



Hybrid ILS-VND Algorithm for the Vehicle Routing Problem with Release Times

William Torres-Tapia^(✉), Jairo Montoya-Torres^(ID), and José Ruiz-Meza^(ID)

Research Group on Logistics Systems, Universidad de La Sabana, km 7 autopista norte de Bogotá, D.C., Chia, Colombia
{williamtorta, jairo.montoya, joseume}@unisabana.edu.co

Abstract. Modern market behavior and globalization have increased the complexity of supply chains, dealing logistics managers to implement strategies to fulfill customers' needs. Hence, product transportation activities are highly dependent on production outputs. In the literature, the product distribution in supply chains has traditionally been approached as a vehicle routing problem, for which several variants have been studied. However, traditional approaches consider that the set of delivery orders are available at the initial time of route design. So, inspired by the complexity of production-distribution problems, this paper addresses the case in which delivery orders have different availability at the time of designing the distribution routes (i.e., each order has its own release time). The problem is modeled as a mathematical program and solved using an Iterated Local Search algorithm hybridized with Variable Neighborhood Descent (ILS-VND). Experiments are run using adapted benchmark instances and results show the efficiency and effectiveness of the proposed algorithm.

Keywords: Metaheuristic · Vehicle routing problem · Variable neighborhood descent · Iterated local search

1 Introduction and Motivation

A supply chain is a network of actors (companies and organizations) directly or indirectly involved in fulfilling a customer request. When dealing with goods manufacturing, the supply chain not only includes manufacturers and suppliers, but also transporters, warehouses, retailers, and customers themselves [16]. From the view of the focal enterprise, the operational functions in the supply chain can be defined as procurement, production, and distribution. These three main activities are highly interconnected, and decisions at each stage do impact the other stages. Indeed, at the operational level, for example, decisions regarding the frequency of raw material arrival from the supplier determines the earliest time a production order can start at the manufacturing floor, while the completion times of production orders at the manufacturer defines the order availability for delivery by the transporter. The opposite is also valid: production planning is influenced by the planning of transportation of finished products.

Supported by School of Engineering Universidad de La Sabana and research grant INGPLD-10-2019 from Universidad de La Sabana.

However, because of the complexity of such integrated problem, decisions regarding production and transportation are most frequently made separately so that efficiencies at each stage can be reached [4]. On another hand, in order to supply these requests, companies are forced to increase inventory levels without taking into account that they impact the logistics costs of the product [11]. According to the literature (e.g., [12, 19]), the logistics costs can constitute up to 30% of the product's final cost. At the same time, transportation and inventory holding are the logistics activities that mainly absorb costs. Empirical studies have shown that each of them represents between 50% and 66% of the total logistics costs [2].

In the search of alternatives that allow obtaining a balance between customer satisfaction and logistics costs, several approaches have been studied in the literature to optimize the design of distribution routes. By doing so, it is possible to achieve a reduction in total operating costs of between 3% and 20% [17]. The problem of designing distribution routes is generally approached in the literature as the Vehicle Routing Problem, which is known to be NP-hard, since it generalizes the Travelling Salesman Problem (TSP) and the Bin Packing Problem (BPP) which are both well-known NP-hard problems [6]. This intractability means that optimal solutions are difficult to obtain in reasonable computational time. Hence, researchers and practitioners alike prefer the use of approximate algorithms to solve these problems.

A huge variety of versions of the VRP has been studied in the literature, as witnessed by several recent literature review papers (i.e., [1, 3, 5, 7–9, 13–15]). Among the different variants, the vehicle routing problem with order ready times has been rarely studied. In such a context, the aim of this paper is to propose both a mathematical formulation and a solution procedure for the VRP with order release times (VRP-RT). This variant of the problem is inspired from industrial applications arising in the combined production scheduling and vehicle routing problem [10, 20].

This paper is organized as follows. Section 2 describes the problem under study and its mathematical formulation. The proposed solution procedure based on the hybridization of Iterated Local Search and Variable Neighborhood Descent (ILS-VND) is presented in Sect. 3. Computational experiments and the analysis of results are presented in Sect. 4. This paper ends in Sect. 5 by presenting some conclusions and drawing opportunities for future research.

2 Problem Description and Mathematical Model

Starting from a set of production orders manufactured at the production stage of a supply chain, they are then delivered to a set of geographically dispersed customers. The completion times of each production order once manufacturing has finished become a release or ready time for delivery start. This delivery of finished products is modeled as a vehicle routing problem with release times (VRP-RT), which is computationally difficult to solve since it belongs to the class of NP-hard problems.

Formally speaking, a set of n production orders are to be manufactured and then shipped to customers once completed. The delivery is carry out using an infinite fleet of vehicles (e.g., trucks) with unlimited capacity Vehicles leave the factory every h hours. Each production order B_i consists of a set j of products $B_i = \{J_{i,1} \dots J_{i,n}\}$ to be delivered

to a customer with a known location. In addition, each delivery order B_i has a release time C_i defined by the completion time of manufacturing. We consider that vehicles must return to the factory after the delivery route is completed. Therefore, the vehicle routing problem is modeled as a VRP with order release times. In addition, each route r includes a start travel time H_r , a total travel time D_r , and the delivery time of the last order CR_r . The longest delivery time CR_{Max} is defined as $\max\{CR_r\} \forall r \in R$. The routing problem is modeled in a complete undirected graph $G' = (V', A')$, where $V' = (1 \dots n)$ is the set of nodes (clients), and A' is the set of arcs connecting each pair of nodes (clients). Each arc has a non-negative integer value $T_{i,h}$ representing the time a vehicle travels from node l to node h .

Since the objective of the problem is to minimize the maximum delivery date of the orders CR_{Max} , a mathematical program can be proposed as follows:
Sets:

- I : Order sets ($i \in I = \{0, 1 \dots n\}$)
- R Set of routes ($r \in R = \{1 \dots R_n\}$)
- Alias: $j = i$

Parameters:

- C_i : Release time of the order i .
- $T_{i,j}$: Distance between points i, j .
- M : Very large number.
- N : Total number of orders.
- R : Total number of routes.
- Mk : Makespan.
- C_{min} : Completion time of first order in production.

Decision variables:

- H_r : Route start time r .
- D_r : Route total travel time r .
- $BN_{i,r} = 1$ if order i is shipped on route r , 0 otherwise
- CR_{MAX} : Delivery time of last order.
- $XR_{r,i,j} = 1$ if order j is delivered after order i on route r , 0 otherwise
- $U_{r,i}$: Number of arcs of node j on route r

Objective function:

$$Min : CR_{MAX} \tag{1}$$

S.A

$$H_r = C_{min} + h \forall r = 1 \tag{2}$$

$$H_r = H_{r-1} + h \forall r > 1 \tag{3}$$

$$H_r + M * (1 - BN_{i,r}) \geq C_i \forall i \forall r \tag{4}$$

$$\sum_{\forall r} BN_{i,r} = 1 \forall i \quad (5)$$

$$\sum_{\forall j \neq i \neq 0} XR_{r,i,j} = BN_{i,r} \forall i > 0 \forall r \quad (6)$$

$$\sum_{\forall i \neq j \neq 0} XR_{r,i,j} = BN_{j,r} \forall j > 0 \forall r \quad (7)$$

$$\sum_{\forall j} XR_{r,0,j} = 1 \forall r \quad (8)$$

$$\sum_{\forall i} XR_{r,i,0} = 1 \forall r \quad (9)$$

$$U_{r,i} + 1 \leq N * (1 - XR_{r,i,j}) + U_{r,j} \quad \forall i > 0, j > 0 \forall j \neq i \forall r \quad (10)$$

$$D_r = \sum_{\forall i} \sum_{\forall j} T_{i,j} * XR_{r,i,j} \forall r \quad (11)$$

$$CR_{MAX} \geq D_r + H_r \forall r \quad (12)$$

$$H_r, D_r \in Z \forall r \quad (13)$$

$$BN_{i,r} \in \{0, 1\} \quad \forall r \forall i \quad (14)$$

$$XR_{r,i,j} \in \{0, 1\} \quad \forall r \forall i \forall j \quad (15)$$

$$CR_{MAX} \in Z \quad (16)$$

In this mathematical formulation, constraints (2) ensure that the first vehicle undertakes the trip h time units after the production system finishes processing the first order. Constraints (3) declare that subsequent trips start at intervals of h time units. Constraints (4) guarantee that orders start after the production completion time. Constraints (5), (6) and (7) make sure that all orders are assigned to a vehicle. Constraints (8) and (9) guarantee that the vehicle on each route starts and ends the route at the factory. Equations (10) correspond to subtour elimination. The total travel time for each r is calculated using Constraints (11). Constraints (12) compute the longest delivery time in the system. Finally, Constraints (13), (14), (15) and (16) ensure values of decision variables.

3 Proposed Solution Procedure

This paper proposes the solution approach shown in Fig. 1. This solution approach is divided into five steps. In the first step, the start times of each vehicle are determined and the orders to be delivered are assigned to the vehicle. For the allocation, the vehicle loads all orders that have a release time lower than the route start time and that have not been allocated yet. Next, the Nearest Neighbor (NN) heuristic is used to obtain an initial solution for each route. To improve these solutions, the Iterative Local Search (ILS) metaheuristic hybridized with the Variable Neighborhood Descent (VND) heuristic. Finally, the maximum delivery time is calculated.

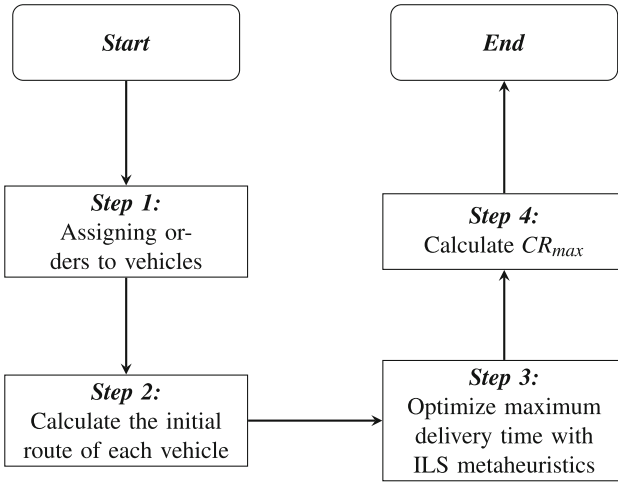


Fig. 1. Solution approach

Algorithm 1. ILS

Require: S_0
 1: $S \leftarrow VND(S_0)$
 2: $BS \leftarrow S$
 3: **repeat**
 4: $S \leftarrow Perturbation(S)$
 5: $S \leftarrow VND(S)$
 6: **if** $S < BS$ **then**
 7: $BS \leftarrow S$
 8: **end if**
 9: **until** The stop criterion was achieved

The ILS metaheuristic is described in Algorithm 1, where S_0 is the initial solution obtained with the NN heuristic. The current solution is defined as S while the best

solution is defined as BS . In line 1, the algorithm improves the initial solution using the VND heuristic. In line 2, the best solution of the problem is updated. The loop between line 4 and line 8 is executed until the running time of the algorithm has elapsed. In line 4, a perturbation is applied to the current solution in order to avoid staying in a local optimum solution. This paper uses the 2-OPT movement. Line 5 refreshes the current solution using the VND heuristic. Line 6 evaluates if the current solution is the best available solution. If so, the best solution of the algorithm is updated in line 7.

The VND algorithm is described in Algorithm 2, where S_0 is defined as the initial solution, S as the current solution and S^* as the local solution of the algorithm. N is the set of neighborhoods, while K is the index of N and T as the total number of neighborhoods. N is composed of the “Relocate” and “Swap” moves and are explained next. The algorithm starts by converting the initial solution into the current solution and into the best solution (see lines 1 and 2). In line 3, the first neighborhood of N is selected. Until the local search has been applied to the last neighborhood, the algorithm is executed cyclically from line 5 to line 11. In line 5, S is updated by applying the local search on the N_k neighborhood. In local search algorithm, best move that improves the assignment is selected. An evaluation is made to see if the current solution is better than the local solution in line 6. If so, the local solution is updated and the search is directed to neighborhood 1 (see lines 7 and 8). Otherwise, the algorithm moves to the next neighborhood (see line 10). Finally in line 12, the best local solution is returned.

Algorithm 2. VND

Require: S_0
Require: $N = \{N_1, \dots, N_T\}$
1: $S \leftarrow S_0$
2: $S^* \leftarrow S_0$
3: $K \leftarrow 1$
4: **while** $K \leq T$ **do**
5: $S \leftarrow LS(S, N_k)$
6: **if** $S < S^*$ **then**
7: $S^* \leftarrow S$
8: $K \leftarrow 1$
9: **else**
10: $K \leftarrow K + 1$
11: **end if**
12: **end while**
13: **Return:** S^*

Neighborhoods. The neighborhoods used to solve this problem were Relocate and Swap. Relocate is performed between the same route and on other routes. In order to apply Relocate on other routes, we evaluate if the vehicle start time is greater than the order release time. When this occurs, Δ_A and Δ_B are determined as shown in Eqs. (17) and (18). In Fig. 2, node x , which belongs to route B and is located between nodes i and j (see graph 1), is relocated on route A between nodes k and l (see graph 2). To

compute the new CR_{Max} , Eq. (19) is used. If the movement is applied on the same route to update the CR_{max} , we use Eq. (20).

$$\Delta_A = T_{i,j} - T_{i,x} - T_{x,j} \tag{17}$$

$$\Delta_B = T_{k,x} + T_{x,l} - T_{k,l} \tag{18}$$

$$CR_{MAX} = \max\{CR_A + \Delta_A, CR_B + \Delta_B, \max\{CR_r\} \forall r \in R - \{A, B\}\} \tag{19}$$

$$CR_A = CR_A + \Delta_A + \Delta_B \tag{20}$$

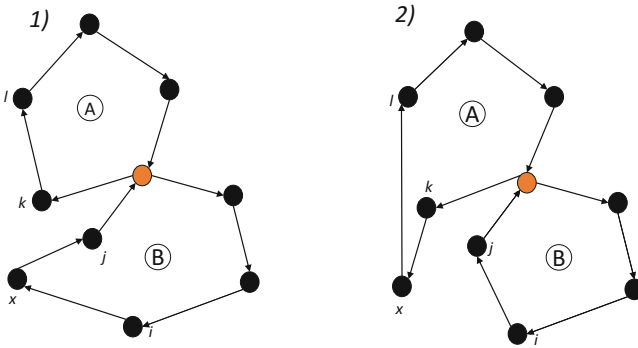


Fig. 2. Relocate

4 Computational Experiments and Results

The proposed solution approach was run on a PC with 4.1 GHz, 4 cores and 20 GB RAM. Since there are not benchmark instances available in the literature for the problem under study, experiments were carried out by adapting the VRP datasets from [18]. The mathematical model was coded in Python programming language and solved using Gurobi 9.12 and the solver CPLEX 20.10, while the algorithm was coded in Java language. To divide the routes, we established that each vehicle starts its delivery every 2 h from the factory.

4.1 Algorithm Calibration

In order to calibrate the parameters of the algorithm, a preliminary experiment was carried out using a sample of five (out of 30) instances. These instances were selected based on their size. The instances selected allowed to define the running time of the algorithm. Each of these instances was run for one hour and convergence curves were drawn (see Fig. 3). Most of the instances converge in less than one minute. However, instance TRM30 converges in around 2 min. Therefore, the running time for the other instances was set to 5 min to prevent losing good results.

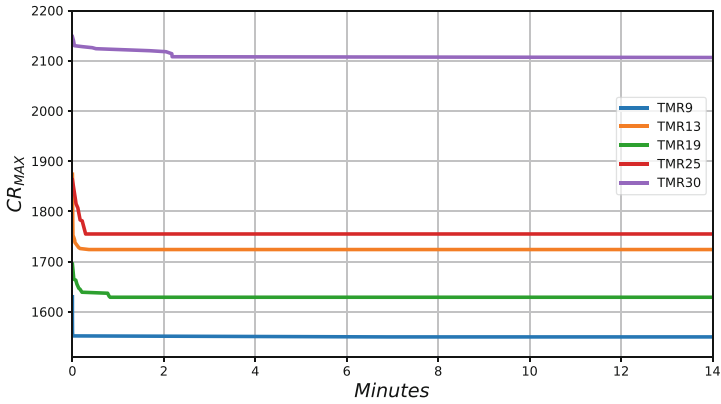


Fig. 3. Algorithm convergence

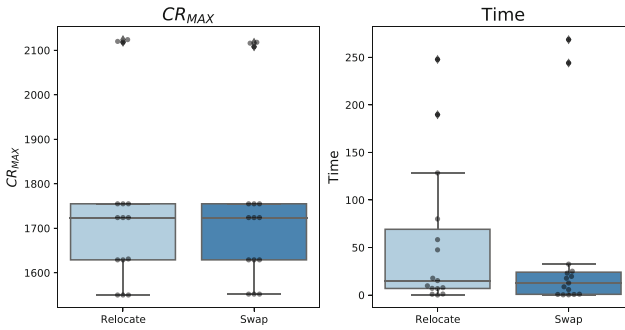


Fig. 4. Box-plot factorial experiment

A factorial experiment was also designed to define the order of execution of the two neighborhoods in the VND algorithm. This experiment is composed of two factors. The first factor evaluates which of the two neighborhoods should be executed first, while the second factor is the set of five instances previously selected. Each experiment was run for 5 min as defined above, and run using three replications to obtain statistical results. The best solution and the time required to reach it are both registered. Figure 4 contains the box plot for the CR_{MAX} and running time. In addition, we verified that output variable CR_{MAX} meets the assumptions of normality, stochasticity and independence. For this, we used Kolmogorov-Smirnov, Bartlett and Durbin-Watson tests. The sample gave us that the residuals do not behave normally, therefore a parametric test cannot be applied. Therefore, nonparametric Wilcoxon signed-rank test for related data was selected with 95% confidence level. It was obtained that the order of execution of the neighborhoods does not significantly influence CR_{MAX} , as illustrated in Fig. 4. Also, to identify if there is a significant difference around the running time of the algorithm, normality assumptions were tested. As a result, the sample errors with respect to time do not behave normally. Considering that the data are not related, the Wilcoxon test was hence performed, giving as a result that there is a significant difference in the time to

obtain the best solutions. To sum up, this experiment showed that there is no significant difference in the order of execution of the neighborhoods, but results are obtained faster if the swap neighborhood is executed before relocate. Therefore, this neighborhood configuration was used for the full computational experiment.

4.2 Computational Results

The computational results for the set of 30 instances are presented in Table 1. For each instance, five replications were performed. In Table 1 the number of clients of the

Table 1. Computations results

Instance	Total customers	CR_{MAX}				Time (seconds)			
		Min	Avg	Max	σ	Min	Avg	Max	σ
TRM0	20	1326	1326.00	1326	0.00	0.00	0.01	0.03	0.01
TRM1	50	938	938.00	938	0.00	12.32	68.20	173.95	63.36
TRM2	50	1111	1111.00	1111	0.00	0.02	0.14	0.39	0.15
TRM3	50	984	984.00	984	0.00	0.37	1.49	2.60	0.93
TRM4	50	1034	1034.00	1034	0.00	0.04	1.13	2.47	0.95
TRM5	50	1023	1023.00	1023	0.00	0.18	17.73	36.40	16.96
TRM6	75	1323	1324.20	1325	1.10	2.00	64.20	216.59	87.62
TRM7	75	1255	1255.00	1255	0.00	25.21	70.71	160.14	53.75
TRM8	75	1566	1568.40	1574	3.58	43.71	157.92	288.12	115.95
TRM9	75	1550	1551.60	1552	0.89	0.24	10.89	53.13	23.61
TRM10	75	1599	1599.40	1601	0.89	23.80	107.38	224.08	86.87
TRM11	100	1978	1992.80	2000	8.56	58.18	128.12	224.24	69.89
TRM12	100	1684	1687.60	1690	2.19	28.30	114.51	236.12	84.82
TRM13	100	1724	1724.00	1724	0.00	0.70	5.97	18.67	7.71
TRM14	100	2023	2025.80	2029	2.68	9.73	161.48	236.92	97.39
TRM15	100	1623	1625.80	1627	1.79	11.16	85.75	165.53	62.44
TRM16	50	1484	1484.00	1484	0.00	0.09	0.43	0.91	0.30
TRM17	50	1162	1163.60	1164	0.89	5.22	74.81	133.92	60.04
TRM18	50	1473	1473.00	1473	0.00	0.00	0.57	1.05	0.40
TRM19	50	1629	1629.00	1629	0.00	2.71	24.97	82.08	32.78
TRM20	50	1429	1429.00	1429	0.00	4.59	42.45	109.15	41.28
TRM21	75	2012	2016.40	2018	2.61	59.30	169.61	282.07	90.45
TRM22	75	1877	1882.20	1887	5.02	49.19	112.37	204.75	63.99
TRM23	75	1664	1664.00	1664	0.00	1.48	2.85	5.45	1.54
TRM24	75	1767	1767.00	1767	0.00	0.54	32.95	74.97	27.07
TRM25	75	1755	1755.00	1755	0.00	0.60	14.98	32.92	14.05
TRM26	100	1967	1970.20	1979	5.22	1.26	79.98	241.66	96.55
TRM27	100	1910	1914.40	1918	2.97	6.76	124.61	246.18	101.51
TRM28	100	1926	1930.00	1934	3.74	29.43	167.35	270.54	103.31
TRM29	100	1806	1809.60	1814	3.58	111.66	159.44	281.26	69.12
TRM30	100	2108	2112.00	2116	4.00	80.56	195.04	252.83	70.92

Table 2. ILS and CPLEX comparison

Instance	Best bound	Time	CPLEX Solution		ILS-VND	
			CR_{MAX}	GAP	CR_{MAX}	GAP
TRM0	1296	10080.3072	1326	2.26%	1326	2.26%
TRM1	830	10080.1479	944	12.08%	938	11.51%
TRM2	942	10080.4242	1111	15.21%	1111	15.21%
TRM3	860	10080.2357	984	12.60%	984	12.60%
TRM4	907	10080.1708	1034	12.28%	1034	12.28%
TRM5	891	10080.2108	1025	13.07%	1023	12.90%
TRM6	1153	10080.2576	1341	14.02%	1323	12.85%
TRM7	1117	10080.3813	1255	11.00%	1255	11.00%
TRM8	1404	10080.2377	1608	12.69%	1566	10.34%
TRM9	1369	10080.494	1560	12.24%	1550	11.68%
TRM10	1542	10080.1304	1599	3.56%	1599	3.56%
TRM11	1891	10080.7479	–	–	1978	4.40%
TRM12	1635	10080.3015	1686	3.02%	1684	2.91%
TRM13	1534	10080.2128	1748	12.24%	1724	11.02%
TRM14	1891	10080.3009	–	–	2023	6.52%
TRM15	1547	10080.3097	1739	11.04%	1623	4.68%
TRM16	1440	10080.1349	1484	2.96%	1484	2.96%
TRM17	1092	10080.168	1172	6.83%	1162	6.02%
TRM18	1473*	1242.91237	1473	0.00%	1473	0.00%
TRM19	1629*	133.738267	1629	0.00%	1629	0.00%
TRM20	1412	10080.113	1429	1.19%	1429	1.19%
TRM21	1993	10080.2151	–	–	2012	0.94%
TRM22	1871	10080.96	–	–	1877	0.32%
TRM23	1664*	694.355682	1664	0.00%	1664	0.00%
TRM24	1767*	1265.75726	1767	0.00%	1767	0.00%
TRM25	1738	10080.2264	1755	0.97%	1755	0.97%
TRM26	1937	10080.1774	1967	1.53%	1967	1.53%
TRM27	1819	10080.1982	1912	4.86%	1910	4.76%
TRM28	1855	10080.1998	2016	7.99%	1926	3.69%
TRM29	1712	10080.2067	1868	8.35%	1806	5.20%
TRM30	2078	10081.0267	–	–	2108	1.42%

instance is shown. The best solution of the algorithm, the average of the five replications, the worst solution found and the standard deviation for each instance are reported. Also, the table reports the lowest, the average and the highest response time for each instance. The first conclusion from these results is that the proposed algorithm has a

low variance. In addition, a comparison with the solution obtained with the mathematical model is presented (see Table 2). The execution time of the mathematical model was limited to three hours per instance. From the results returned by the solver, the table reports the relaxed lower bound, the integer feasible solution (if found), the GAP between them, and the running time. From these results, four optimal solutions were found (TRM18, TRM19, TRM23 and TRM24), while the solver could not find feasible solutions for instances TRM11, TRM14, TRM21, TRM22 and TRM30. For the solutions that are feasible, the most complex instances are TRM0, TRM1, TRM2, TRM3, TRM4, TRM5, TRM6, TRM7, TRM8 and TRM9.

Comparing the feasible results returned by the ILS-VND algorithm with the solver, it is observed that the algorithm in a short period of time yields to better solutions. In addition, we observe that the algorithm is able to find optimal solutions as it finds the optimum of the solutions TRM18, TRM19, TRM23 and TRM24. Moreover, for several instances (TRM0, TRM12, TRM16, TRM20, TRM21, TRM22, TRM25, TRM26 and TRM27) the gap is even lower than 3%, which is very good performance. Although for the most complex instances outlined above, the gap with respect to the relaxed lower bound is very high, and in most cases it outperforms the solutions found by the solver.

5 Conclusions and Perspectives

This paper proposed a mathematical model and a hybrid algorithm to solve the vehicle routing problem with order release times. This problem, inspired by practical real-life situations of production-distribution systems, has been rarely studied in the literature, making this paper one of the first contributions. The proposed algorithm consists on an Iterated Local Search and a Variable Neighborhood Descent. Computational experiments were carried out using adapted benchmark instances from the literature. Experimental results are promising and motivate further studies about this problem.

Following the findings of this work, several opportunities for further research are identified. For example, several parameter sets of the metaheuristic can be tested, followed by the design of a solution procedure that addresses the joint production and routing problem. It would also be interesting to formulate a multi-objective algorithm, to address also environmental or social performance indicators, in addition to routing costs or times. In addition, other constraints could be added to the problem, such as time windows, service times, vehicle capacity, etc.

Acknowledgements. The work presented in this paper was supported by Universidad de La Sabana, Colombia, under a postgraduate scholarship from the School of Engineering awarded to the first author, and by internal research funds under project INGPLD-10-2019.

References

1. Archetti, C., Feillet, D., Gendreau, M., Grazia Speranza, M.: Complexity of the VRP and SDVRP. *Transp. Res. Part C: Emerg. Technol.* **19**(5), 741–750 (2011)
2. Ballou, R.: *Logística. Administración de la cadena de suministro*, Pearson Educación (2004)
3. Cordeau, J.F., Laporte, G.: The dial-a-ride problem: models and algorithms. *Ann. Oper. Res.* **153**(1), 29–46 (2007)

4. Díaz-Madroñero, M., Mula, J., Peidro, D.: A mathematical programming model for integrating production and procurement transport decisions. *Appl. Math. Model.* **52**, 527–543 (2017)
5. Eksioğlu, B., Vural, A.V., Reisman, A.: The vehicle routing problem: a taxonomic review. *Comput. Ind. Eng.* **57**(4), 1472–1483 (2009)
6. Garey, M., Johnson, D.: *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman, Mathematical Sciences Series (1979)
7. Gendreau, M., Laporte, G., Potvin, J.Y.: Metaheuristics for the capacitated vrp. *Discrete Mathematics and Applications*, pp. 129–154 (2002)
8. Irnich, S., Toth, P., Vigo, D.: Chapter 1: The family of vehicle routing problems. *Vehicle Routing*, pp. 1–33
9. Laporte, G.: The vehicle routing problem: an overview of exact and approximate algorithms. *Eur. J. Oper. Res.* **59**(3), 345–358 (1992)
10. Martins, L.d.C., Gonzalez-Neira, E.M., Hatami, S., Juan, A.A., Montoya-Torres, J.R.: Combining production and distribution in supply chains: The hybrid flow-shop vehicle routing problem. *Comput. Ind. Eng.* **159**, 107486 (2021)
11. Mentzer, J.T., Flint, D.J., Hult, G.T.M.: Logistics service quality as a segment-customized process. *J. Mark.* **65**(4), 82–104 (2001)
12. Min, H., Zhou, G.: Supply chain modeling: past, present and future. *Comput. Ind. Eng.* **43**(1), 231–249 (2002)
13. Montoya-Torres, J.R., López Franco, J., Nieto Isaza, S., Felizzola Jiménez, H., Herazo-Padilla, N.: A literature review on the vehicle routing problem with multiple depots. *Comput. Ind. Eng.* **79**, 115–129 (2015)
14. Mourgaya, M., Vanderbeck, F.: The periodic vehicle routing problem: classification and heuristic. *RAIRO - Oper. Res.* **40**(2), 169–194 (2006)
15. Psaraftis, H.N.: Dynamic vehicle routing: status and prospects. *Ann. Oper. Res.* **61**(1), 143–164 (1995)
16. Shapiro, L.: The embodied cognition research programme. *Philos Compass* **2**(2), 338–346 (2007)
17. Solina, V., Mirabelli, G.: Integrated production-distribution scheduling with energy considerations for efficient food supply chains. *Procedia Comput. Sci.* **180**, 797–806 (2021)
18. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper. Res.* **35**(2), 254–265 (1987)
19. Thomas, D.J., Griffin, P.M.: Coordinated supply chain management. *Eur. J. Oper. Res.* **94**(1), 1–15 (1996)
20. Torres-Tapia, W., Montoya-Torres, J.R., Ruiz-Meza, J., Belmokhtar-Berraf, S.: A matheuristic based on ant colony system for the combined flexible jobshop scheduling and vehicle routing problem. In: 10th IFAC Conference on Manufacturing Modelling, Management and Control. Elsevier, Nantes, France (2022)