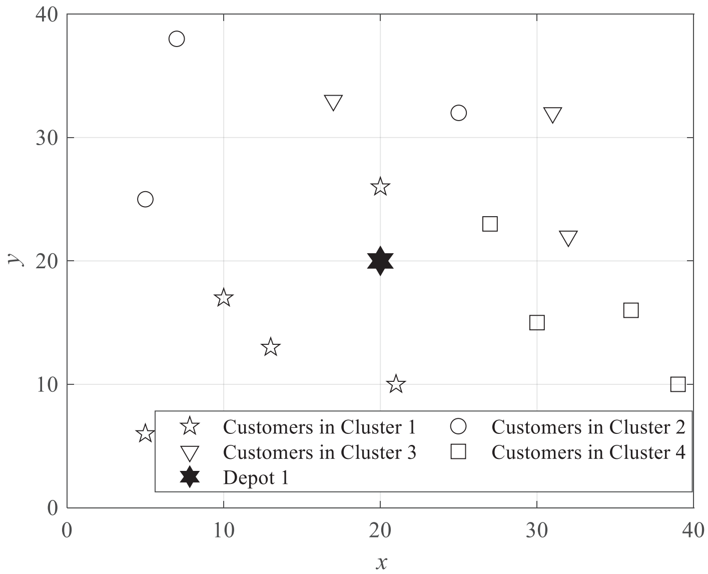

Fig. 6. Clustering with α=1 (4 clusters and 4 optimization problems).

of required vehicles for each depot is determined. As an example, for the second depot in P01, the algorithm declares the need for 3 vehicles. Then, α could assume three values of 0, 0.5, and 1, so vehicles could be clustered in 3 ways. For $\alpha = 0$, we have a single cluster with three vehicles, $\alpha = 0.5$. We have two clusters, one with a single vehicle and another with two vehicles; and $\alpha = 1$, we have three clusters, each including one vehicle. Acceptable values for α depend on the number of vehicles and operator definition.

4- Solving the optimization problems

After transforming the main problem into subdivided problems, these problems are solved by linear integer programming, discussed in Section 2. There are several approaches to solving the defined optimization problems. Due to the complexity and NP-hardness of the MDVRP, the exact methods could not be relied on when the original problem has to be solved, especially when the solutions are required in real-time. In this paper, as mentioned before, linear programming as an exact method is used, which could provide suitable and real-time solutions for low-dimensional problems such as those defined in the second level of decompositions. The main novelty of the present work is to break the original problem into small ones so that the exact methods can be implemented in real-time.

Solving the linear programming discussed before could end up with an improper route loop where vehicles' route passes a depot aside from the beginning and end of the service time. There is no linear constraint to avoid these unsuitable solutions. Adding the following constraint to the problem (1), however, could efficiently reduce the probability of obtaining these solutions.

$$\sum_{k=1}^{v} x_{ij}^{k} + x_{ji}^{k} \leq 1, \quad \forall i,j \in \{2,3,\cdots,N\} \tag{11}$$

Meanwhile, the obtained solutions are checked for improper loops, and in this case, the problem is solved again with an additional constraint that leads to the elimination of these loops. The following lemma is provided to back this claim.

Lemma 1. Let the vertices of G be labeled $v_1, v_2, \cdots, v_n$ and the adjacency matrix of G be A with row/column i of A corresponding to v_i. To have a purely linear algebraic approach, consider $L = D - A$, where D is the diagonal matrix with $(D)_{ij} = \deg(v_i)$. This matrix is called the Laplacian matrix of G. G is acyclic if and only if $\mathrm{trace}(L) = 2\mathrm{rank}(L)$.

Proof: G is acyclic if and only if it is a forest, i.e., G has c components and exactly $n - c = \mathrm{rank}(L)$ edges. On the other hand, by the handshaking lemma, we have $|E(G)| = 0.5\,\mathrm{trace}(L)$.

Another issue that could be observed in the solution of the linear optimization problem is the interference of vehicles when performing a similar task. It is possible to have solutions such that while the vehicle v_i moves from customer i to customer $i+1$, vehicle v_j moves from customer $i+1$ to the next one. The following constraint is added to the problem (1) to prevent vehicles from doing such service interference.

Table 3. Solution time and cost for P01.

α for depot 1	α for depot 2	α for depot 3	α for depot 4	Cost	T (sec)
1	1	1	1	634.55	17.7
1	1	1	0	623	59.57
0.66	0.5	0.5	1	632.83	35.65
0.33	0.5	0.5	1	631.03	40.87
0.33	0.5	0.5	0	620.56	66.5
0.33	0.5	0	0	603.74	146.91

Table 4. Vehicle routes in P01 for α=1.

Depot 1	Vehicle 1	51	19	40	41	13	51	
	Vehicle 2	51	18	25	51			
	Vehicle 3	51	42	44	45	15	37	51
	Vehicle 4	51	17	12	47	4	51	
Depot 2	Vehicle 1	52	6	24	14	52		
	Vehicle 2	52	8	26	7	43	23	52
	Vehicle 3	52	27	48	1	32	46	52
Depot 3	Vehicle 1	53	9	50	16	21	34	53
	Vehicle 2	53	10	33	39	30	53	
	Vehicle 3	53	49	5	11	38	53	
Depot 4	Vehicle 1	54	29	20	36	35	54	
	Vehicle 2		54 3 28	31	22	2	54	

$$x_{ij} + \sum_{h=1}^{v \neq k} x_{ij}^{k} \leq 1, \quad \forall k \in V, \quad \forall i, j \in \{2, 3, \cdots, N\} \qquad (12)$$

The total cost is the sum of all costs in each subdivided problem. One can use the tuning parameter α to adjust the complexity of the related linear integer programming. The choice of lower values for α implies that customers of a depot are divided into fewer groups. This results in lower total cost compared to the case where the customers are divided into more groups. Table 3 describes the results of tuning parameter α in problem P01 from [11]. As can be seen, by enlarging the value of α, the solution time decreases while the total cost increases. Table 4 indicates the vehicle routes in problem P01. This problem includes 54 nodes where nodes 1-50 are customers and 51-54 are depots.

5- Simulation results

Simulation results of the proposed algorithm implementation are presented and discussed in this section. Results were obtained by performing the algorithm in the MATLAB environment in a computer system with Core i7-7700HQ, 2.80 GHz, and 4GB RAM characteristics. To compare the results, 11 different and standard scenarios were picked up from Courdeau's instances [18]. In these examples, we have $n = 50$ to 360 customers, $d = 2$ to 9 depots, and $m = 2$ to 6 vehicles. We have also implemented GVNS [20] and Sadati's [21] methods for comparison.

Table 5 presents the results found by GVNS and Sadati's methods and our proposed approach with $\alpha = 1$ in each depot. The averaged values and the execution time in each instance, are presented for 3 approaches. For the GVNS method, each instance was solved 30 times with the following parameter values: $iterMax = 100$ and $maxTime = 30$ minutes. The GVNS algorithm was coded in C++ and tested on a computer

Table 5. Compare the results with GVNS and Sadati's methods.

Inst	n	m	d	BKS	GVNS ave.	T (sec) ave.	Sadati ave.	T (sec) ave.	New approach	T (sec)
P01	50	4	4	576.84	591.09	36	591.09	9.35	634.55	17.7
P02	50	2	4	473.53	476.66	33.6	476.66	20.87	515.47	17.37
P03	75	3	5	641.19	641.49	134.6	641.49	141.86	697.36	21.75
P04	100	8	2	1001.4	1025.44	702.6	1008.47	258.75	1072.2	23.4
P05	100	5	2	750.03	757.46	564.6	758.87	159.10	866	35.4
P06	100	6	3	876.50	889.79	502.8	881.76	194.81	981.76	23.5
P07	100	4	4	881.97	898.31	405	896.96	94.75	977.5	28
P12	80	5	2	1318.95	1326.85	166.2	1318.95	13.75	1429.2	17
P15	160	5	4	2505.42	2567.62	1421.4	2522.79	255.44	2841	30.06
P18	240	5	6	3702.85	3866.25	1800.6	3802.9	302.91	4288	56.55
P23	360	4	9	6095	6130	----------	6145.58	609.22	6544	79

with Intel Core i3-2370M, 2.40 GHz, and 4GB RAM [20]. Sadati's algorithm was coded in C# and tested on a Dell Precision T7810 with Intel Xeon E5-2690, 2.40 GHz, and 32GB RAM [21]. For our proposed approach, the algorithm is implemented just once.

BKS is the best-known solution cost found so far [21]. A comparison between the computer system used to solve the problem in this article and the computer systems used in the other methods has been made on the site www.cpubenchmark.net. A comparison between the program execution time in [22] shows that the speed of C++ and C# is about ten and six times faster than MATLAB, respectively. Table 5 shows that the new approach solution time is far less than the GVNS and Sadati's algorithms. Although the GVNS and Sadati's algorithm costs are less than our solution, they are unable to solve the problems in real-time due to the long computation time.

6- Conclusion

The main findings can be summarized as follows. The results discussed in Section 5 show that the new algorithm's computation time could be far less than those of the existing algorithms. In this regard, we introduced tuning parameter α that could further help one to compromise between solution time and total shipping cost. Our direction in future research is to solve a DVRP where the number of customers, traffic congestion, and so forth could be changed unknowingly. Thus, we need a fast algorithm to make the decision and solve the problem in real-time.

References

[1] D. Reyes, M. Savelsbergh, A. Toriello, Vehicle routing with roaming delivery locations, Transportation Research Part C: Emerging Technologies, 80 (2017) 71-91.

[2] R. Moghdani, K. Salimifard, E. Demir, A. Benyettou, The green vehicle routing problem: A systematic literature review, Journal of Cleaner Production, 279 (2021) 123691.

[3] M. Asghari, S.M.J. Mirzapour Al-e-Hashem, Green vehicle routing problem: A state-of-the-art review, International Journal of Production Economics, 231 (2021) 107899.

[4] B. Eksioglu, A.V. Vural, A. Reisman, The vehicle routing problem: A taxonomic review, Computers & Industrial Engineering, 57(4) (2009) 1472-1483.

[5] H. Zhang, H. Ge, J. Yang, Y. Tong, Review of vehicle routing problems: Models, classification and solving algorithms, Archives of Computational Methods in Engineering, 29(1) (2022) 195-221.

[6] J.-S. Pan, X. Wang, S.-C. Chu, T. Nguyen, A multi-group grasshopper optimisation algorithm for application in capacitated vehicle routing problem, Data Science and Pattern Recognition, 4(1) (2020) 41-56.

[7] J. Ochelska-Mierzejewska, A. Poniszewska-Marańda, W. Marańda, Selected genetic algorithms for vehicle routing problem solving, Electronics, 10(24) (2021) 3147.

[8] M.A. Islam, Y. Gajpal, T.Y. ElMekkawy, Hybrid particle swarm optimization algorithm for solving the clustered

vehicle routing problem, Applied Soft Computing, 110 (2021) 107655.

[9] S. Delalić, E. Žunić, A. Alihodžić, E. Selmanović, A discrete bat algorithm for the rich vehicle routing problem, in: 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO), IEEE, 2021, pp. 1058-1063.

[10] D. Diastivena, S. Wahyuningsih, D. Satyananda, Grey wolf optimizer algorithm for solving the multi-depot vehicle routing problem and its implementation, Journal of Physics: Conference Series, IOP Publishing, 2021, pp. 012001.

[11] Y.-H. Jia, Y. Mei, M. Zhang, A bi-level ant colony optimization algorithm for capacitated electric vehicle routing problem, IEEE Transactions on Cybernetics, (2021).

[12] A.B. Pratiwi, I.Y. Rakhmawati, E. Winarko, Flower pollination algorithm for vehicle routing problem with time windows (VRPTW), Tensor: Pure and Applied Mathematics Journal, 2(2) (2021) 45-52.

[13] L. Abualigah, M. Alkhrabsheh, Amended hybrid multi-verse optimizer with genetic algorithm for solving task scheduling problem in cloud computing, The Journal of Supercomputing, 78(1) (2022) 740-765.

[14] Q. Zhou, U. Benlic, Q. Wu, J.-K. Hao, Heuristic search to the capacitated clustering problem, European Journal of Operational Research, 273(2) (2019) 464-487.

[15] R. Paradiso, R. Roberti, D. Laganá, W. Dullaert, An exact solution framework for multi-trip vehicle-routing problems with time windows, Operations Research, 68(1) (2020) 180-198.

[16] E. Kozłowski, D. Mazurkiewicz, B. Kowalska, D. Kowalski, Binary linear programming as a decision-making aid for water intake operators, in: International Conference on Intelligent Systems in Production Engineering and Maintenance, Springer, 2017, pp. 199-208.

[17] G.D. Konstantakopoulos, S.P. Gayialis, E.P. Kechagias, Vehicle routing problem and related algorithms for logistics distribution: A literature review and classification, Operational research, (2020) 1-30.

[18] J.F. Cordeau, M. Gendreau, G. Laporte, A tabu search heuristic for periodic and multi-depot vehicle routing problems, Networks: An International Journal, 30(2) (1997) 105-119.

[19] M.J. Negreiros, N. Maculan, P.L. Batista, J.A. Rodrigues, A.W. Palhano, Capacitated clustering problems applied to the layout of IT-teams in software factories, Annals of Operations Research, (2020) 1-29.

[20] S. Nunes Bezerra, M.J.F. Souza, S.R.d. Souza, V. Nazário Coelho, A VNS-based algorithm with adaptive local search for solving the multi-depot vehicle routing problem, in: International Conference on Variable Neighborhood Search, Springer, 2018, pp. 167-181.

[21] M.E.H. Sadati, D. Aksen, N. Aras, The r-interdiction selective multi-depot vehicle routing problem, International Transactions in Operational Research, 27(2) (2020) 835-866.

[22] S.B. Aruoba, J. Fernandez-Villaverde, A comparison of programming languages in economics (No. w20263), in, National Bureau of Economic Research Cambridge, 2014.

HOW TO CITE THIS ARTICLE

M. A. Zajkani, R. Rahimi Baghbadorani, M. Haeri, Using a Tuning Parameter to Compromise Computation Time and Shipping Cost in an MDVRP, AUT J. Elec. Eng., 55(3) (Special Issue) (2023) 323-332.

DOI: 10.22060/eej.2022.21404.5474