Networks

WILEY

SPECIAL ISSUE ARTICLE

A Multi-Trip Vehicle Routing Problem With Release Dates and Interrelated Periods

Raquel Bernardino[1,2] | João Janela[1,2] | Carlos Martins[1,2] | Maria Cândida Mourão[1,2] | Leonor Santiago Pinto[1,2] | Filipe Rodrigues[1,2]

[1]ISEG/UL - Lisbon School of Economics & Management, Universidade de Lisboa, Lisbon, Portugal | [2]REM - Research in Economics and Mathematics, CEMAPRE, Lisbon, Portugal

Correspondence: Raquel Bernardino (rbernardino@iseg.ulisboa.pt)

Received: 2 January 2024 | **Revised:** 7 October 2024 | **Accepted:** 11 October 2024

Funding: This work was partially supported by the Project CEMAPRE/REM - UIDB/05069/2020, financed by FCT - Fundação para a Ciência e Tecnologia through national funds.

Keywords: interrelated periods | matheuristic | multi-trip vehicle routing problem | release dates

ABSTRACT

This article proposes a new multi-trip vehicle routing problem variant, originally motivated by a practical application, namely a car components distribution to repair centers (warehouses). The problem is named the multi-trip vehicle routing with release dates and interrelated periods (MTVRP-RDIP). The routes may start at different pre-defined periods, called departure periods, and may have different durations. Thus, two routes that start at different periods may be active at the same period, leading to the so-called interrelated periods. Moreover, a route may only start at a period if a vehicle is available, and the availability of the vehicle depends not only on the existing vehicles but also on the active routes at that time period. The clients are classified according to their importance and represent warehouses that require car components that are made available throughout the time horizon, thus leading to the consideration of release dates. Delays in satisfying client orders and not serving orders in the time horizon result in penalty costs that must be minimized. The objective to minimize also includes routing costs and vehicle utilization costs. Two metrics are defined to compute an order's delay, each leading to a different mixed integer linear model. Computational results showed that one model clearly outperforms the other and even this one is only suited to address the smallest instances. A matheuristic based on a rolling-horizon process that iteratively solves the better-performing model with fewer periods was designed to tackle the largest instances. The matheuristic can provide feasible solutions for all test instances in an efficient manner that are, on average, better than the ones provided by the model in most of the compared key performance indicators.

1 | Introduction

This article studies a problem originally motivated by an auto supply components distributing company that seeks to optimize its delivery routes departing and returning to a depot. Each vehicle can perform several routes during the time horizon. Multiple vehicles of different types exist and may be available to perform routes with varying durations that can only leave the depot at specific times called departure periods. The company's logistics justify the existence of departure periods to avoid crowds during order preparation and vehicle loading, allow the distribution of orders at different times, and consider the different schedules of the drivers. Because of these operational constraints the time horizon is divided into periods, and the duration of a route represents the number of periods the route lasts. The duration of the longest route is the maximum number of consecutive periods that a driver can work without a break.

Clients are classified according to their importance and represent warehouses that place orders throughout the time horizon. The components ordered by a client are grouped into boxes in a specific area of the company's warehouse, using the existing stock, or, as frequently happens, new parts are purchased. Thus, the existence of release dates, that is, the date in which orders are ready to leave the warehouse, is a critical feature of the routing problem considered. Additionally, delays in satisfying clients' orders are admitted under penalties. Clients' orders may not be served during the time horizon. In that case, the orders are considered unserved with high penalty values. The company aims to minimize operating costs by considering fixed costs of using vehicles, variable costs depending on the distance traveled, and penalties for delays in the distribution of orders and for unserved orders.

Several decisions must be taken at each departure period, which are known previously and represent periods in the time horizon when vehicles may start routes. Therefore, when planning routes, the duration, number, vehicles to use, clients' sequence at each route, and unserved orders must be decided. Since there are routes of varying durations and several departure periods, the periods are interrelated, as in Example 1. In fact, decisions in each period depend on the decisions taken in previous periods. Hence, this problem can be seen as a multi-trip vehicle routing with release dates and interrelated periods (MTVRP-RDIP).

Example 1. Assume that the time horizon is divided into ten periods, including five departure periods $(1, 2, 4, 6, \text{ and } 8)$ and two vehicles. Consider also a solution where both vehicles perform three routes. Vehicle 1 performs one route that lasts two periods and two lasting three periods, while the first route of vehicle 2 lasts one period only, and the other two three periods, as depicted in Figure 1. As routes may only start at the beginning of a departure period, the first route chosen for vehicle 1 invalidated the generation of a new route for this vehicle at departure period number 2, which does not happen for vehicle 2. Thus, at the second departure period, only vehicle 2 is available. Therefore, decisions made in the first departure period will affect the ones that could be made in the following departure periods.

The MTVRP-RDIP is a problem worth studying due to three main reasons. First, it is a challenging problem because it is a variant of an NP-hard problem, namely the vehicle routing problem. Second, as far as we know, this problem is here defined for the first time, and thus, there are no mathematical formulations for it. Third, the MTVRP-RDIP, motivated by a practical application, may be applied to other real-world problems involving product distribution with release dates and interrelated periods.

The main contributions of this article are:

- the proposal of the multi-trip VRP with release dates and interrelated periods to address a distribution problem;
- mixed integer linear programming (MILP) models for the problem that are computationally compared;
- a matheuristic based on a rolling-horizon heuristic to generate feasible solutions for the biggest instances;
- a comparison of the proposed methods through stakeholders' defined key performance indicators (KPIs).

This article is organized as follows. After the introduction, including the aims and scope of this article, the problem is formally defined in Section 2. A literature review gives shape to the third section, before the proposed models, which are described in Section 4. Section 5 details the matheuristic developed, followed by the analysis of the computational results in Section 6. Finally, the conclusions end the article in Section 7.

2 | The Multi-Trip Vehicle Routing Problem With Release Dates and Interrelated Periods (MTVRP-RDIP)

As mentioned in Section 1, the objective is to optimize delivery routes departing and returning to a depot. As mentioned above, distribution routes that can begin at different times, the departure periods, during a given time horizon (e.g., a day). The time horizon is thus divided into periods of the same duration (α, in time units). The set of periods is $P = \{1, \dots, p, \dots, |P|\}$, which includes the set of departure periods $DP \subseteq P$.

At each departure period, more than one vehicle may be available to perform routes that may have different durations, that is, different types that define the set R. Thus, a type $r \in R$ route is a route that lasts r periods. The duration of the longest route is the maximum number of consecutive periods that a driver can work without a break.

Clients are warehouses placing orders which are released from the warehouse's company throughout the time horizon. Moreover, clients are classified into two segments, I and II, according to their importance. To simplify, the type of an order is defined as the type of the client placing the order. The company seeks to serve clients' orders I and II within a given time interval (in time periods) after they are released; however, delays are admitted under penalties. So, no penalty is charged if a client is served within the defined time interval. Clients in Segment I allow shorter delays than those of Segment II, and delays on orders from Segment I clients are penalized more heavily. Segment III represents unserved orders from the previous time horizon.

The distribution network is modeled by a directed graph, $G = (V, A)$, where $V = O \cup \{0\}$ is the node set to represent clients' orders O and the depot, node 0. Links between clients and the depot define the arc set $A = \{a = (i, j) \in V \times V : i \neq j\}$. Since O represents the orders, it includes as many replicas of each client as the number of orders released during the time horizon. Arcs between orders of the same client are defined in only one

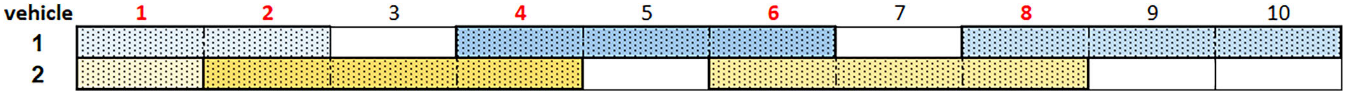

FIGURE 1 | Interrelated periods.

direction to remove symmetry. Thus, if $i, j \in O$ are orders of the same client, the only arc included in A is $(i, j) \in A$, with $i < j$. To simplify the notation, we still refer to the arc set without the mentioned arcs as A.

Additionally, set O is partitioned into three disjoint sets to distinguish the respective clients' segments, $O = O_I \cup O_{II} \cup O_{III}$. While O_I includes orders from clients of segment I and O_{II} from segment II, set O_{III} includes orders left unserved during the previous time horizon.

Each order $i \in O$ is released (i.e., ready for delivery) from the company's warehouse, in a known period $\delta_i \in P$. Thus, order i can only be served after period δ_i. Let π_i be the first departure period after δ_i. To simplify the ensuing explanations, set $B_p = \{i \in O : \delta_i < p\}, \forall p \in DP$ identifies the orders that are released before time period p, and consequently that can be served in any route that starts at or after $p \in DP$. The correspondent set of arcs is denoted by $A_p = \{(i, j) \in A : i, j \in B_p \cup \{0\}\}, \forall p \in DP$. Period $\delta_i = 0$ is used for orders $i \in O_{III}$ left unserved during the previous time horizon.

Each order $i \in O$ has a service time at the correspondent client, denoted by σ_i (in time units), and the time to traverse arc $(i, j) \in A$ is τ_{ij} (in time units). Moreover, any order $i \in O$ served (or delivered) to the client within θ_i periods after release is considered without delay. Nevertheless, orders can be served with delay, and two metrics are used to compute an order's delay, namely the exact period delay (EPD) and the start period delay (SPD). The EPD corresponds to the difference between the period in which a client is served and the last period it could be served without delay. In contrast, the SPD corresponds to the difference between the departure period in which the route that serves a client leaves the depot and the last period it could be served without delay, as illustrated in Example 2. These two metrics motivated the two mathematical models presented in Sections 4.1 and 4.2.

A penalty of $\beta_\#$ per period of service delay for type $\#(\# = I, II, III)$ order must be payed. Each order from clients of segment $\#$ that will be left unserved during the present time horizon has an associated penalty of $\eta_\#$, with $\eta_\# >> \beta_\#$.

Set K includes the existing heterogeneous vehicles that may be used. Even if the vehicles have the same dimensions, the cost of using them may differ, for example, due to their age. Thus, a fixed cost f^k of using vehicle $k \in K$ is considered. Arcs traversal costs, c_{ij}^k, are also influenced by several factors, such as the driver assigned to it.

The defined notation and main parameters are summarized in Table 1.

Example 2 is next presented to illustrate both the MTVRP-RDIP and a feasible solution for it.

Example 2. Consider that in Example 1 six clients placed orders that are released as characterized in Table 2. The first rows of the table identify the orders that were left unserved in the previous time horizon (thus are orders of segment III). For example, client C1 from segment I placed two orders: orders O4 and O5, that are released in periods one and three, respectively. Additionally, client C1 has an order left unserved from the previous time horizon: order O1. Note that an order released in a period p can be served in any period after p.

A feasible solution for this example is obtained by designing routes for the vehicles assigned to the services according to Figure 2. Vehicle 1 is assigned to three routes, the first of type 2 and the other two of type 3, starting at departure periods 1, 4, and 8, while vehicle 2 must perform one type 1 route, starting at the first departure period, and two type 3 routes, starting at departure periods 2 and 6. During its first route, vehicle 1 visits

TABLE 1 | Notation and parameters.

Notation	Definition	Notation	Definition
$\lvert \bullet \rvert$	Cardinality of set $\bullet$	K	Set of vehicles ($k \in K$)
$\lceil \bullet \rceil$	Nearest integer greater than or equal to $\bullet$	R	Set of the types of routes ($r \in R$)
$O_\#$	Set of orders of type # ($\# = I, II, III$)	r	Number of periods that type $r \in R$ route lasts
O	Set of all orders, $O = O_I \cup O_{II} \cup O_{III}$	α	Duration (in time units) of each time period
0	Depot node	θ_i	Maximum allowed delay (in periods) to serve $i \in O$ with no penalty
$G = (V, A)$	Distribution directed network	δ_i	Period in which order $i \in O$ is released
V	Set of nodes ($V = \{0\} \cup O$)	σ_i	Service time of order $i \in O$ in time units
A	Set of arcs $A = \{(i, j) \in V \times V : i \neq j \wedge i < j$ if i, j are orders of the same client$\}$	π_i	First departure period after order $i \in O$ is released
P	Set of time periods ($p \in P$)	τ_{ij}	Traversal time of $(i, j) \in A$ in time units
DP	Set of departure periods ($DP \subseteq P$)	f^k	Fixed cost of using vehicle $k \in K$
B_p	Set of orders released before period $p \in DP$ $B_p = \{i \in O : \delta_i < p\}$	c_{ij}^k	Cost of traversing $(i, j) \in A$ with $k \in K$
A_p	Set of arcs between orders of B_p, $p \in DP$	$\eta_\#$	Penalty to not serve an order of a type #
		$\beta_\#$	Penalty per period delay of a type #

clients C1 and C3 to meet orders O1 and O2. During its second route, it serves orders O5, O7, O9, and O10, all released until the third period, from clients C1, C2, C3, and C4, respectively. The third route is used to fulfill orders O13, and O14 from client C5 and O16 from client C6. In this example, although all the orders are met during the time horizon, some are served with delay (e.g., order O12 is released at period 3 and served in a route starting at the departure period 6). Note that, the duration of the routes generated must be compatible with the respective route types.

As mentioned above, two metrics are used to compute the delays, EPD and SPD. Consider once more, order O12, released in period 3 and served in period 8, in a route that starts at period 6. Suppose that the maximum allowed delay is two periods ($\theta_{12} = 2$). Thus, order O12 exact period delay is $EPD = 8 - (\delta_{12} + \theta_{12}) = 8 - (3 + 2) = 3$. However, its SPD is computed considering the departure period in which the route that serves order O12 starts, period 6, and is equal to SPD $= 6 - 5 = 1$.

Hence, the multi-trip VRP with release dates and interrelated periods (MTVRP-RDIP) simultaneously decides which vehicles are used at each departure period, the type of routes performed by each vehicle, the orders to deliver in each route and, consequently, the orders to be fulfilled on time, those fulfilled with delay, and those that are left unserved. The goal is to minimize the total operating cost resulting from the fixed costs of using the vehicles, the variable costs depending on the traveled distances, and the penalties resulting from delays and unserved orders.

The defined problem is formulated with two MILP models, as described in Section 4.

TABLE 2 | Characteristics of clients and respective orders.

Client	Segment	No. of orders	Order	Released in period
C1	III	1	O1	0
C3	III	1	O2	0
C6	III	1	O3	0
C1	I	2	O4	1
			O5	3
C2	I	3	O6	1
			O7	2
			O8	5
C3	I	1	O9	3
C4	I	2	O10	2
			O11	4
C5	II	3	O12	3
			O13	5
			O14	7
C6	II	2	O15	1
			O16	6

3 | Literature Review

The vehicle routing problem (VRP) was introduced by Dantzig and Ramser [1], and since then, it has become one of the most studied problems worldwide. The vast practical application is one of the facts that justifies the importance given to VRPs. As a result, several VRP variants have been studied over the years to accommodate different practical requirements (see, e.g., [2–8]).

Since the literature on VRPs is extensive, we will just focus on the literature on VRPs with release dates (VRP-RD), which is one of the main features of the proposed problem. A survey on multi-trip VRPs—another essential feature of MPVRP-RDIP—can be found in [9]. Contrary to traditional VRPs, in the VRP-RD the goods are not initially available at the depot but arrive over time. Release dates can be either deterministic or uncertain, and both cases fit different practical applications. This article focuses on a VRP with deterministic release dates since they are known in advance, that is when the planning is made. This is not the case of uncertain release dates, as, for instance, [10–14].

The VRP-RD was first introduced by Cattaruzza, Absi, and Feillet [15]. The authors proposed a genetic algorithm hybridized with local search to solve the multi-trip VRP-RD with time windows. The complexity of the VRP-RD and the VRP-RD with due dates, where clients must be visited between the release and due dates, was studied in [16, 17], respectively. Shelbourne [18] and Shelbourne, Battarra, and Potts [19] integrated aspects of machine scheduling into the VRP-RD and solved the problem using a path-relinking algorithm enhanced with a neighborhood search.

More recently, VRPs have emerged within e-commerce. Liu, Li, and Liu [20] solved the problem by a Tabu search algorithm enhanced with a local search procedure, and a Lagrangian relaxation algorithm was used to derive lower bounds. This work was extended by Li et al. [21] to allow multiple trips of vehicles. They proposed an adaptive large neighborhood search enhanced with a labeling procedure to solve the problem. Also, in the

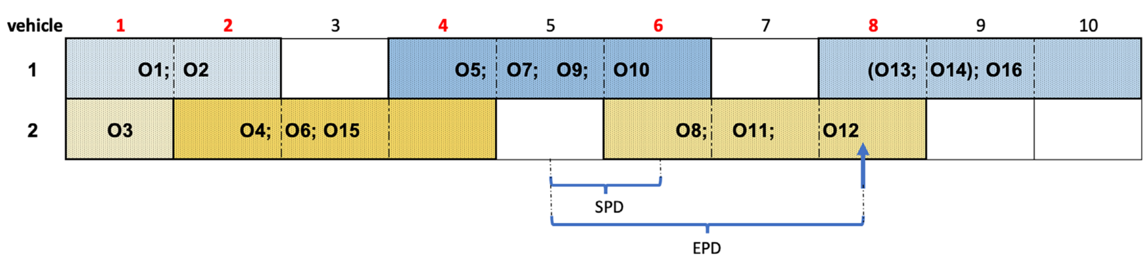

FIGURE 2 | Assignment of vehicles to routes and services.

e-commerce context, Moons et al. [22] integrated the VRP-RD with the order-picking problem.

Zhen et al. [23] were the first to study the multi-depot, multi-trip VRP-RD, where each depot has its own fleet of vehicles. The problem is solved by both a particle swarm optimization and a genetic algorithm. Li et al. [24] studied the VRP-RD arising in an electronic home appliances company where product and service delivery are considered simultaneously. In this problem, each vehicle is associated with a driver and a technician who must be able to serve the assigned clients. An adaptive large neighborhood search is designed to solve this problem. Sun, Li, and Li [25] studied the VRP-RD with flexible time windows in the context of fresh food delivery. The authors use an exact branch-and-cut algorithm to solve the problem and propose several valid inequalities for it.

All the previously mentioned works focus on VRP-RD where the vehicles serve the clients in a single period, for instance, a day. This differs from the multi-period VRP-RD, where a set of periods is considered, and delivery plans are made for each period. However, in the multi-period VRP-RD, each route is confined to a single period. Thus, no interrelated periods are considered. The multi-period VRP-RD was first studied by Archetti, Jabali, and Speranza [26]. They developed three different formulations and several valid inequalities for the problem, which were compared through computational results. The models consider due dates, holding costs at the depot, and penalties for unserved clients. However, the models do not specify the exact moment each client is served (the moment the products are distributed to the client), but only the period in which the service takes place. Later, Larrain et al. [27] proposed a new algorithm for this problem, iterating between two phases: an exact solution phase (branch-and-bound) and a local search phase (variable neighborhood search). The key idea of this algorithm is to suspend the execution of the branch-and-bound when it reaches a new upper bound and then apply a variable neighborhood search algorithm to the current solution. The branch-and-bound is then resumed after the local search phase, and the process is repeated. Soman and Patil [28] also solved a multi-period VRP-RD with departure periods coinciding with the release date periods by a greedy randomized adaptive search procedure. This work was later extended in [29] by considering a heterogeneous fleet and the possibility of not serving clients due to storage capacity. A scatter search method is proposed to solve this generalized version. In the pharmaceutical distribution sector context, Campelo et al. [30] proposed the consistent VRP-RD where client satisfaction through a consistent service is required. A fix-and-optimize algorithm is used to identify a set of fixed routes to be repeated at each period, which is the main objective of the problem.

A common feature of all previously mentioned works is that the vehicles can only leave the depot after the release date associated with the clients they will serve. Recently, [31] studied the traveling salesman problem with release dates (TSP-RD) and drone resupply, where drones are used to resupply the en-route vehicle with orders (products) with release dates after the vehicle's departure time. The problem is solved by a decomposition algorithm. The TSP-RD was previously studied by Archetti et al. [32] where a single vehicle is allowed to perform multiple trips and an iterated local search algorithm is proposed.

Table 3 presents a summary of the main characteristics of the VRPs and multi-period VRPs (MPVRPs) with deterministic release dates studied in the literature, as well as the

TABLE 3 | Summary of the studies in VRPs and multi-period VRPs with deterministic release dates.

		Features considered						Optimized objectives						
	Paper	**IP**	**MT**	**RL**	**DD**	**VC**	**TW**	**RC**	**FC**	**WT**	**HC**	**PD**	**DT**	**UC**
VRP-RD	[15]		✓			✓	✓	✓						
	[19]				✓	✓		✓				✓		
	[20]					✓		✓		✓				
	[32]		✓										✓	
	[22]			✓		✓	✓	✓	✓					
	[21]		✓			✓							✓	
	[23]		✓			✓	✓	✓						
	[24]					✓	✓	✓						
	[31]												✓	
	[25]					✓	✓		✓					
MPVRP-RD	[26]				✓	✓	✓	✓			✓			✓
	[27]				✓	✓	✓	✓			✓			✓
	[30]			✓		✓	✓	✓						
	[28]				✓	✓		✓			✓	✓		
	[29]				✓	✓		✓			✓	✓		✓
	This article	✓	✓	✓	✓			✓	✓			✓	✓	✓

Note: **Features**={IP-interrelated periods, MT-multiple trips, RL-routes' time limit, DD-due dates, VC-vehicles' capacity, TW-time windows}. **Objectives**={RC-routing cost (travel time distance or cost), FC-vehicles' fixed cost, WT-waiting times at the depot, HC-holding cost, PD - penalties for delays, DT-delivery time, UC-unserved clients}.

characteristics of the MTVRP-RDIP proposed in this article. As shown in Table 3, few studies consider deterministic release dates, and the existing ones are recent. To the best of our knowledge, our article is the first to consider a VRP-RD with interrelated periods. It also considers multi-trips and several types of routes that differ in the time they last. Routes with maximum durations are only considered in [22, 30], but multi-trips are not allowed. Multi-trips are addressed in [21, 23] for VRPs-RD, but in both papers, a continuous time is considered and neither departure periods nor different type routes are considered. Multi-trips are also contemplated in [32] but for the TSP-RD. In [28, 29], a set of departure periods coinciding with the set of all release dates is assumed; however, the number of vehicles available at the depot in each period is a model parameter, which means that neither multi-trips nor interrelated departure periods are tackled.

4 | Mixed Integer Linear Models

This section presents two MILP models to solve the MTVRP-RDIP. The exact period delay model (EPDM) uses metric EPD to compute the orders' delay, while the start period delay model (SPDM) uses metric SPD.

EPDM and SPDM are used to generate feasible solutions for the MTVRP-RDIP problem, which are compared through some KPIs in Section 6. The reason for developing models based on two distinct delay metrics is the poor performance of the EPDM.

4.1 | Exact Period Delay Model (EPDM)

The first model presented aims to measure the exact period delay associated with each order since it was designed to determine the exact time a client is served. The EPDM uses binary variables to associate clients' orders with routes and vehicles and continuous variables to determine the time in which the clients are served (in time units). The detailed description of all decision variables used in the EPDM is as follows.

- $x_{ij}^{krp} = 1$ if arc $(i,j) \in A_p$ is traversed by vehicle $k \in K$ in a type $r \in R$ route starting at period $p \in DP$; and 0 otherwise.
- $y_i^{krp} = 1$ if order $i \in B_p$ is served by vehicle $k \in K$ in a type $r \in R$ route starting at period $p \in DP$; and 0 otherwise.
- $v^{krp} = 1$ if vehicle $k \in K$ is used in a type $r \in R$ route starting at period $p \in DP$; and 0 otherwise.
- $u^k = 1$ if vehicle $k \in K$ is used to perform at least one route during the time horizon; and 0 otherwise.
- q_i instant, in time units, in which order $i \in O$ starts to be served.
- d_i service delay, in periods, from the last period at which order $i \in O$ can be served without penalty (i.e., $\theta_i + \delta_i$) to the period where order i is served.
- h_i auxiliary variables used to linearize $\lfloor \frac{q_i}{\alpha} \rfloor$, that is, $h_i = \lfloor \frac{q_i}{\alpha} \rfloor$, for $i \in O$.

The EPDM for the VRP-RDIP can then be formulated as follows.

$$\min \sum_{k\in K}\left(f^k u^k + \sum_{r\in R}\sum_{p\in DP}\sum_{(i,j)\in A_p} c_{ij}^k x_{ij}^{krp}\right) + \sum_{\#\in\{I,II,III\}}\sum_{i\in O_\#}\left[\beta_\# d_i + \eta_\#\left(1 - \sum_{k\in K}\sum_{r\in R}\sum_{p\in DP} y_i^{krp}\right)\right] \tag{1}$$

subject to:

$$\sum_{j:(i,j)\in A_p} x_{ij}^{krp} = \sum_{j:(j,i)\in A_p} x_{ji}^{krp} \quad k\in K,\ r\in R,\ p\in DP,\ i\in B_p\cup\{0\} \tag{2}$$

$$\sum_{j:(i,j)\in A_p} x_{ij}^{krp} = y_i^{krp} \quad k\in K,\ r\in R,\ p\in DP,\ i\in B_p \tag{3}$$

$$\sum_{k\in K}\sum_{r\in R}\sum_{p\in DP:p>\delta_i} y_i^{krp} \le 1 \quad i\in O \tag{4}$$

$$\sum_{j:(0,j)\in A_p} x_{0j}^{krp} = v^{krp} \quad k\in K,\ r\in R,\ p\in DP \tag{5}$$

$$\sum_{r\in R} v^{krp} \le 1 - \sum_{p'\in DP:p'<p}\ \sum_{r'\in R:r'+p'>p} v^{kr'p'} \quad k\in K,\ p\in DP \tag{6}$$

$$y_i^{krp} \le v^{krp} \quad k\in K,\ r\in R,\ p\in DP,\ i\in O \tag{7}$$

$$\sum_{r\in R} v^{krp} \le u^k \quad k\in K,\ p\in DP \tag{8}$$

$$\sum_{(i,j)\in A_p} \tau_{ij} x_{ij}^{krp} + \sum_{i\in B_p}\sigma_i y_i^{krp} \le \alpha r \quad k\in K,\ r\in R,\ p\in DP \tag{9}$$

$$q_i \ge \sum_{k\in K}\sum_{r\in R}\sum_{p\in DP}(\alpha(p-1)+\tau_{0i})x_{0i}^{krp} \quad i\in O \tag{10}$$

$$q_j \ge q_i + (\tau_{ij}+\sigma_i)\sum_{k\in K}\sum_{r\in R}\sum_{p\in DP} x_{ij}^{krp} - \alpha|P|\left(1 - \sum_{k\in K}\sum_{r\in R}\sum_{p\in DP} x_{ij}^{krp}\right) \quad i,j\in O:(i,j)\in A \tag{11}$$

$$q_j \le q_i + (\tau_{ij}+\sigma_i)\sum_{k\in K}\sum_{r\in R}\sum_{p\in DP} x_{ij}^{krp} + \alpha|P|\left(1 - \sum_{k\in K}\sum_{r\in R}\sum_{p\in DP} x_{ij}^{krp}\right) \quad i,j\in O:(i,j)\in A \tag{12}$$

$$d_i \ge h_i - (\delta_i + \theta_i) \quad i\in O \tag{13}$$

$$h_i \le \frac{q_i}{\alpha} \le h_i + 1 \quad i\in O \tag{14}$$

$$d_i \ge (\pi_i - \delta_i - \theta_i)\sum_{k\in K}\sum_{r\in R}\sum_{p\in DP:p>\delta_i} y_i^{krp} \quad i\in O:\pi_i \ge \delta_i + \theta_i \tag{15}$$

$$x_{ij}^{krp}\in\{0,1\} \quad k\in K,\ r\in R,\ p\in DP,\ (i,j)\in A_p \tag{16}$$

$$y_i^{krp} \in \{0,1\} \qquad k \in K,\ r \in R,\ p \in DP,\ i \in B_p \tag{17}$$

$$v^{krp} \in \{0,1\} \qquad p \in DP,\ k \in K,\ r \in R \tag{18}$$

$$u^k \in \{0,1\} \qquad k \in K \tag{19}$$

$$d_i, h_i \geq 0 \text{ and integer} \quad i \in O \tag{20}$$

$$q_i \geq 0 \qquad i \in O \tag{21}$$

The objective (1) is to minimize the total cost, which is comprised of the fixed cost of using vehicles, the vehicles' travel cost, the penalization of orders met after their admissible delay, and the penalization of orders not fulfilled within the time horizon. Constraints (2) impose the equilibrium of nodes in terms of the indegree and the outdegree, while constraints (3) ensure that if a client order is served by a vehicle performing a route of a given type leaving at a departure period, then that vehicle traverses an arc leaving that client during the same route. The latter constraints also show that variables y are auxiliary. Each client order is served at most once by constraints (4). The identification of the vehicles used at each departure period by route type is given by constraints (5), while constraints (6) guarantee that only vehicles finishing their routes until departure period p can be used in that departure period. Note that an existing vehicle is available in a departure period if it has never left the depot or returned to it before that departure period. This means that the departure period in which it left plus the route duration is less than or equal to the current period. Constraints (7) ensure that clients' orders are only served by the vehicles in use, whereas constraints (8) are needed to charge the fixed costs properly. The time limit for each vehicle performing a type of route leaving at each departure period is imposed by constraints (9). Constraints (10–12) define the instant to start to serve each order. More precisely, constraints (10) impose a lower bound for the moment the first order of each route is served, which is given by the time at the beginning of the departure period plus the travel time between the depot and the correspondent client. Constraints (11) and (12) define the moment to serve an order (j) as the sum between the moment the previous order (i) is served and the travel time between the corresponding clients. These constraints are redundant if order j is not served immediately after order i. As usual, these constraints also forbid subtours, that is, tours not including the depot. Constraints (13) and (14) define the service delay in terms of periods, while (15) relate the service delay with the first departure period that may be used whenever it is after the release date plus the order' allowed delay. Constraints (15) are not necessary for defining a valid formulation for the MTVRP-RDIP but define a lower bound on the value of the d variables, increasing the linear programming relaxation value. Finally, the domains of variables are set through (16–21).

We included in the model standard symmetry-breaking constraints for the vehicles that state that the kth-vehicle can only be used if the $(k-1)$th-vehicle was used. These constraints are only valid for the first departure period and for identical vehicles, that is, vehicles with the same fixed cost and travel cost.

Preliminary computational experiments showed that constraints (7), although not needed to define a valid model for the MTVRP-RDIP, significantly improve the linear programming relaxation value and thus are included in the model.

4.2 | Start Period Delay Model (SPDM)

As will be seen in the computational results, the EPDM cannot even address the smallest size instances because of the variables needed to compute the orders' delay by the EPD metric. Hence, in this section, we propose the start period delay model (SPDM), where the delay associated with each order is computed by the SPD metric.

The SPDM uses variables $x_{ij}^{krp}, y_i^{krp}, v^{krp}$, and u^k introduced in the previous section and three new sets of variables next defined.

- z_{ij}^p flow that traverses the arc $(i,j) \in A_p$, which corresponds to an upper bound on the number of orders that still have to be served in period $p \in DP$.
- s_i departure period in which the route that serves order $i \in O$ starts ($s_i \in DP$).
- $\bar{d}_i$ service delay in periods from the last period at which order $i \in O$ can be served without penalty (i.e., $\theta_i + \delta_i$) to the period in which the route that serves order i leaves the depot (s_i).

Both models share constraints (2–9) and (16–19). The SPDM can then be formulated as follows.

$$\min \sum_{k\in K}\left(f^k u^k + \sum_{r\in R}\sum_{p\in DP}\sum_{(i,j)\in A_p} c_{ij}^k x_{ij}^{krp}\right) + \sum_{\#\in\{I,II,III\}}\sum_{i\in O_\#}\left[\beta_\# \bar{d}_i + \eta_\#\left(1 - \sum_{k\in K}\sum_{r\in R}\sum_{p\in DP} y_i^{krp}\right)\right] \tag{22}$$

$$s.t. \quad (2–9), (16–19) \tag{23}$$

$$\sum_{j:(0,j)\in A_p} z_{0j}^p = |B_p| \sum_{k\in K}\sum_{r\in R} v^{krp} \quad p \in DP \tag{24}$$

$$\sum_{j:(j,i)\in A_p} z_{ji}^p - \sum_{j:(i,j)\in A_p} z_{ij}^p = \sum_{k\in K}\sum_{r\in R} y_i^{krp} \quad p \in DP,\ i \in B_p \tag{25}$$

$$z_{ij}^p \leq |B_p| \sum_{k\in K}\sum_{r\in R} x_{ij}^{krp} \qquad p \in DP,\ (i,j) \in A_p \tag{26}$$

$$s_i \geq p - |P|\left(1 - \sum_{k\in K}\sum_{r\in R} y_i^{krp}\right) \qquad p \in DP,\ i \in B_p \tag{27}$$

$$s_i \leq p + |P|\left(1 - \sum_{k\in K}\sum_{r\in R} y_i^{krp}\right) \qquad p \in DP,\ i \in B_p \tag{28}$$

$$\bar{d}_i \geq s_i - (\delta_i + \theta_i) \qquad i \in O \tag{29}$$

$$\bar{d}_i \geq (\pi_i - \delta_i - \theta_i) \sum_{k\in K}\sum_{r\in R}\sum_{p\in DP:p>\delta_i} y_i^{krp} \quad i \in O : \pi_i \geq \delta_i + \theta_i \tag{30}$$

$$z_{ij}^p \geq 0 \quad p \in DP,\ (i,j) \in A_p \tag{31}$$

$$s_i, \bar{d}_i \geq 0 \quad i \in O \tag{32}$$

The SPDM objective function (22) only differs from the one of the EPDM in the third term due to the distinct metrics used, leading to the objective function values of these models not being comparable.

The constraint system (24–26) is used to prevent subtours. Constraints (24) equals the units of flow leaving the depot in a departure period to an upper bound on the number of orders satisfied in the routes starting at that departure period. Constraints (25) are the flow conservation constraints, while constraints (26) guarantee that flow can only traverse an arc that was used and such flow is at most $|B_p|$. This value represents the number of orders released before period p and is an upper bound for the number of orders satisfied in that departure period. Constraints (27) and (28) identify the departure period at which the route where the orders are served starts and are only not redundant when the orders are in fact served. Constraints (29) define the service delay in terms of departure periods, while (30) relate the service delay with the first departure period that may be used whenever it is after the release date plus the order' allowed delay and are similar to constraints (15) of the EPDM. Finally, the domains of the new variables are set through (31) and (32).

5 | Matheuristic

As will be seen in Section 6, the EPDM and the SPDM are not suited to address the largest test instances. Therefore, we developed a matheuristic based on a rolling-horizon approach with the SPDM, called RH matheuristic, to decompose the problem and efficiently obtain feasible solutions.

Rolling-horizon-type procedures have been successfully applied in complex real-world problems. Recently, Anjos, Gendron, and Joyce-Moniz [33] proposed a rolling-horizon heuristic to solve realistically-sized instances of a multi-period optimization problem for the siting of electric vehicle charging stations. The authors address the problem by solving a single period per iteration and fixing the generated solution values for the following periods. Nevertheless, the interrelation among periods inherent to our problem brings additional complexity, even when applying this type of heuristic.

As in [33], the main idea of the RH matheuristic is to decompose the problem into smaller subproblems by only considering some departure periods. Thus, we define $\overline{p}$ as the number of departure periods included in each subproblem.

Preliminary computational experiments showed that the SPDM is the most efficient and effective model, as validated by the results in Section 6.2. Therefore, the RH matheuristic will use the SPDM to solve the subproblems. However, solving the subproblems for the largest instances, even with $\overline{p} = 1$, continues to be very time-consuming. Therefore, we consider only one route type, which will be explained next how it is computed, and the vehicles available in the departure periods that are part of the subproblem.

To ensure that the SPDM efficiently solves the subproblem, we consider $\overline{p} = 1$, limit the number of orders added to the subproblem and set a tolerance and a time limit. More precisely, the maximum number of orders (including the depot) in the subproblem is 20, and we start by adding the orders of clients from Segment III, then from Segment I, and finally, from Segment II. The orders are added to the subproblem according to a lexicographic ordering. Additionally, we set a tolerance of five percent and a time limit of 600 s. The presented values are the ones that provided the best results, on average, among the ones tested.

As mentioned previously, the RH matheuristic iteratively solves subproblems with subsets of departure periods with the settings mentioned in the previous paragraphs until all periods are considered. The first subproblem includes the first $\overline{p}$ departure periods, the second subproblem includes the subsequent $\overline{p}$ departure periods, and this process continues until all periods have been assigned to a subproblem. Since the sets of departure periods for each subproblem are constructed sequentially, they are disjoint. For each set of departure periods, the route type is fixed and is computed according to the set of periods in the subproblem and an estimation of the route duration needed to visit all unserved orders (Algorithm 1: line 7). More precisely, if the set of periods contains the first departure period and there are unserved orders from Segment III, the route type is fixed to the one with the smallest duration that allows to fulfill all orders from Segment III. Otherwise, the nearest neighbor algorithm is applied to find a route that contains all the unserved orders. The duration of this route, including the service times, is an upper bound on the necessary time to serve all these orders. Thus, the route type is fixed to the one with the duration closest to this upper bound that ends before the planning horizon. The resolution of a subproblem produces a set of routes $\mathcal{R}$ and a set of unserved orders $\overline{\mathcal{O}}$ (Algorithm 1: line 8). Afterwards, the routes with an occupation rate lower than a given percentage μ are discarded. More precisely, a route $l \in \mathcal{R}$ with a duration, in time units, less than $\mu \alpha r_l$, where r_l is the route type assign to l, is discarded (Algorithm 1: lines 9–13). In that case, the orders fulfilled in route l are added to the set $\overline{\mathcal{O}}$, and the vehicle assigned to route l remains available. By discarding these routes, the vehicle is available for the next departure period and, hopefully, can fulfill more orders. Next, if there are unserved orders, the RH matheuristic attempts to insert them in routes constructed so far through the procedure *InsertOrders* (Algorithm 1: lines 16–19).

After all departure periods have been included in the subproblems, an adaptation of the *NearestNeighbor* heuristic is executed considering the set of unserved orders for the departure periods with available vehicles (Algorithm 1: line 21). The procedure *NearestNeighbor* constructs a route with a route type fixed using the procedure explained previously. Thus, orders are added to the route until there are no more feasible orders or the route size is exceeded. Next, a *SwitchOrders* procedure is applied (Algorithm 1: line 22). This procedure tries all possible switches of two orders in a given route and performs the one minimizing the EPD metric. To conclude, the *InsertOrders* procedure is executed again (Algorithm 1: lines 23–25).

Algorithm 1 details the pseudocode of the RH matheuristic.

The preliminary computational experiment showed no value of the parameter μ that dominates all others. Therefore, we decided to execute the RH matheuristic for the three values of μ that provided the best results on average among the ones tested, namely $\mu \in \{0.7, 0.8, 0.9\}$. The solution obtained by the RH

ALGORITHM 1 | Pseudocode of the RH matheuristic.

Require: MTVRP-RDIP instance; $\bar{p}$; μ

$\mathcal{F} \leftarrow \emptyset$ ▷ Feasible solution
$\overline{\mathcal{O}} \leftarrow \emptyset$ ▷ Set of unserved orders
$\mathcal{P} \leftarrow \emptyset$ ▷ Set of departure periods included in the subproblem
$\mathcal{N} \leftarrow K$ ▷ Set of available vehicles
while There are departure periods never included in $\mathcal{P}$ **do**
  Include in $\mathcal{P}$ the next $\bar{p}$ departure periods that were not included yet in any subproblem.
  Determine route type r_l to be fixed in the SPDM.
  Solve SPDM for $\mathcal{P}$, $\overline{\mathcal{O}}$, and $\mathcal{N}$ and obtain the set of routes $\mathcal{R}$. ▷ SPDM solved with described setting
  for $l \in \mathcal{R}$ **do**
    if Duration route $l < \mu\alpha r_l$ **then**
      Discard route l: remove it from $\mathcal{R}$.
    end if
  end for
  Update $\overline{\mathcal{O}}$ and $\mathcal{N}$ according to $\mathcal{R}$.
  Insert in $\mathcal{F}$ the routes from $\mathcal{R}$.
  if $\overline{\mathcal{O}} \neq \emptyset$ **then**
    Apply procedure *InsertOrders*($\mathcal{F}, \overline{\mathcal{O}}$).
    Update $\mathcal{F}$ and $\overline{\mathcal{O}}$.
  end if
end while
Apply *NearestNeighbor*($\overline{\mathcal{O}}, \mathcal{N}$) and update $\overline{\mathcal{O}}$, $\mathcal{N}$, and $\mathcal{F}$.
Apply procedure *SwitchOrders*($\mathcal{F}$).
if $\overline{\mathcal{O}} \neq \emptyset$ **then**
  Apply procedure *InsertOrders*($\mathcal{F}, \overline{\mathcal{O}}$).
end if

Ensure: A feasible solution $\mathcal{F}$.

matheuristic is the best of the three runs, and the time taken to obtain such solution is the sum of the computational time of the three runs.

6 | Computational Experiment

The objective of the computational experiment is threefold: (i) first, both models, EPDM and SPDM, are compared regarding the efficiency and quality of the generated solutions. The number of instances solved to optimality, the gap values obtained by the solver and the execution times are used to measure efficiency, while the quality of the solutions is measured in terms of the KPIs defined next since their objective functions are different and both have penalizing terms; (ii) the best-performing model is used to obtain solutions for all test instances, which are also evaluated according to the KPIs; (iii) third, the performance of the RH matheuristic is compared with the best-performing model.

The KPIs that were considered relevant for the auto supply components distributing company were the following:

- number of used vehicles;
- number of unserved orders;
- number of orders served with delay (exact period or start period); and
- duration of the delays (exact period or start period).

Thus, these KPIs will be used to compare the solutions obtained by the proposed solution methods. The importance of analyzing the solutions through KPIs is related to the fact that the objective functions of both the EPDM and the SPDM have penalized terms, and their values depend on the penalizations defined by the stakeholders.

The mathematical models were implemented in C++ and solved using the Concert Technology from CPLEX 20.1 [34]. The RH matheuristic was also implemented in C++. All computational experiments were carried out using a computer with an AMD Ryzen Threadripper 3960X (24 - core) with 64 GB, and run in a multi-thread mode.

6.1 | Generated Instances

The company operates in six geographical areas, each with a number of clients ranging from 14 to 24. The clients may place several orders that are released during the time horizon. Consequently, the number of orders to serve per area varies between 18 and 36. Therefore, we generated several test instances by randomly merging geographical areas to mimic the company's operation. Two main sets of instances are thus considered: small and large. Small instances represent one or two areas, while the large ones contain five or six. The company aims to optimize its operation, generally between 130 and 140 orders.

To generate the other instances' characteristics besides the number of clients and orders, we assumed that each area has one vehicle and that all vehicles are identical. Additionally, we consider three route types with a duration of two, three, and four periods. Finally, the number of periods is set to 20, and the number of departure periods is five: periods 2, 5, 11, 15, and 17. Instances with these characteristics comprise the base instances' set.

To increase the number of instances, we considered the instances from the base set and varied the values of the following parameters: number of vehicles, route types, and departure periods. More precisely, for the instances with two areas, we generated similar instances with three vehicles (the base set has two). The number of different route types, although set equal to three, may be defined with different durations. Thus, two additional settings for the route types were added: routes of types two, four, and six and two, five, and seven. Finally, one additional setting with seven departure periods was considered instead of the five from the base set. The seven departure periods are: 2, 5, 9, 11, 13, 15, and 19. This is summarized in Table 4.

Table 4 contains the number of areas merged (#*areas*), the number of instances of the base set (#*inst*), the interval of the number of released orders (#*orders*) for the instances with the same number of merged areas, and the number of different settings for the number of vehicles (vehicles), route types (routes), and departure periods (d_periods), and the total number of instances generated (#*inst_total*). Thus, Table 4 depicts the

TABLE 4 | Summary of the characteristics of the instances set.

			# of different settings			
#areas	*#inst*	*#orders*	Vehicles	Routes	d_periods	*#inst_total*
1	7	18 to 36	1	3	2	42
2	5	47 to 65	2	3	2	60
5	6	129 to 144	1	3	2	36
6	2	162 to 182	1	3	2	12

main characteristics of the 150 test instances, which, jointly with the complete computational results, are available at https://hal.iseg.ulisboa.pt/˜rbernardino/mtvrp-rdip/.

6.2 | Comparison Between EPDM and SPDM

This section compares the two models according to the number of instances solved to optimality, gap values, execution times, and the defined KPIs.

Table 5 summarizes the results obtained within the set time limit of 3600 s for both models. To establish which is the most appropriate model to address the MTVRP-RDIP, we used a subset of instances, namely the instances with a maximum of 36 orders, as can be seen in the first column of Table 5. The results are grouped by the number of orders ($|O|$). For each group and each model, Table 5 depicts the number of instances solved to optimality ($\#solved$), the average percentage of gap provided by the solver at the end of the time limit ($\overline{gap}$), the average execution time, in seconds, to obtain the optimal value ($\overline{tcpu}$), and the average number of subproblems solved (or branch-and-bound nodes explored) ($\overline{\#B\&B}$).

Table 5 clearly shows the better performance of SPDM over EPDM. SPDM was able to solve 25 (column seven) out of the 42 (column two) instances, while EPDM solves five (column three). All remaining indicators corroborate this. In fact, the average gap provided by the SPDM is 8.20 whilst it is 26.32 for the EPDM. The time limit of 3600 s is reached by EPDM for all test instances, except for the instances with 18 orders where the average computational time is 2603.00, while SPDM uses only between 99.42 and 3374.83 s. The average number of subproblems solved is also much higher in the EPDM. More precisely, the average number of subproblems solved by the EPDM is 803 010.43, whereas the SPDM solved an average of 53 248.17 subproblems.

The average values of the KPIs for both models are depicted in Table 6. Thus, for each group of instances and each model, columns two and three include the average number of orders not served ($\overline{nserved}$), columns four and five give the average number of orders that were served with a positive exact period delay ($\overline{\#epdelay}$), columns six and seven the average exact time delay ($\overline{epdelay}$), for EPDM and SPDM. Note that the EPD may be computed for any feasible solution of the SPDM.

The average number of unserved orders is never greater for the SPDM than for the EPDM. Additionally, the average number of orders served with delay, and the average exact period delay, although greater for the SPDM model, are very similar in both models. We recall that the EPD metric is not considered in the objective function of the SPDM. Even so, these results show that the SPDM provides a good approximation for the EPD metric.

To conclude this section, it is clear that the SPDM is the better model and, thus, the one to be used henceforward to address the remaining instances.

6.3 | SPDM Results

As mentioned in the previous section, the SPDM will be used, with a time limit of 3600 s, to obtain the optimal values of the test instances. Table 7 summarizes the results obtained within the set time limit and is organized similarly to Table 5.

From Table 7, we can see that the SPDM could only obtain the optimal value of instances with a maximum of 36 orders. The optimal value was found for all instances with 18 orders, three with 29 orders, eight with 32 orders, one with 33 orders, and one with 36 orders. Additionally, and as expected, the average gap tends to increase with the number of orders, reaching more than 50% for the instances with 61 and 65 orders, which is a substantial average gap. Instances with 129 orders or more have a dash symbol in the column gap because in the feasible solution obtained within the time limit, no orders are served, and, thus, the computed gap values have no meaning.

The average computational time of the solved instances is 658.76 s, which is a reasonable value. In fact, the average computational time to obtain the optimal value of instances with 18 orders, which are the smallest ones, was 99.42 s.

It is interesting to notice that, for the instances with more than 100 orders, the number of subproblems solved is small. Actually, no subproblems are solved for these instances in the time limit of 3600 s.

From the presented results, the SPDM could address the instances with a maximum of 36 orders since it could find the optimal values for most of the instances (59.52%). The average gap values were substantial for the remaining instances. Next, we will analyze the quality of the feasible solutions obtained regarding the KPIs defined previously.

Table 8 shows a summary of the KPIs of the solutions provided by the SPDM. However, it should be noted that the feasible

TABLE 5 | Summary of the results for EPDM and SPDM.

		EPDM				SPDM			
$\|O\|$	$\#inst$	$\#solved$	$\overline{gap}$(%)	$\overline{tcpu}$ (s)	$\overline{\#B\&B}$	#solved	$\overline{gap}$ (%)	$\overline{tcpu}$ (s)	$\overline{\#B\&B}$
18	12	5	6.00	2603.00	463 807.83	12	0.01	99.42	5573.83
29	6	0	40.22	3600.00	1 561 959.17	3	7.97	2186.33	105 170.67
32	12	0	29.22	3600.00	1 191 512.42	8	8.22	2048.17	50 773.92
33	6	0	37.15	3600.00	366 733.67	1	10.06	3090.83	64 966.83
36	6	0	36.45	3600.00	381 739.67	1	22.92	3374.83	89 904.17
Summary		5	26.32	3316.31	803 010.43	25	8.20	1849.60	53 248.17

TABLE 6 | Summary of the KPIs for EPDM and SPDM solutions.

	$\overline{nserved}$		$\overline{\#epdelay}$		$\overline{epdelay}$	
$\|O\|$	EPDM	SPDM	EPDM	SPDM	EPDM	SPDM
18	2.67	2.67	6.92	7.67	1.34	1.47
29	4.00	4.00	18.17	19.33	2.99	2.83
32	6.92	6.50	19.08	20.67	2.11	2.25
33	5.50	4.67	20.17	22.50	2.59	2.79
36	11.67	10.83	15.17	18.00	2.34	2.61
Average	5.76	5.40	15.07	16.64	2.12	2.24

TABLE 7 | Summary of the SPDM results.

$\|O\|$	$\#inst$	$\#solved$	$\overline{gap}$ (%)	$\overline{tcpu}$ (s)	$\overline{\#B\&B}$
18	12	12	0.01	99.42	5573.83
29	6	3	7.97	2186.33	105 170.67
32	12	8	8.22	2048.17	50 773.92
33	6	1	10.06	3090.83	64 966.83
36	6	1	22.92	3374.83	89 904.17
47	12	0	46.63	3600.00	7930.92
50	12	0	14.98	3600.00	9768.83
52	12	0	45.98	3600.00	3614.67
61	12	0	66.54	3600.00	1888.00
65	12	0	73.08	3600.00	2792.58
129	6	0	—	3600.00	0.00
130	12	0	—	3600.00	0.00
133	6	0	—	3600.00	0.00
144	12	0	—	3600.00	0.00
162	6	0	—	3600.00	0.00
182	6	0	—	3600.00	0.00
Summary		25	—	3110.67	16 989.09

solution obtained for some instances corresponds to the empty solution where no orders are served. Thus, for these solutions, we cannot compute most of the KPIs. Hence, these instances will be omitted from the summary. For the instances with the same number of orders ($|O|$), Table 8 contains the number of instances considered ($\#inst$), the average number of vehicles used in the solution ($\overline{vehicles}$), the average number of orders not served ($\overline{nserved}$), the average number of orders served with start period delay ($\overline{\#spdelay}$), the average start period delay ($\overline{spdelay}$), the average number of orders served with a positive exact period delay ($\overline{\#epdelay}$), and the average exact period delay ($\overline{epdelay}$).

TABLE 8 | Summary of the KPIs for SPDM solutions for instances with a non-empty solution.

$\lvert O \rvert$	$\#inst$	$\overline{vehicles}$	$\overline{nserved}$	$\overline{\#spdelay}$	$\overline{spdelay}$	$\overline{\#epdelay}$	$\overline{epdelay}$
18	12	1.00	2.67	5.83	0.91	7.67	1.47
29	6	1.00	4.00	12.67	1.66	19.33	2.83
32	12	1.00	6.50	11.58	1.13	20.67	2.25
33	6	1.00	4.67	15.50	1.66	22.50	2.79
36	6	1.00	10.83	12.67	1.55	18.00	2.61
47	12	2.50	10.50	15.58	1.08	25.08	2.09
50	12	2.50	8.25	14.50	0.91	27.33	1.72
52	12	2.50	9.00	23.42	1.43	33.92	2.66
61	12	2.50	21.42	18.00	1.43	27.58	2.44
65	12	2.50	26.75	18.92	1.41	27.67	2.25
129	4	4.50	93.75	17.75	1.84	22.50	2.56
130	5	4.20	104.40	15.60	2.44	18.00	2.85
133	3	4.00	101.67	20.00	2.66	25.00	3.59
144	5	3.60	114.80	16.60	1.50	25.00	1.95
162	1	3.00	136.00	17.00	1.96	20.00	3.00
182	1	4.00	154.00	15.00	1.36	21.00	2.04
Average		2.21	26.48	15.40	1.38	23.28	2.30

Regarding the average number of vehicles used, we can see that instances up to 129 orders use all existing vehicles. The instances with a number of orders between 129 and 144 use almost all existing vehicles since the average number of existing vehicles is 5, and the average number of vehicles used is at least 3.60. Focusing on the instances with 162 and 182 orders, even though there are six vehicles, the solutions obtained only use an average of three or four, which is suboptimal since there are several unserved orders and unused vehicles, and the fixed cost of using a vehicle is inferior to the penalization of not serving an order.

The average number of unserved orders increases with the instance dimension, which is expected since the number of orders also increases. Nevertheless, the ratio between the average number of unserved orders and the number of orders also increases. More precisely, instances with 65 orders have a ratio of unserved orders of approximately 41.15% while instances with 182 orders have a ratio of 84.62%, which is more than the double.

A significant amount of orders were served with a positive start period delay. In fact, for the instances with 50 orders from the average of 41.75 (50 − 8.25) orders served, an average of 14.50 have start period delays, corresponding to approximately 35% of the orders. There seems to exist no relation between the number of orders with start period delay and the number of orders from the instance. The average start period delay is at most 2.66.

Finally, regarding the average exact period delay, as expected due to the definition of metrics EPD and SPD, it is always larger than the average start period delay. The average number of orders served with exact period delay corresponds to more than half of the average number of orders served, and the average exact period delay is at most 3.59 periods. To conclude, the average number of orders served with start period delay is 15.40 while the one with exact period delay is 23.28, corresponding to an increase of 51%. Nevertheless, the average start period delay is 1.38, whereas the average exact period delay is 2.30, which corresponds to one extra period on average. The difference in the average delays is very reasonable, considering the different definition of the metrics. However, it is important to note that the exact period delay KPI is not considered in the SPDM's objective function.

The feasible solutions obtained by the SPDM showed poor performance regarding the defined KPIs. This poor quality is mainly due to the size and complexity of the instances, which motivated the RH matheuristic presented in Section 5.

6.4 | RH Matheuristic Results

As mentioned previously, no value of μ (occupation rate parameter to discard routes) provided significantly better results in the preliminary computational experiment. Therefore, the RH matheuristic will be run considering three values of μ, namely 0.70, 0.80, and 0.90, and the final solution obtained is the best of the three runs.

Table 9 shows a summary of the RH matheuristic results, including the KPIs of the solutions provided for the instances where the SPDM found a feasible solution different from the empty one (the ones in Table 8). The reported results correspond to the best solutions found in the three runs, except for the computational time, where its sum is presented. For the instances with the same number of orders ($\lvert O \rvert$) the table contains the average percentage of gap ($\overline{gap}$) between the solution value obtained by the RH matheuristic and the upper bound provided by the SPDM ($gap = 100\times$(RH matheuristic solution value - SPDM solution value)/SPDM solution value), the average sum of the

TABLE 9 | Summary of the RH matheuristic solutions' KPIs for instances with a SPDM non-empty solution.

$\|O\|$	$\overline{gap}$ (%)	$\overline{tcpu}$ (s)	$\overline{vehicles}$	$\overline{nserved}$	$\overline{\#spdelay}$	$\overline{spdelay}$	$\overline{\#epdelay}$	$\overline{epdelay}$
18	14.99	0	1	3.00	7.50	1.49	8.92	2.23
29	20.11	0	1	5.67	12.67	1.58	19.50	2.78
32	19.36	0	1	8.25	13.00	1.56	20.58	2.78
33	24.22	0	1	6.50	16.67	2.07	23.50	3.34
36	12.14	2.00	1	12.33	13.83	1.84	19.33	3.31
47	−16.03	52.42	2.5	8.42	23.00	1.93	30.75	3.30
50	10.06	4.00	2.5	10.17	19.83	1.80	30.33	3.03
52	−26.49	40.92	2.5	6.92	27.83	1.80	36.17	3.09
61	−41.80	66.83	2.5	10.92	28.58	1.66	40.50	3.10
65	−49.38	86.92	2.5	11.75	34.33	2.00	43.33	3.33
129	−62.25	312.00	5	33.25	58.25	1.76	77.25	3.14
130	−66.48	367.00	5	31.40	61.60	1.88	76.20	2.97
133	−66.31	276.33	5	32.67	59.67	1.82	76.67	3.03
144	−63.77	253.80	5	35.20	69.60	1.92	90.00	3.26
162	−68.53	419.00	6	35.00	77.00	1.74	99.00	3.11
182	−55.76	190.00	6	73.00	64.00	2.23	80.00	3.57
Average	−16.17	72.88	2.39	12.66	27.41	1.78	36.78	3.03

computational time in the three runs ($\overline{tcpu}$), in seconds, to obtain the solutions and the KPIs previously defined.

A negative average gap points to a better performance of the RH matheuristic over the SPDM. Thus, Table 9 seems to indicate that the RH matheuristic provides better solutions, on average, than the SPDM (−16.17%). In detail, we can see a positive average gap for instances up to 36 orders and instances with 50 orders. These results are unsurprising since the SPDM obtained the optimal value of most instances up to 36 orders. For the instances with 52 orders or more, the RH matheuristics always provided, on average, a solution with a lower objective function value. Regarding the computational time, the RH matheuristic took an average of 72.88 seconds to provide the feasible solutions for the presented instances, representing a value much lower than that registered by the SPDM, where, in general, the time limit was reached and the average reported computational time is 3110.67 s. Additionally, we recall that 72.88 s correspond to the sum of three district runs of the RH matheuristic.

The detailed results show that the number of vehicles used corresponds to the number of existing vehicles for all test instances. This differs from the solutions obtained by the SPDM since in this case only some of the existing vehicles are used. Nevertheless, this is unsurprising as the fixed cost of using a vehicle is less than the penalization of not serving an order, and the RH matheuristic was tailored to serve the largest number of orders possible.

The average number of unserved orders generally increases with the number of orders of the instance, similar to what happened in the SPDM solutions. When comparing the average number of unserved orders of the solutions of SPDM to the ones of the RH matheuristic, we can see that the latter is significantly smaller, with an average number of 12.66 unserved orders compared to 26.48. This difference is even greater for the biggest instance, with 182 orders, where the average number of unserved orders in the SPDM (154) is more than the double that of the RH matheuristic (73).

Regarding the number of orders with start period delay, the average of the solutions obtained with the RH matheuristic is 27.41, and the one obtained with SPDM is 15.40, which is significantly smaller. Since we are using a tolerance of 5% as a stopping criterion in the resolution of the subproblems of the RH matheuristic, it is not surprising that the solutions present start period delays even though the objective function includes their minimization. The average start period delay is 1.78, approximately half a period more than the one obtained with the SPDM model (1.38). Although the average number of orders with delay in the RH matheuristic is more significant than in the solutions of the SPDM, the delay is not as significant.

Finally, the average number of orders with exact period delay is 36.78 for RH matheuristic solutions, and it is 23.28 for SPDM solutions, while the average delay is, respectively, 3.03 and 2.30. Despite the SPDM dominating the RH matheuristic in terms of the KPIs related to the delays, it is essential to also note that the number of orders served in the solutions of the RH matheuristic is larger; consequently, more orders may have delays.

Table 10 shows a summary of the computational time and KPIs of the solutions obtained by the RH matheuristic for the instances where the SPDM provided the empty solution and it has a similar layout as Table 9.

The RH matheuristic was able to obtain solutions for the instances for which the SPDM could only obtain the empty solution in an average of 351.38 s. It is interesting to notice that

TABLE 10 | Summary of the RH matheuristic solutions' KPIs for instances with a SPDM empty solution.

$\|O\|$	$\overline{tcpu}$ (s)	$\overline{vehicles}$	$\overline{nserved}$	$\overline{\#spdelay}$	$\overline{spdelay}$	$\overline{\#epdelay}$	$\overline{epdelay}$
129	415.00	5.00	27.50	60.50	1.69	81.00	3.23
130	274.29	5.00	27.57	66.29	1.98	80.29	3.38
133	327.00	5.00	25.00	62.33	1.82	84.67	3.19
144	305.43	5.00	36.14	64.57	1.93	84.29	3.22
162	439.40	6.00	38.60	73.60	1.88	98.20	3.19
182	424.80	6.00	50.40	74.60	2.05	97.40	3.23
Average	351.38	5.34	35.21	37.76	1.93	87.79	3.25

instances with 129 orders are among the most time-consuming instances, even though they are not the biggest instances.

Similar conclusions can be drawn from the KPIs in Table 10. More precisely, the number of vehicles used is always equal to the number of existing vehicles for all test instances. The number of unserved orders increases with the instance dimension, except for instances with 133 orders. Finally, the average number of orders with penalty delays is significant.

To conclude, to make a fair assessment of the performance of the RH matheuristic, we make a thorough comparison of its results to the ones obtained with the SPDM for the instances with 18 orders since this is the only number of orders where all instances were solved to optimality. The average gap between the solution value provided by the RH matheuristic and the optimal value is 14.99%, which is reasonable considering that the objective function includes high value penalties. Focusing on the defined KPIs, the average number of unserved orders in the SPDM's solutions is 2.67. In contrast, it is 3.00 for the RH matheuristic solutions, corresponding to an increase of approximately 12%, which is not significant. The difference between the KPIs related to delays, especially for the start period delay, is more significant than the one previously presented. More precisely, the difference in the average number of orders served with start period delay and exact period delay is 1.67 (5.83 vs. 7.50) and 1.25 (7.67 vs. 8.92). These values correspond to an increase of 29% and 16% for the start and the exact period delay, respectively. The difference in the average delay is 0.58 (0.91 vs. 1.49) for the start period delay and 0.76 (1.47 compared to 2.23) for the exact period delay. Finally, the RH matheuristic took an average of one second to obtain these solutions, while the SPDM took 99.42 s on average to obtain the optimal solutions. Thus, the RH matheuristic can provide efficient solutions similar to the optimal ones, especially regarding the number of unserved orders.

To summarize, the RH matheuristic can provide feasible solutions for all test instances in an efficient manner, which does not happen with the SPDM. For the instances where both methods found non-empty feasible solutions, the ones obtained by the RH matheuristic serve, on average, more orders, while the ones obtained by the SPDM have, on average, smaller delays. Even for the instances with a known optimal value, the solutions obtained with the RH matheuristic are of reasonable quality. Therefore, using the RH matheuristic is advisable for instances with 42 orders or more or when efficiency is a priority.

7 | Conclusion

This article studies the MTVRP-RDIP, a variant of the MTVRP that was motivated by a practical application of an auto supply components distributing company where the planning horizon is divided into periods, and the clients' orders are made available during the planning horizon, imposing release dates. Another peculiarity of the MTVRP-RDIP is that the routes can only start in a subset of periods called departure periods and may last several periods, thus creating the interrelated periods.

We proposed two mathematical formulations for the MTVRP-RDIP: the EPDM and the SPDM. The former measures the delay of a served order as the difference between the period in which it was served and the latest period it could be served without delay. The latter considers the difference between the period in which the route that serves the client leaves the depot and the latest period it could be served without delay.

The computational results with instances based on real data show that the SPDM outperforms the EPDM. The EPDM could only obtain the optimal value of five of the smaller test instance within the set time limit. This was the primary motive for developing the SPDM, where the notion of delay is relaxed. Nevertheless, the solutions obtained by the SPDM have, on average, similar exact period delays than those provided by the EPDM.

Even though the SPDM could find optimal values of several instances, it could not efficiently address the largest ones. More precisely, the SPDM could only provide optimal solutions for all of the smallest instances with a maximum of 18 orders. For most of the largest instances, the feasible solution obtained within the time limit corresponds to the one where no order is served.

The RH matheuristic was developed to address the largest instances. This matheuristic decomposes the problem into subproblems by successively considering a subset of departure periods. The RH matheuristic also incorporates some procedures based on local movements to fulfill more orders. The RH matheuristic is very efficient, providing feasible solutions for the instances with 182 orders in an average of 385.67 seconds. When comparing the solutions obtained by the RH matheuristic to the ones obtained with the SPDM, it may be seen that despite fulfilling more orders, more orders are fulfilled with delay. This is not surprising because both the penalizations used in the SPDM and the local procedures were developed to favor the fulfillment of orders instead of minimizing the delays, which reflects the company's desire.

From this work, we may conclude that the MTVRP-RDIP is a challenging problem to be addressed by the proposed mathematical models, even within the setting of a matheuristic. Therefore, developing a metaheuristic as a solution method for this problem would be interesting in the future.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. G. Dantzig and J. Ramser, "The Truck Dispatching Problem," *Management Science* 6 (1959): 80–91.

2. K. Braekers, K. Ramaekers, and I. Van Nieuwenhuyse, "The Vehicle Routing Problem: State of the Art Classification and Review," *Computers & Industrial Engineering* 99 (2016): 300–313.

3. C. Koç, T. Bektaş, O. Jabali, and G. Laporte, "Thirty Years of Heterogeneous Vehicle Routing," *European Journal of Operational Research* 249, no. 1 (2016): 1–21.

4. R. Lahyani, M. Khemakhem, and F. Semet, "Rich Vehicle Routing Problems: From a Taxonomy to a Definition," *European Journal of Operational Research* 241, no. 1 (2015): 1–14.

5. G. Laporte, "Fifty Years of Vehicle Routing," *Transportation Science* 43, no. 4 (2009): 408–416.

6. A. Mor and M. G. Speranza, "Vehicle Routing Problems Over Time: A Survey," *4OR* 18 (2020): 129–149.

7. P. Toth and D. Vigo, *Vehicle Routing: Problems, Methods, and Applications* (Philadelphia, PA: SIAM, 2014).

8. H. Zhang, H. Ge, J. Yang, and Y. Tong, "Review of Vehicle Routing Problems: Models, Classification and Solving Algorithms," *Archives of Computational Methods in Engineering* 29 (2022): 195–221.

9. D. Cattaruzza, N. Absi, and D. Feillet, "Vehicle Routing Problems With Multiple Trips," *4OR* 14 (2016): 223–259.

10. C. Archetti, D. Feillet, A. Mor, and M. Speranza, "Dynamic Traveling Salesman Problem With Stochastic Release Dates," *European Journal of Operational Research* 280, no. 3 (2020): 832–844.

11. M. A. Klapp, A. L. Erera, and A. Toriello, "Request Acceptance in Same-Day Delivery," *Transportation Research Part E: Logistics and Transportation Review* 143 (2020): 102083.

12. M. W. Ulmer, B. W. Thomas, A. M. Campbell, and N. Woyak, "The Restaurant Meal Delivery Problem: Dynamic Pickup and Delivery With Deadlines and Random Ready Times," *Transportation Science* 55, no. 1 (2021): 75–100.

13. W. J. A. van Heeswijk, M. R. K. Mes, and J. M. J. Schutten, "The Delivery Dispatching Problem With Time Windows for Urban Consolidation Centers," *Transportation Science* 53, no. 1 (2019): 203–221.

14. S. A. Voccia, A. M. Campbell, and B. W. Thomas, "The Same-Day Delivery Problem for Online Purchases," *Transportation Science* 53, no. 1 (2019): 167–184.

15. D. Cattaruzza, N. Absi, and D. Feillet, "The Multi-Trip Vehicle Routing Problem With Time Windows and Release Dates," *Transportation Science* 50, no. 2 (2016): 676–693.

16. C. Archetti, D. Feillet, and M. G. Speranza, "Complexity of Routing Problems With Release Dates," *European Journal of Operational Research* 247, no. 3 (2015): 797–803.

17. D. Reyes, A. L. Erera, and M. W. Savelsbergh, "Complexity of Routing Problems With Release Dates and Deadlines," *European Journal of Operational Research* 266, no. 1 (2018): 29–34.

18. B. Shelbourne, "The Vehicle Routing Problem With Release and due Dates: Formulations, Heuristics and Lower Bounds" (PhD thesis, University of Southampton, 2016).

19. B. C. Shelbourne, M. Battarra, and C. N. Potts, "The Vehicle Routing Problem With Release and due Dates," *INFORMS Journal on Computing* 29, no. 4 (2017): 705–723.

20. L. Liu, K. Li, and Z. Liu, "A Capacitated Vehicle Routing Problem With Order Available Time in e-Commerce Industry," *Engineering Optimization* 49, no. 3 (2017): 449–465.

21. W. Li, Y. Wu, P. N. Ram Kumar, and K. Li, "Multi-Trip Vehicle Routing Problem With Order Release Time," *Engineering Optimization* 52, no. 8 (2020): 1279–1294.

22. S. Moons, K. Ramaekers, A. Caris, and Y. Arda, "Integration of Order Picking and Vehicle Routing in a B2C e-Commerce Context," *Flexible Services and Manufacturing Journal* 30 (2018): 813–843.

23. L. Zhen, C. Ma, K. Wang, L. Xiao, and W. Zhang, "Multi-Depot Multi-Trip Vehicle Routing Problem With Time Windows and Release Dates," *Transportation Research Part E: Logistics and Transportation Review* 135 (2020): 101866.

24. W. Li, K. Li, P. N. R. Kumar, and Q. Tian, "Simultaneous Product and Service Delivery Vehicle Routing Problem With Time Windows and Order Release Dates," *Applied Mathematical Modelling* 89 (2021): 669–687.

25. X. Sun, K. Li, and W. Li, "The Vehicle Routing Problem With Release Dates and Flexible Time Windows," *Engineering Optimization* 54, no. 12 (2022): 2123–2139.

26. C. Archetti, O. Jabali, and M. G. Speranza, "Multi-Period Vehicle Routing Problem With due Dates," *Computers & Operations Research* 61 (2015): 122–134.

27. H. Larrain, L. C. Coelho, C. Archetti, and M. G. Speranza, "Exact Solution Methods for the Multi-Period Vehicle Routing Problem With due Dates," *Computers & Operations Research* 110 (2019): 148–158.

28. J. T. Soman and R. Patil, "A Reactive Grasp Heuristic Algorithm for Vehicle Routing Problem With Release Date and due Date Incurring Inventory Holding Cost and Tardiness Cost," in *2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (New York: IEEE, 2019), 1393–1397, https://doi.org/10.1109/IEEM44572.2019.8978851.

29. J. T. Soman and R. J. Patil, "A Scatter Search Method for Heterogeneous Fleet Vehicle Routing Problem With Release Dates Under Lateness Dependent Tardiness Costs," *Expert Systems with Applications* 150 (2020): 113302.

30. P. Campelo, F. Neves-Moreira, P. Amorim, and B. Almada-Lobo, "Consistent Vehicle Routing Problem With Service Level Agreements: A Case Study in the Pharmaceutical Distribution Sector," *European Journal of Operational Research* 273, no. 1 (2019): 131–145.

31. J. C. Pina-Pardo, D. F. Silva, and A. E. Smith, "The Traveling Salesman Problem With Release Dates and Drone Resupply," *Computers & Operations Research* 129 (2021): 105170.

32. C. Archetti, D. Feillet, A. Mor, and M. G. Speranza, "An Iterated Local Search for the Traveling Salesman Problem With Release Dates and Completion Time Minimization," *Computers & Operations Research* 98 (2018): 24–37.

33. M. F. Anjos, B. Gendron, and M. Joyce-Moniz, "Increasing Electric Vehicle Adoption Through the Optimal Deployment of Fast-Charging

Stations for Local and Long-Distance Travel," *European Journal of Operational Research* 285 (2020): 263–278.

34. IBM, "Introducing IBM ILOG CPLEX Optimization Studio 20.1.0," 2021, https://www.ibm.com/docs/en/icos/20.1.0?topic=documentation-introducing-ilog-cplex-optimization-studio-2010.