An Improved Genetic Algorithm Approach for Vehicle Routing Problems with Backhauls

HAJEM A. DAHAM
Department of Mathematics, College of Education for Pure Science
Al Muthanna University
Samawa City
IRAQ

Abstract: In this paper, we address the vehicle routing problems with backhauls. This type of vehicle routing problem consists of two types of delivery, linehaul and backhaul cases. For the linehaul case, vehicles deliver goods/products from a single depot to receiver customers. In contrast, for the backhaul case, the delivery process starts from the sender customers and ends at the depot. This problem is classified as NP-hard, indeed, a high cost and a high number of vehicles will be caused in case of poor planning and management. Thus, it is important to develop an efficient strategy to manage this process in a way to reduces the overall cost. In this paper, we first formulated an assignment integer linear programming model, which is proven to solve instances of limited size. For large-scale instances, we developed an evolutionary genetic algorithm approach, which has been tested to be efficient for solving large-scale problems. The result shows that combining the linehaul and backhaul cases saves on delivery cost and vehicles.

Key-Words: - Linehauls; Backhauls; Jointly; Genetic Algorithm

Received: May 11, 2024. Revised: March 9, 2025. Accepted: April 12, 2025. Published: May 19, 2025.

1 Introduction

A type of vehicle routing problem (VRP) is called vehicle routing problem with backhauls (VRPB), which exists in many practical and industrial applications. In the VRPB, the delivery of all customers' requests starting from a depot to customers' locations is called the linehaul case [8]. In contrast, the pickup of customers' requests from customers' locations to the origin depot is called the backhaul case. However, unlike the common vehicle routing problem with the mixed pickup and delivery, the combined case in the VRPB, in which all linehauls are first visited by the vehicle, followed by all backhauls, is the combined linehauls and backhauls case [9]. This case arises to avoid the mixed reloading of vehicles during the service. The paper [10] reported that the VRPB in general is a saving cost for companies, since the use of backhauling for vehicles will significantly contribute to reducing the overall delivery cost. However, the restriction of visiting all linehauls followed by all backhauls on the same route increased the complexity of the VRPB. An efficient delivery plan should be designed to confront the difficulty and complexity of the VRPB. The standard vehicle routing problem with backhauls (SVRPB) is the general case of the VRPB, where each vehicle has a specific capacity that needs to be satisfied in the delivery plan. Satisfying the customer demand with the existence of the vehicle capacity restrictions in the SVRPB has also raised the difficulty of the problem and made it NP-hard to solve. In this paper, we studied the SVRPB consisting of the delivery of linehauls and backhauls cases by a fleet of vehicles. An assignment integer linear programming model is formulated to solve this type of problem. In order to solve large-sized instances, an evolutionary genetic algorithm approach is created. The computational results show that the genetic algorithm is efficient in solving the SVRPB, and the combination of linehauls and backhauls cases is considered to save cost.

The paper is structured as follows: The literature is reviewed in Section 2. The problem description is illustrated in Section 3, while the optimization model is illustrated in Section 4. The computational results are presented in Section 5, and finally, Section 6 is for the conclusion.

2 Literature Review

To the best of our knowledge, only two exact methods were developed to solve the SVRPB. The first exact method was introduced by [16], where the integer mathematical model of the SVRPB was solved using a Lagrangian method. Later, [11] created the second exact approach, which was based on solving the set partitioning model of the problem using the dual linear programming relaxation. On the other hand, a number of articles have been published to solve the problem considering heuristic methods. [3] classified these heuristics into different types, comprising: classical, local search, population search, and neural networks heuristics.

Table 1: Classical Heuristics' Literature

Literature	Type of Heuristic
[6]	Extension of the Clarke and Wright method
[9]	Space-filling curves
[10]	The LHBH based on GAP
[16]	Route construction heuristic for the VRPBTW
[18]	Improved cluster-first, route-second method (directed VRPB)

Classical heuristic where introduced before the year 2000, as in Table 1. In contrast to the classical heuristics, the local search heuristics were those methods that were introduced after the year 2000 and are based on the neighborhood of the search space, as illustrated in Table 2.

Table 2: Local Search Heuristics' Literature

Literature	Type of Local Search Heuristic
[13]	Reactive tabu search based on two route construction methods
[1]	Reactive tabu search combined with two constructive methods
[14]	Adaptive neighborhood search method
[20]	Reactive tabu search combined with an adaptive memory scheme
[7]	An ant colony system
[23]	Variable neighborhood local search
[4]	An iterated local search algorithm
[21]	Clustering, routing and local search
[2]	A deterministic iterated local search algorithm
[12]	Proxy Home Agent (PHA)

In this paper, we investigate the delivery cost of the SVRPB from the perspective of the separated delivery of linehauls and backhauls cases compared to the combined case (combined linehauls and backhauls case). This idea is inspired by [5, 22], where the pick up and delivery of containers individually and in combined cases is investigated. An assignment integer linear programming model, followed by an improved genetic algorithm (GA) approach, is applied to solve these different cases.

3 Problem description

The vehicle routing problem with backhauls (VRPB) can be illustrated as a directed graph $G = (V, A)$, where V is the set of nodes consisting of $D, \mathcal{L}$ and $\mathcal{B}$ nodes. Node D is the central depot, where vehicles start/end their trips (routes). $\mathcal{L}$ is the set of nodes of linehauls' customers, where $\mathcal{L} = 1, \ldots, l$, while $\mathcal{B}$ is the set of nodes of backhauls' customers, where $\mathcal{B} = b + 1, \ldots, l + b$. On the other hand, A is the set of arcs between nodes, where $A = (i,j) : i,j \in V, i \neq j$. A demand of d_j is associated with each customer (l *and* b) and a number of $\mathcal{H}$ vehicles of Q capacity, which are available for the service. In the VRPB, there are two cases of delivery/pickup, in the first case, loaded vehicles start trips from the depot to linehauls' customers (see Figure 1), where the load needs to be delivered.

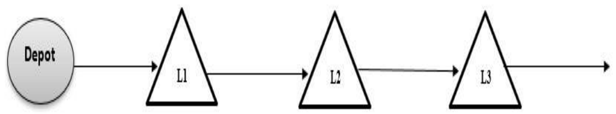

Figure 1: Linehauls case

However, in the second case, empty vehicles start to collect loads from backhaul customers and return to the depot as in Figure 2.

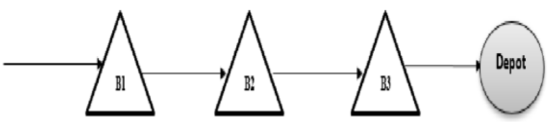

Figure 2: Backhauls case

If the two cases are combined in the same route, this is called the combined linehaul and backhaul case, as in Figure 3.

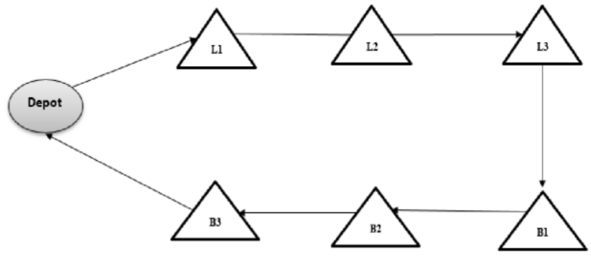

Figure 3: Combined linehauls and backhauls case

Our objective in this paper is to investigate the two separated cases and compare them with the combined case to minimize the total delivery/pickup cost.

4 Optimization Model

4.1 Decision Variables

Binary variables x_{ijh} and x_{ih} are created to denote the decision on how the orders should be delivered, as combined or separately for delivery. For example, $x_{ijh} = 1$ means linehauls i and backhauls j need to be delivered on the same route by vehicle h. However, $xi = 1$ means that linehsuls/backhauls i need to be delivered separately.

4.2 Mathematical Model

An Assignment Integer Linear Programming (AILP) model is formulated to find the best combination of decisions for delivery route planning. The model consists of the most important practical constraints that are normally used in industry.

$$\min \sum_{i \in \mathcal{L}} \sum_{j \in \mathcal{B}} \sum_{h \in \mathcal{H}} c_{ij} x_{ijh} + \sum_{i \in \mathcal{L} \cup \in \mathcal{B}} \sum_{h \in \mathcal{H}} c_{ij} x_{ih}; \quad (1)$$

$s.t.$

$$\sum_{j\in\mathcal{B}}\sum_{h\in\mathcal{H}} x_{ijh}; = 1,\ \forall_i \in \mathcal{L}; \quad (2)$$

$$\sum_{i\in\mathcal{L}}\sum_{h\in\mathcal{H}} x_{ijh}; = 1,\ \forall_j \in \mathcal{B}; \quad (3)$$

$$\sum_{h\in\mathcal{H}} x_{ih} = 1,\ \forall_i \in \mathcal{L} \cup \mathcal{B}; \quad (4)$$

$$\sum_{h\in\mathcal{H}} x_{ijh} d_{ij} \leq Q,\ \forall_i \in \mathcal{L}, \forall_i \in\ \mathcal{B}; \quad (5)$$

$$\sum_{h\in\mathcal{H}} x_{ih} d_i \leq Q,\ \forall_i \in \mathcal{L} \cup \mathcal{B}; \quad (6)$$

$$x_{ijh}, x_{ih}\ \in \{0, 1\}\ \forall_i, j\ \in \mathcal{N}, \forall_j \in \mathcal{H}; \quad (7)$$

The objective function (1) is to minimize the total travelling cost between the depot and customers for the separated and joint cases. Constraints (2)-(4) force all orders to be delivered separately or jointly. Constraints (5)-(6) are to ensure that the maximum capacity of orders (separately and jointly) for all routes is less than or equal to the total vehicle capacity. Finally, constraint (7) is to represent the domain of variables.

4.3 Genetic algorithm approach

The genetic algorithm (GA) has been applied as a metaheuristic approach for solving different types of optimization problems. The GA is usually applied to solve complicated and large-scale optimization problems, which cannot be solved by exact methods. The GA starts by creating initial solutions for the investigated problems, and the solutions are evaluated based on their fitness values to choose a group of these solutions for the combination processes. The crossover and mutation are then applied to these selected solutions to create new solutions (offspring). The GA process continues repeatedly until a best (good) solution is obtained. Usually, the exact optimization approaches cannot be applied for solving the VRPB as the latter is classified as NP-hard [18]. Thus, heuristic/metaheuristic approaches are required to solve these types of problems. The GA is believed and tested as an efficient method for solving complicated and large-scale problems [19]. For this reason, we applied the GA to solve the VRPB in order to find near optimal solution. In the next sections, we will describe the GA in more detail.

4.4 Chromosomes (Solutions) demonstration

In the VRPB, the chromosome refers to the solution of this problem, which is a permutation of the linehauls and backhauls customers as shown in Figure 4(A). The capacity constraints cause infeasible solutions of the chromosomes. To tackle the infeasibility, we sample the chromosomes from the delivery category (linehauls) until the demand of linehauls reaches the maximum capacity of the vehicle. Then the sample process starts from the pickup (backhaul) category and continues until the vehicle capacity is full. When the capacity is full for the delivery and pickup then the depot is inserted at the end to terminate the route, which represents a vehicle, as shown in Figure 4(B).

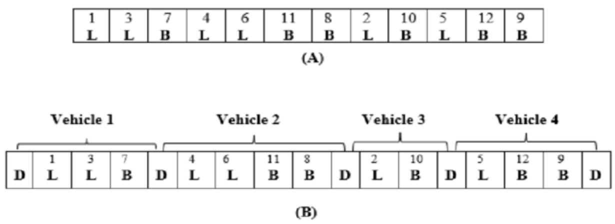

Figure 4: Chromosomes (Solutions) demonstration of the VRPB

However, some routes in the chromosome consist of only linehauls or only backhauls customers when we deal with the two cases separately. For this purpose, we used an insertion method as demonstrated in Algorithm 1, which helps in creating feasible routes in terms of vehicle capacity. However, we still face the feasibility of using a larger number of fleets rather than the available fleet, which we tackle within the fitness function by penalizing the large number of fleets, which helps to exclude them in the evaluation process.

In more detail, we first assume that an empty route starts from the depot, then we first insert linehauls until the vehicle capacity is reached. Similarly, for the backhauls case, we insert them in the created route until the vehicle capacity is reached; in this case, the current route (vehicle) is terminated, and then we start a new route until all routes are completed. To illustrate the insertion method, an example of 6 linehauls and 6 backhauls is considered, which also includes the depot (D) and 4 vehicles. As shown in Figure 4(A), a chromosome is a random permutation of all linehauls and backhauls, and then we insert depots (D) into the chromosome to divide it into routes (vehicles) (see Figure 4(B)). In more details, the first vehicle (Vehicle 1) starts from the depot, and then visit the linehauls (1, 3), and since that the vehicle capacity is reached, then the same vehicle travels to backhaul (7), and similarly since the capacity of the vehicle is full, thus we terminate this route (vehicle) by inserting the depot at the end. The same procedure is applied for all other vehicles until this chromosome is completed.

Algorithm 1 - Insertion method

Step 1: Let N refer to the total number of linehauls (L) and backauls (B). Let $I = 1$.

Step 2: Initialize an empty route starting from the depot (D). Let $R = 1$.

Step 3: ***while*** $I \leq N$

Step 2.1: Insert the linehauls (L) from N.

If the total demand of the linehauls (L) does not violate the capacity of the vehicle

then set $I = I + 1$ and go to Step 3;

elseif the demand of the linehauls (L) violates the capacity of the vehicle, then go to Step 4:

Step 4: Insert the backhauls (B), I, from N.

If the total demand of the backhauls does not violate the capacity of the vehicle

then set $I = I + 1$ and go to Step 3;

elseif the demand of the backhauls violates the capacity of the vehicle, then go to **Step 5:**

Step 5: Insert the depot (D) - terminate route (R=1) for the current vehicle and go to Step 1.

4.5 The initial population, evaluation, and selection

As we mentioned, the initial population for the VRPB is built based on the permutation of the delivery and pickup process. Considering the feasibility of the chromosomes regarding the capacity of vehicles, we insert the depot at the end of the routes. In this case, all the chromosomes (solutions) have a similar length, but the number of vehicles may be different for each chromosome based on the feasibility of the chromosomes. When the initial population is constructed, then each chromosome is evaluated based on the fitness values of chromosomes, which in this problem is represented by the objective function (9). As we know that the objective of the VRPB is to minimize the transportation cost to satisfy all customers.

$$Fit(i) = OF(i) \quad (8)$$

$$= TC(i) + P(i) \quad (9)$$

Where $Fit(i)$ is the fitness value for chromosome i, while $OF(i)$ is the original objective function of the problem, which comprises the total transportation cost $TC(i)$ and the penalty of a larger number of vehicles $P(i)$, adding to solution (i). Based on the fitness values, the selection process, which is the Roulette Wheel Selection (RWS) approach, is applied to select a group of solutions for the next recombination processes of the GA. Solutions are chosen based on the proportion of the fitness values of each chromosome to the total fitness of all chromosomes. Solutions with the smaller fitness values have a higher chance of being selected as a parent for the next recombination process.

4.6 Crossover and Mutation Operators

To apply the recombination process, two operators of crossover and mutation are used. In the crossover process, a number of elements of the first parent are chosen to start the first offspring, then to complete the first offspring, non-repeated elements are chosen from the second parent. Conversely, the second offspring is constructed by selecting the first part of this offspring from the second parent, and the second part of the second offspring is completed from the first parent. To illustrate the crossover process, two parents are presented in Figure 5.

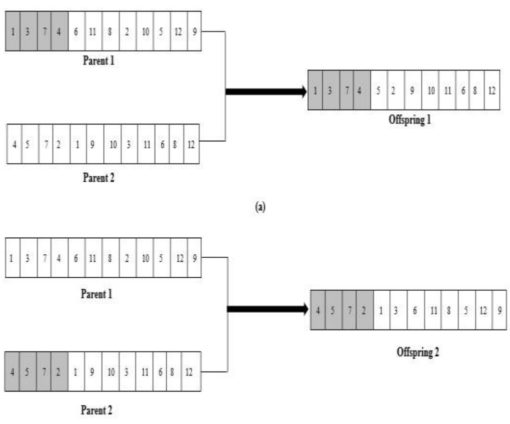

Figure 5: Crossover Process

Firstly, elements (1, 3, 7, 4) are chosen from Parent 1 to start Offspring 1, then the rest of the elements are completed from Parent 2 as shown in Figure 5(a). Similarly to Figure 5(b), Offspring 2 is constructed, but this time the first part of Offspring 2 is picked from Parent 2 (2, 5, 7, 4), while the rest of the elements of Offspring 2 are completed from non-repeated elements of Parent 1. The mutation operator is the second process which usually applied to explore the search space of the GA. In the mutation process, two elements are selected from a solution that is chosen randomly from the population. These two elements are swapped with each other to create the new solution. Figure 6 demonstrates the mutation operator where the two elements 4 and 5 are chosen from the solution as in Figure 6(a) and then swapped

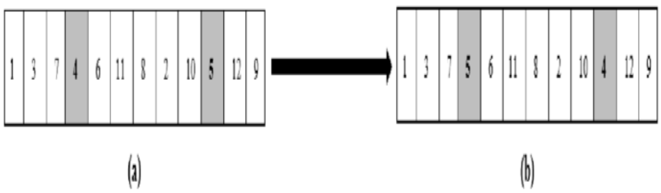

Figure 6: Mutation Process

All the above operations are executed repeatedly, and the GA terminates when the best solution among generations is achieved. The termination of the GA processes is based on a specific stopping criterion, which for the VRPB we use a large number of generations.

5 Computational experiments

A computational experiments are illustrated in this section for randomly generated instances.

In these instances, an approximated random geographical information is simulated for a single depot and a group of linehaul and backhaul customers. The distances between the central depot and customers, and also between customers, are rescaled approximately. A variant number of linehauls and bachauls orders and fleets are considered for the pickup and delivery. Both solution methods are coded in MATLAB R2017b and executed by CPLEX 12.6.1. To tackle randomness, the instances of the GA are solved for an average of 25 times.

Table 3: Results for solving examples by AILP and GA.

#	Number& Type of orders		# Fleet		Tota l cost	CPU	(%) Gap of GA
Or.	Li.	Ba.	ava.	us.		time	from MILP
MILP solution							
5	3	2	2	2	347	0:11	-
	2	3		2	329	0:11	-
	3	2		1	364	0:12	-
	2	3		2	372	0:12	-
	3	2		1	389	0:11	-
10	5	5	4	3	585	0:43	-
	6	4		3	577	0:42	-
	5	5		2	564	0:43	-
	4	6		3	553	0:43	-
	5	5		2	521	0:44	-
15	8	7	6	4	864	0.06 389	-
	7	8		4	821	0.06 528	-
	9	6		3	833	0.05 764	-
	6	9		3	822	0.06 389	-
	8	7		3	810	0.05 833	-
20	10	10	8	5	1124	2:23	-
	11	9		6	1215	2:25	-
	12	8		5	1182	2:26	-
	9	11		6	1195	2:24	-
	8	12		5	1204	2:23	-
30	15	15	10	8	1524	3:41	-
	16	14		7	1489	3:55	-
	17	13		8	1567	0.17 014	-
	14	16		7	1534	3:44	-
	13	17		8	1566	0.18 403	-
GA solution							
5	3	2	2	2	387	0:18	11%
	2	3		2	372	0:19	13.1 0%
	3	2		1	392	0:18	7.70 %
	2	3		2	402	0:19	8.10 %
	3	2		1	413	0:18	6.20 %

10	5	5	4	3	643	0:24	9%
	6	4		3	639	0:21	10.70%
	5	5		2	625	0:20	10.80%
	4	6		3	618	0:21	11.70%
	5	5		2	587	0:24	12.60%
15	8	7	6	4	924	0:33	7%
	7	8		4	911	0:34	11%
	9	6		3	917	0:37	10.10%
	6	9		3	915	0:34	11.30%
	8	7		3	904	0:38	11.60%
20	10	10	8	5	1216	0:44	8.20%
	11	9		6	1296	0:47	6.70%
	12	8		5	1287	0:42	8.80%
	9	11		6	1283	0:45	7.40%
	8	12		5	1305	0:43	8.40%
30	15	15	10	8	1684	0.04375	10.50%
	16	14		7	1633	0.04722	9.70%
	17	13		8	1698	0.04861	8.30%
	14	16		7	1689	0.04375	10.10%
	13	17		8	1695	0.04306	8.20%

5.1 Results for the AILP and GA

We tested the cost gap of the solution of the GA approach from the solution obtained from solving the AILP model for small-sized instances. Different sizes of instances (5, 10, 15, 20, 30) are tested using the AILP and GA as shown in Table 3. The number and type of orders are given in the first, second, and third columns of Table 3, in addition to the number of the fleet, which is illustrated in the fourth column.

We can see from the result that the gap of the total cost obtained from the GA is higher than the solution obtained from the AILP by only 6-13% of the real total cost of the delivery plan. The number of the used fleet for the AILP and GA is the same, which is approximated to the nearest integer number since the number of the used vehicles is the average of the fractional solution of the GA as we run each instance 25 times. The computational times for the AILP are larger than the solution time of the GA for each instance. It is clear from the gap between the total cost of the GA solution from the AILP solution that the GA approach performs acceptably. This means that the developed GA approach is efficient in solving this type of problem. As we can see from the result in Table 3, the AILP model in this case can only solve up to 30 orders because of the memory capacity. Thus, we develop the GA to solve larger instances.

5.2 Results for linehauls and bachauls cases

As we found in the previous section that the AILP solved only 30 orders. Thus, we will only use the GA in this section to investigate the difference of using the two cases (linehauls and bachauls) separately and jointly. As shown in Table 4, a number of 20-50 linehaul and backhaul orders have been tested separately. Next, the two cases are combined to solve them jointly for up to 100 orders. The available fleet sizes for all these separate cases are between 8-22 vehicles, while the available fleet for the joint cases is between 16-44 vehicles. As we mentioned before that we ran the GA solution 25 times, thus, the values of the result in this case refer to the average of 25 values. As demonstrated in Table 4, the average total cost for the combined case is reduced by 9.5-12% than the summation of the two separated cases. At the same time, the used fleet is also reduced for the joint case compared to the summation of the two separated cases. For instance, in the example of 90 orders, servicing the two cases separately needs 13 vehicles for the linehaul orders and 12 vehicles for the backhaul orders. While for the joint case, just 21 vehicles are needed to execute the pickup and delivery for orders, which means that 4 vehicles will be saved in this case. It is noticed that the solution time to perform the joint cases is larger compared to the two separated and the reason is the large number of orders in the joint case.

Table 4: Results for Separately and Jointly *Linehaul* and *Backhaul* Cases

Or.	**# Fleet**		**Mean of**	**CPU**	**Saving cost(%)**
	av.	**us.**	**total cost**	**time**	
20-Li.	8	6	1117	3:45	
20-Ba.	8	5	1084	3:47	
40-Sep.			2201	0.31389	
40-Joi.	16	9	1987	0.35208	10.80%
25-Lin.	12	6	1215	0.18958	
25-Ba.	12	7	1185	0.19097	
50-Sep.			2400	0.35278	
50-Joi.	24	10	2191	8:45	9.50%
30-Lin.	14	7	1321	4:27	
30-Ba.	14	7	1294	4:32	
60-Sep.			2615	8:59	
60-Joi.	28	12	2344	9:37	11.60%
35-Lin.	16	8	1514	5:57	
35-Ba.	16	9	1487	5:46	
70-Sep.			3001	11:03	
70-Joi.	32	15	2723	12:07	10.20%

40-Lin.	18	11	1690	0.30347	
40-Ba.	18	10	1710	0.29583	
80-Sep.			3400	13.43	
80-Joi.	36	17	3052	14:54	11.40%
45-Lin.	20	13	1823	7:33	
45-Ba.	20	12	1793	7:45	
90-Sep.			3616	0.6375	
90-Joi.	40	21	3228	0.66736	12%
50-Lin.	22	14	2023	0.37639	
50-Ba.	22	15	1989	0.38264	
100-Sep.			4012	17:33	
100-Joi.	44	24	3605	0.80625	11.20%

6 Conclusions

In this paper, we investigated the delivery of linehauls and backhauls cases by a fleet of vehicles. An assignment integer linear programming model is formulated to solve this type of problem. This mathematical model solves only limited-sized instances. Thus, an evolutionary genetic algorithm approach is also developed to solve large-sized instances. The procedures of the genetic algorithm have been designed to fit this type of problem. The result shows that the gap of the total cost between the solution obtained from the assignment integer linear programming model and the genetic algorithm solution is small, which suggests that this type of evolutionary algorithm is efficient to solve the problem. Both cases of the linehauls and backhauls are investigated separately and jointly, and the result shows that it is worthwhile to combine the linehauls and backhauls cases in this type of delivery a cost savings. As a future work, other heuristic or even meta-heuristic approaches can be applied to this type of vehicle routing problem, and the result may be improved from this work or even past work.

References:

[1] Brandão, J. (2006). A new tabu search algorithm for the vehicle routing problem with backhauls. European Journal of Operational Research, 173(2), 540-555. https://doi.org/10.1016/j.ejor.2005.01.042
[2] Brandão, J. (2016). A deterministic iterated local search algorithm for the vehicle routing problem with backhauls. Top, 24(2), 445-465. https://doi.org/10.1007/s11750-015-0404x
[3] Koç, Ç., & Laporte, G. (2018). Vehicle routing with backhauls: Review and research perspectives. Computers & Operations Research, 91, 79-91. https://doi.org/10.1016/j.cor.2017.11.003
[4] Cuervo, D. P., Goos, P., Sörensen, K., & Arráiz, E. (2014). An iterated local search algorithm for the vehicle routing problem with backhauls. European Journal of Operational Research,237(2),454-464. https://doi.org/10.1016/j.ejor.2014.02.011
[5] Daham HA, Yang X, Warnes MK (2017). An efficient mixed integer programming model for pairing containers in inland transportation based on the assignment of orders. Journal of the Operational Research Society 68(6):678-694. https://doi.org/10.1057/s41274-0160132-0
[6] Deif, I., & Bodin, L. (1984). Extension of the Clarke and Wright algorithm for solving the vehicle routing problem with backhauling. In Proceedings of the Babson conference on software uses in transportation and logistics management (pp. 75-96). Babson Park, MA.

[7] Gajpal, Y., & Abad, P. L. (2009). Multi-ant colony system (MACS) for a vehicle routing problem with backhauls. European Journal of Operational Research,196(1),102-117. https://doi.org/10.1016/j.ejor.2008.02.025

[8] Ghaziri, H., & Osman, I. H. (2006). Self-organizing feature maps for the vehicle routing problem with backhauls. Journal of Scheduling, 9(2), 97-114. https://doi.org/10.1007/s10951-006-6774-z

[9] Goetschalckx, M., & Jacobs-Blecha, C. (1989). The vehicle routing problem with backhauls.European Journal of Operational Research, 42(1), 39-51. https://doi.org/10.1016/03772217(89)90057-X

[10] Jacobs-Blecha, C. and Goetschalckx, M., (1992). The vehicle routing problem with backhauls: properties and solution algorithms. National Transportation Research Board, 13.

[11] Mingozzi, A., Giorgi, S., & Baldacci, R. (1999). An exact method for the vehicle routing problem with backhauls. Transportation Science, 33(3),315-329. https://doi.org/10.1287/trsc.33.3.315

[12] Novak, R. (2004). Proxy home agents for route optimization in HAP networks. WSEAS Transactions on Computers, 3(6), 1819-1824.

[13] Osman, I. H., & Wassan, N. A. (2002). A reactive tabu search meta-heuristic for the vehicle routing problem with back-hauls. Journal of Scheduling, 5(4),263-285. https://doi.org/10.1002/jos.122

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

[15] Sahar, G. S., Okba, K., Abdelkader, L., Amine, Y. M., Euler, R., Bounceur, A., & Hammoudeh, M. (2021). An optimized scalable multi-ant colony system for multi-depot vehicle routing problems using a reactive multi-agent system. WSEAS Transactions on Systems, 20, 249-259.

[16] Thangiah, S. R., Potvin, J. Y., & Sun, T. (1996). Heuristic approaches to vehicle routing with backhauls and time windows. Computers & operations research, 23(11), 1043-1057.

[17] Toth, P., & Vigo, D. (1997). An exact algorithm for the vehicle routing problem with backhauls. Transportation science, 31(4), 372-385.

[18] Toth, P., & Vigo, D. (1999). A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls. European Journal of Operational Research, 113(3), 528-543. https://doi.org/10.1287/trsc.31.4.372

[19] Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2014). A unified solution framework for multi-attribute vehicle routing problems. European Journal of Operational Research,234(3), 658-673. https://doi.org/10.1016/j.ejor.2013.09.045

[20] Wassan, N. (2007). Reactive tabu adaptive memory programming search for the vehicle routing problem with backhauls. Journal of the Operational Research Society, 58(12), 1630-1641. https://doi.org/10.1016/j.ejor.2013.09.045

[21] Yalcın, G. D., & Erginel, N. (2015). Fuzzy multi-objective programming algorithm for vehicle routing problems with backhauls. Expert Systems with Applications, 42(13), 5632-5644. https://doi.org/10.1016/j.eswa.2015.02.060

[22] Yang, X., & Daham, H. A. (2020). A column generation-based decomposition and aggregation approach for combining orders in inland transportation of containers. OR Spectrum, 42(1), 261-296.

[23] Zachariadis, E. E., & Kiranoudis, C. T. (2012). An effective local search approach for the vehicle routing problem with backhauls. Expert Systems with Applications, 39(3), 3174-3184. https://doi.org/10.1016/j.eswa.2011.09.004

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The author contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The author has no conflict of interest to declare that is relevant to the content of this article.