CIRRELT-2022-01

Delivering in Congested Urban Environments: The Benefits of Several Depots and a Diverse Fleet

Carise E. Schmidt
Arinei C. L. Silva
Maryam Darvish
Leandro C. Coelho

January 2022

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval, sous le numéro FSA-2022-001

Bureau de Montréal
Université de Montréal
C.P. 6128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec
Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

Delivering in Congested Urban Environments: The Benefits of Several Depots and a Diverse Fleet[ǂ]

Carise E. Schmidt[1], Arinei C.L. Silva[2], Maryam Darvish[3], Leandro C. Coelho[3,*]

1. LABICON, Instituto Federal de Santa Catarina (IFSC), Brazil
2. PPGMNE, Universidade Federal do Paraná, Brazil
3. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Operations and Decision Systems, Université Laval, Québec, Canada

Abstract. This paper considers an extension of classical distribution problems and introduces the time-dependent fleet size and mix multi-depot vehicle routing problem. This is an important class of problems with applications in urban logistics and service design. The solution to this problem impacts the performance of distribution companies, and can help design policies to improve traffic and congestion issues. We propose a mathematical model for this new and challenging problem, along with several generic and problem-specific valid inequalities. A powerful matheuristic is proposed to solve large instances of the problem generated from real traffic data. Our matheuristic is also assessed on a set of instances from the literature. For the newly introduced problem, we provide good solutions with an important reduction in the execution time. Our computational results demonstrate the importance of using a powerful algorithm to solve complex optimization problems. The solutions demonstrate that in dense urban environments, it is important to properly explore all facilities and use all types of vehicles available in order to better serve customers and improve the performance.

Keywords: Urban logistics, integrated logistics, service design, vehicle routing, time-dependent.

Acknowledgements. This project was partly funded by the Natural Sciences and Engineering Council of Canada (NSERC) under grants 2020-00401 and 2019-00094. This support is greatly acknowledged. We thank Hamza Heni and Khaled Belhassine for their help in creating the instances.

[ǂ] Revised version of the CIRRELT-2020-13.

---

* Corresponding author: leandro.coelho@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2022

1. Introduction

City logistics faces several major challenges, among which congestion and traffic are of great concern. Finding an efficient and effective way to transport goods in urban areas has become the definition of city logistics itself (Savelsbergh and Van Woensel, 2016). This efficient and effective distribution within cities is not only related to cost reduction, but it also aims to alleviate and avoid congestion and, consequently, greenhouse gas emissions (GHG). After all, research on green freight transportation highlights the effect of congestion on fuel consumption and GHG emissions (Demir et al., 2011; Speranza, 2018; Rincon-Garcia et al., 2020).

Several decisions directly impact distribution costs and GHG emissions in a city logistics context. Other than purely distribution decisions, the use of different depots, timing, and fleet composition are the main decisions to optimize (Koç et al., 2016b). Research and practice show that all these decisions are interdependent and, therefore, must be jointly optimized. Several integrated supply chain optimization problems that combine different decisions have been proposed in the literature. Among them and related to our work, we can mention location decisions and capacity planning (Fu et al., 2020), time-dependent issues related to traffic, transit-time, and urban distribution in dense areas (Rincon-Garcia et al., 2020; Tyworth and Zeng, 1998; Snoeck and Winkenbach, 2020; Jaballah et al., 2021).

The multi-depot vehicle routing problem (MDVRP) is a direct generalization of the vehicle routing problem (VRP) that explores routing decisions from several depots (see Montoya-Torres et al. (2015) for a review). At the same time, the optimization of the fleet size and mix can have important tactical and operational advantages considered in the fleet size and mix VRP (FSM-VRP) (Golden et al., 1984; Renaud and Boctor, 2002). These two problems have also been combined in what is known as the fleet size and mix multi-depot vehicle routing problem (FSM-MDVRP) (Salhi and Sari, 1997; Salhi et al., 2014; Vidal et al., 2014; Mancini, 2016; Lahyani et al., 2018). These problems are very applicable to real-world situations in which a fleet of heterogeneous vehicles is available to perform distribution.

One of the main drawbacks of these problems is that routing decisions are optimized without taking traffic congestion into account. Congestion evolves throughout the day and can cause some routes to be excessively delayed, making them infeasible due to time window or maximum duration violations. In this paper, we study the time-dependent FSM-MDVRP (TD-FSM-MDVRP) to account for these issues. Our main contribution lies in considering a routing costs dependent on the time of the day, i.e., a time-dependent routing cost. We use a real traffic database to estimate the routing costs considering the travel time.

In addition to introducing the TD-FSM-MDVRP, the contributions of this work are as follows. We formally define and model the problem and solve it with both exact and approximate approaches. We design a fast and efficient heuristic algorithm and post-optimization procedures based on mathematical programming to polish its solutions. Several instances of the problem are developed using real traffic data from Québec City. These instances are used to show the effectiveness of the proposed solution algorithms. They are generated from real data helping us gain insights on the impact of location and routing decisions on city logistics issues. The results indicate that while vehicles of medium capacity (and cost) are used more often, it is important to have a few smaller and larger vehicles available, as these also help decrease costs and improve performance. The results also indicate that to serve an urban area, it is beneficial to exploit several small depots geographically dispersed instead of using fewer larger facilities. Moreover, our algorithms are evaluated on classical FSM-MDVRP instances, and compared with two competing algorithms from the literature, showing their performance in achieving good solutions in reasonable execution time, including a new best known solution.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of the studies related to our problem. In Section 3, we present the formal description of the problem and its mathematical formulation. Our proposed matheuristic is described in Section 4. This is followed by the results of extensive computational experiments in Section 5 where we assess the performance of our algorithms on TD-FSM-MDVRP and the FSM-MDVRP. Finally, we draw the conclusions of our study in Section 6.

2. Literature review

This section reviews time-dependent routing problems and the two building blocks of our problem, namely the FSM-MDVRP and the MDVRP itself. We provide a brief review of the recent state-of-the-art in these domains.

The MDVRP is a well-known generalization of the standard VRP, in which a more realistic situation is considered by optimizing vehicle routes to reduce logistics costs in multi-depot networks (Li et al., 2018). In this problem, multiple vehicles from several depots perform deliveries to a set of customers. Each vehicle must start from and end at the same depot, minimizing total travel costs. A few exact methods are available in the literature, while several heuristic procedures have been proposed to solve it. Baldacci and Mingozzi (2009) have developed an exact method for solving different classes of VRPs, including the MDVRP. This algorithm is based on a set partitioning formulation that first applies a procedure to generate routes, followed by three bounding procedures to reduce the number of variables. Contardo and Martinelli (2014) propose an exact algorithm for the MDVRP under capacity and route length constraints that uses a vehicle-flow and a set partitioning formulation. Several classes of valid inequalities are added to strengthen both formulations, including new ones to forbid cycles. Alternatively, Salhi and Sari (1997) have proposed a multi-level heuristic for the MDVRP that was also tested on the problem with a heterogeneous fleet. Ho et al. (2008) have developed a simple and a hybrid genetic algorithms. The first one generates initial solutions randomly, while the second one incorporates the Clarke and Wright savings method, the nearest neighbor heuristic, and the iterated swap procedure. Yu et al. (2011) also propose a parallel improvement ant colony optimization for the MDVRP, applied when a virtual central depot is added to the problem. Vidal et al. (2013) solve the periodic MDVRP combining a population-based evolutionary search, a neighborhood-based metaheuristics, and an advanced population-diversity management.

The FSM-VRP was introduced by Golden et al. (1984) and is a well-established class of routing problems combining complex assignment and routing decisions under the objective of minimizing fixed vehicle costs and variable routing costs. This problem differs from the heterogeneous VRP as the fleet is considered to be unlimited (Koç et al., 2016b). Arguing that a fleet of different capacities is usually available, Salhi and Sari (1997) incorporate heterogeneous vehicles in the multi-depot context. They propose a multi-level composite heuristic based on integrating and modifying efficient heuristics designed for the single depot FSM-VRP. Later, Salhi et al. (2014) proposed a mixed-integer linear programming formulation for the problem with a new set of valid inequalities, and a variable neighborhood search metaheuristic. Vidal et al. (2014) develop a unified algorithmic framework tackling different classes of MDVRPs with and without mixed-fleet, and with unlimited fleet size, using a multi-start iterated local search and a hybrid genetic algorithm. To solve a real and new variant of the MDVRP with heterogeneous fleet, multiple periods and different levels of incompatibility constraints, Mancini (2016) proposes an adaptive large neighborhood search metaheuristic. Recently, Lahyani et al. (2018) propose five different formulations for the FSM-MDVRP along with sets of new valid inequalities for each model. The authors implement branch-and-bound and branch-and-cut algorithms for each formulation.

The common assumption in most fleet size and mix vehicle routing studies is the constant traveling time, whereas in reality, travel time and cost vary with respect to traffic behavior. The average traveling time (or speed) can be defined as a function of the time of the day. Malandraki and Daskin (1992) introduced and modeled the time-dependent VRP (TD-VRP). For a comprehensive review of the literature, see Gendreau et al. (2015). Despite the recent interest in time-dependent routing, studies on variants of the TD-VRP are scarce and mainly restricted to cases with time windows (e.g., Figliozzi (2012); Taş et al. (2014); Heni et al. (2019)). In the green vehicle routing and scheduling problem (Xiao and Konak, 2016), heterogeneous vehicles and time-varying traffic congestion are considered. Only one paper studies the time-dependent location routing problem (TD-LRP) (Schmidt et al., 2019), where a limited fleet of homogeneous vehicles is considered and a single depot must be selected.

Despite the practical importance of the TD-VRP with a heterogeneous fleet of vehicles and multiple depots for distribution companies, city officials, and policy makers, there is an evident gap regarding these problems in the literature. This lack has inspired this paper to incorporate fleet optimization and depot choice in a time-dependent routing context.

3. Problem description and formulation

In this section, we formally describe the TD-FSM-MDVRP and present its mathematical formulation. The problem is defined on a directed graph $G = (N, A)$, where N represents the node set and A is the set of arcs. Let N_d be the set of depots and N_c be the set of customers. We also consider a set of dummy nodes called terminals, denoted by N_t, to be used by each type of vehicle as they return to the depot, such that $N_d \cap N_c = \emptyset$, $N_d \cap N_t = \emptyset$ and $N_c \cap N_t = \emptyset$.

Let K be the set of $|K|$ types of vehicles, each with a limited capacity Q_k. Each terminal is linked to only one type of vehicle, i.e., for each depot $i \in N_d$, we define $\delta_k(i)$ as a unique subset of terminals linked to vehicles of type $k \in K$, that is, $\delta_k(i) \subseteq N_t$. Therefore, we have one terminal node $v \in \delta_k(i)$ for each type of vehicle from the fleet, in each depot. This notation is used to indicate that each vehicle has to return to the depot it belongs to. Let also A_{dc}, A_{cc}, A_{ct} be arc sets such that each arc (i, j) is given from the Cartesian products as $A_{dc} = N_d \times N_c; A_{cc} = N_c \times N_c$, $i \neq j$, and $A_{ct} = \cup_{k \in K} \cup_{i \in N_d} \{N_c \times \delta_k(i)\}$ such that $A = A_{dc} \cup A_{cc} \cup A_{ct}$.

The graph is time-dependent, meaning that as the traffic condition changes, the time it takes to traverse arc (i, j) also changes. We define H as the set of time intervals, where an interval is a period of time over which traffic pattern is constant. A deterministic travel time t_{ij}^h is associated wih each arc $(i, j) \in A$ during each interval $h \in H$. We consider a single period (i.e., a day) divided into $m + 1$ intervals, where each time interval $h \in H = \{0, 1, .., h, .., m\}$ has the same length of $\overline{T}$ seconds. Therefore, $[h\overline{T}, (h + 1)\overline{T} - \varepsilon]$ represents the time interval associated with h, where ε is a very small positive number representing the smallest time unit, i.e., one second.

The demand and the service time associated with customer $i \in N_c$ are denoted by q_i and s_i. The fixed cost for each vehicle of type $k \in K$ is denoted by F_k. Let W_i be the capacity of depot $i \in N_d$ and C be a coefficient used to convert the travel time into its cost equivalent.

We define our formulation based on the following binary variables: x_{ij}^h indicate whether arc (i, j) is traversed by a vehicle during interval h; f_i^d take value one if node $i \in N \backslash N_d$ is associated with depot $d \in N_d$. We also define the following continuous variables: a_i represent the departure time from customer $i \in N_c$; b_{ij} represent the departure time from depot $i \in N_d$ (in this case j is a customer) or the arrival time at terminal $j \in N_t$ (in this case, i is a customer); u_i represent a bound on the accumulated deliveries to all customers already visited before departing from customer $i \in N_c$. The minimum value for these variables is the accumulated delivery, and the maximum value is the capacity of the vehicle used for the delivery. Therefore, if the total demand delivered by a vehicle is less than the vehicle capacity, this variable may not represent the accumulated demand delivered for all customers from the route, as the difference between the vehicle capacity and the total demand of that route represents a slack in these variables.

The mathematical formulation is as follows:

$$\min \sum_{(i,j) \in A} \sum_{h \in H} C t_{ij}^h x_{ij}^h + \sum_{k \in K} \sum_{j \in N_c} \sum_{d \in N_d} \sum_{v \in \delta_k(d)} \sum_{h \in H} F_k x_{jv}^h \tag{1}$$

subject to:

$$\sum_{i \in (N_c \backslash \{j\}) \cup N_d} \sum_{h \in H} x_{ij}^h = 1, \quad \forall j \in N_c \tag{2}$$

$$\sum_{j \in (N_c \setminus \{i\}) \cup N_t} \sum_{h \in H} x_{ij}^h = 1, \quad \forall i \in N_c \tag{3}$$

$$\sum_{j \in N_c} \sum_{h \in H} x_{dj}^h = \sum_{j \in N_c} \sum_{k \in K} \sum_{v \in \delta_k(d)} \sum_{h \in H} x_{jv}^h, \quad \forall d \in N_d \tag{4}$$

$$u_i - u_j + \max_{k \in K}\{Q_k\} \sum_{h \in H} x_{ij}^h \leq \max_{k \in K}\{Q_k\} - q_j, \quad \forall i, j \in N_c, i \neq j \tag{5}$$

$$q_i \leq u_i \leq \max_{k \in K}\{Q_k\}, \quad \forall i \in N_c \tag{6}$$

$$u_v \geq u_j - \max_k\{Q_k\}(1 - \sum_{h \in H} x_{jv}^h), \quad \forall j \in N_c, \forall v \in \delta_k(i), \forall k \in K, \forall i \in N_d \tag{7}$$

$$u_v = Q_k, \quad \forall v \in \delta_k(i), \forall k \in K, \forall i \in N_d \tag{8}$$

$$\sum_{i \in N_c} f_i^d q_i \leq W_d, \quad \forall d \in N_d \tag{9}$$

$$b_{ij} + s_j + t_{ij}^h - 2\overline{T}|H|(1 - x_{ij}^h) \leq a_j \leq b_{ij} + s_j + t_{ij}^h + \overline{T}|H|(1 - x_{ij}^h), \quad \forall i \in N_d, \forall j \in N_c, \forall h \in H \tag{10}$$

$$a_i + s_j + t_{ij}^h - 2\overline{T}|H|(1 - x_{ij}^h) \leq a_j \leq a_i + s_j + t_{ij}^h + \overline{T}|H|(1 - x_{ij}^h), \quad \forall i, j \in N_c, i \neq j, \forall h \in H \tag{11}$$

$$a_j + t_{jv}^h - 2\overline{T}|H|(1 - x_{jv}^h) \leq b_{jv} \leq a_j + t_{jv}^h + \overline{T}|H|(1 - x_{jv}^h), \quad \forall j \in N_c, \forall v \in N_t, \forall h \in H \tag{12}$$

$$b_{ij} \leq (\overline{T}|H| - \varepsilon) \sum_{h \in H} x_{ij}^h, \quad \forall (i,j) \in A_{ct} \tag{13}$$

$$\sum_{j \in (N_c \setminus \{i\}) \cup N_t} \sum_{h \in H} h\overline{T} x_{ij}^h \leq a_i \leq \sum_{j \in (N_c \setminus \{i\}) \cup N_t} \sum_{h \in H} [(h+1)\overline{T} - \varepsilon] x_{ij}^h, \quad i \in N_c \tag{14}$$

$$\sum_{h \in H} h\overline{T} x_{ij}^h \leq b_{ij} \leq \sum_{h \in H} [(h+1)\overline{T} - \varepsilon] x_{ij}^h, \quad \forall (i,j) \in A_{dc} \tag{15}$$

$$\sum_{d \in N_d} f_i^d = 1, \quad \forall i \in N \setminus N_d \tag{16}$$

$$f_v^d = 1, \quad \forall v \in \delta_k(d), \forall k \in K, \forall d \in N_d \tag{17}$$

$$f_i^d + \left(\sum_{h \in H} x_{ij}^h - 1 \right) \leq f_j^d \leq f_i^d + \left(1 - \sum_{h \in H} x_{ij}^h \right), \quad \forall d \in N_d, \forall i \in N_c, \forall j \in N \setminus N_d, i \neq j \tag{18}$$

$$f_i^d \geq \sum_{h \in H} x_{di}^h, \quad \forall d \in N_d, \forall i \in N \setminus N_d \tag{19}$$

$$x_{ij}^h \in \{0, 1\}, \quad \forall (i,j) \in A, \forall h \in H \tag{20}$$

$$f_i^d \in \{0, 1\}, \quad \forall d \in N_d, \forall i \in N \setminus N_d \tag{21}$$

$$a_i \in \mathbb{R}_+, \quad \forall i \in N_c \tag{22}$$

$$u_i \in \mathbb{R}_+, \quad \forall i \in N_c \cup N_t \tag{23}$$

$$b_{ij} \in \mathbb{R}_+, \quad \forall (i,j) \in A \setminus A_{cc}. \tag{24}$$

The objective function (1) minimizes the sum of routing and fixed vehicle costs. Assignment constraints (2) and (3) ensure that each customer is visited exactly once. Constraints (4) enforce that vehicles leave only from the selected depots and that they must return to their associated terminal node. Constraints (5) and (6) are the extensions of the Miller-Tucker-Zemlin subtour elimination, here adapted to account for the heterogeneous fleet by using the capacity of the largest vehicle. Constraints (7) control the accumulated demand delivered and constraints (8) restrict the vehicle capacity. Since this constraint forces the accumulated delivered demand to be equal to the vehicle's capacity, it produces a slack on these variables if the demand of customers assigned to this route is less than the capacity of its vehicle. The total demand assigned to a depot cannot exceed its capacity as imposed by constraints (9). We control the departure time from the first customer of each route using constraints (10). Similarly, constraints (11) control the departure time from all the other customers. The same control is applied for the arrival time to the terminals with constraints (12). Constraints (13) ensure that the vehicle performs its route within the planning horizon. The departure time from each node i is linked to its corresponding time intervals by (14) and (15). Constraints (10)–(15) enforce that if arc (i, j) is traversed by a vehicle in time interval h, then h is the time interval considered in the departure from the origin i. Constraints (16) guarantee that each customer and terminal are associated with only one depot. Constraints (17) link terminals to their respective depots. Constraints (18) and (19) enforce the customer-depot assignment. Finally, constraints (20)–(24) enforce integrality and non-negativity conditions on the variables.

3.1. Valid inequalities from the literature

As suggested in Kara et al. (2004) and applied for homogeneous fleet in Schmidt et al. (2019), constraints (6) can be lifted as in (25) (adapted for a heterogeneous fleet):

$$u_i - u_j + \max_{k \in K}\{Q_k\} \sum_{h \in H} x_{ij}^h + \left(\max_{k \in K}\{Q_k\} - q_i - q_j\right) \sum_{h \in H} x_{ji}^h \leq \max_{k \in K}\{Q_k\} - q_j, \quad \forall i, j \in N_c, i \neq j. \tag{25}$$

The problem can be further reduced in size by removing some variables associated with the departure interval of the vehicles. Based on Schmidt et al. (2019), we can remove several arc traversal variables. We consider two cases as presented by constraints (26) and (27). Arc (i, j) can be removed for interval h if the sum of the shortest time to traverse any incoming arc (from the depot or any other customers) to customer i and the service time of i exceeds the upper bound of that interval, imposed by constraints (26). This logic is also true if the time to reach the closest customer to any depot in interval h and the service time required for customer i exceeds the upper bound of interval h, as imposed by constraints (27).

$$x_{ij}^h = 0 \quad \forall i \in N_c \mid \left(\min_{a \in (N_c \setminus \{i\}) \cup N_d} \{t_{ai}^h\} + s_i \right) \geq (h+1)\overline{T}, \quad \forall j \in N_c \cup N_t, j \neq i, \forall h \in H \tag{26}$$

$$x_{ij}^h = 0 \quad \forall i \in N_c \mid \left(\min_{\substack{a \in N_d \\ b \in N_c}} \{t_{ab}^h\} + s_i \right) \geq (h+1)\overline{T}, \quad \forall j \in N_c \cup N_t, j \neq i, \forall h \in H. \tag{27}$$

Similarly, Schmidt et al. (2019) remove some variables related to the terminal nodes. They test if the last interval is not sufficient to leave a customer and reach the depot (including the travel time to the depot in the last time interval (m), or the sum of the shortest arrival time to a customer, its service time, and the shortest time to reach a terminal):

$$x_{ij}^m = 0, \quad \forall i \in N_c, \forall j \in N_t \mid t_{ij}^m \geq \overline{T} \tag{28}$$

$$x_{ij}^m = 0, \quad \forall j \in N_c \mid \left(\min_{a \in N_c \setminus \{j\}} \{t_{aj}^m\} + s_j + \min_{b \in N_t} \{t_{jb}^m\} \right) \geq \overline{T}, \quad \forall i \in N_c, i \neq j. \tag{29}$$

Finally, we improve the routing part of the model by forbidding subtours of sizes two and three:

$$\sum_{h \in H} x_{ij}^h + \sum_{h \in H} x_{ji}^h \leq 1, \quad \forall i, j \in N_c, i \neq j \tag{30}$$

$$x_{ij}^h + x_{ji}^h \leq 1, \quad \forall i, j \in N_c, i \neq j, \forall h \in H \tag{31}$$

$$\sum_{h \in H} (x_{ij}^h + x_{ji}^h + x_{iv}^h + x_{vi}^h + x_{jv}^h + x_{vj}^h) \leq 2, \quad \forall i, j, v \in N_c, i \neq j \neq v. \tag{32}$$

3.2. New and problem-specific valid inequalities

First, we establish lower bounds for each type of cost of the objective function. A lower bound can be set on the fixed cost of using vehicles as in (33). Let f' be the minimum cost required to serve all customers, considering the capacity of different types of vehicles available and customers' demands. It is obtained as a solution to a variable cost and size bin packing problem.

$$\sum_{k \in K} \sum_{j \in N_c} \sum_{d \in N_d} \sum_{v \in \delta_k(d)} \sum_{h \in H} F_k x_{jv}^h \geq f'. \tag{33}$$

We also set a lower bound on the routing costs. Let k' be the minimum number of vehicles required to meet all demands considering the maximum vehicle capacity of $\max_{k\in K}\{Q_k\}$, also obtained as a solution of a bin packing problem. First considering all intervals, we identify the minimum cost arc leaving each depot to reach every customer $s_{di} = \min_{h\in H}\{t^h_{di}\}$, $\forall d \in N_d, \forall i \in N_c$. Second, we do the same for each arc returning from every customer i to every depot d as $r_{id} = \min_{h\in H}\{t^h_{id}\}$. Then, we arrange all the $|N_c|$ values obtained for each depot d for s_{di} and r_{id} in an increasing order, defining s^n_{di} and r^n_{id} as the s_{di} and r_{id} values in the n^{th} position. Moreover, for each $d \in N_d$, let $f^n_d = s^n_{di} + r^n_{id}$. Once again, we arrange all the values obtained for f^n_d in an increasing order, where f^n is the value in the n^{th} position. We then establish that at least the first k' values of the vector f^n will be considered as routing costs to leave and return to any depot. Finally, we set $g_c = \min_{\substack{a\in N_c\setminus\{c\}\\ h\in H}}\{t^h_{ca}\}, \forall c \in N_d \cup N_c$. We also arrange g_c in an increasing order. Therefore, g^n is defined as the value in the n^{th} position. This term allows us to establish a lower bound on routing costs for reaching customers.

$$\sum_{(i,j)\in A} \sum_{h\in H} t^h_{ij} x^h_{ij} \geq \sum_{n=1}^{k'} f^n + \sum_{n=1}^{|N_c|-k'} g^n. \tag{34}$$

The minimum number of vehicles (of largest capacity) k' is set as a lower bound for all departures from all depots.

$$\sum_{i\in N_d} \sum_{j\in N_c} \sum_{h\in H} x^h_{ij} \geq k'. \tag{35}$$

Finally, for each arc there can be at most one period in which it is used:

$$\sum_{h\in H} x^h_{ij} \leq 1, \quad \forall (i,j) \in A. \tag{36}$$

4. Matheuristic algorithm

The general structure of our proposed matheuristic is based on evolutionary search. It is inspired by some elements proposed in Koç et al. (2016a), adjusted for a time-dependent problem.

The algorithm encompasses three main phases: initialization, evolution, and improvement. In the first phase, we generate an initial population of solutions. The second phase creates new solutions and selects promising ones among them. The final phase comprises an additional process to improve the quality of the selected solutions by solving a restricted mathematical model. An overview is presented in Algorithm 1 and detailed as follows.

The initialization consists of creating solutions by assigning customers to depots and then constructing routes to serve them. Then, using different operators, we populate the initial solution set until it reaches η individuals. In the evolution phase, we perform crossover, mutation, and to select the surviving population. When the number of iterations without improvement (δ) in the current population is reached, this phase ends, and the intensification phase begins. These first two phases try to determine the best departure time from the depot to take advantage of the time-dependent graph. The third phase applies a method to restrict our mathematical model from Section 3, using a set of best solutions taken from the previous step. It is the final phase of our matheuristic.

Algorithm 1 General structure of the matheuristic algorithm

Generation of initial solutions.{`//Initialization: Section 4.1`}
while initial population $\leq \eta$ **do**
    Increase the population.
end while
while the number of iterations without improvement $< \delta$ **do** {{`//Evolution: Section 4.2`}}
    Select a set of parents from the current generation.
    Apply crossover operators and directed mutation operators.
    Update the surviving population.
end while
Return the ϑ best solutions.
for each one of the ϑ best solutions **do** {{`//Improvement: Section 4.3`}}
    Let free all binary variables from the selected set that take value 1 on this solution.
end for
Fix all non-free variables to zero and solve the mathematical model.
Return the best solution.

4.1. Initialization phase

The initialization of our algorithm consists of creating a set of initial solutions by assigning customers to depots, followed by route construction. The assignment is based on the travel time between each depot and a customer. Different assignments can be generated because the travel time varies for each interval $h \in H$. Therefore, we generate four time matrices based on: (i) the travel time in each interval $h \in H$, (ii) maximum times, (iii) minimum times, and (iv) average travel times.

Given any of these four matrices, we assign each customer $i \in N_c$ to its closest depot. This is performed based on the shortest travel time between the depot and the customer, for both departure and return. Then, we sort customers in a non-increasing order of travel time. We assign customers to their closest depot, considering the depot capacity constraints. This procedure continues until all customers are assigned. At this point, we try to improve the assignment by closing depots. We first close one depot and try to assign its customers to other ones. The same procedure is applied for closing two depots.

After the allocation phase, we start the construction of routes for each vehicle type. This allows creating diverse routes that exploit the heterogeneous fleet. Aiming to diversify the solutions, we apply two heuristics as follows.

In the first heuristic, the routes are constructed sequentially, i.e., once the capacity of the vehicle is reached for the first route, the second route can be started. For each depot, we randomly select a customer and insert it at the end of the current route. This operation is repeated for all open depots and continues until the complete solution is generated. This process is repeated for each of the four time matrices. For each of the generated routes, we apply an improvement step described next.

The second heuristic uses the Clarke and Wright (1964) algorithm to generate routes. However, we use the four travel time matrices instead of using a distance matrix, as previously described. Note that the selected time matrix affects the assignment of customers to depots, and now again, it affects the route generation.

In what we call the intra-route improvement step, a permutation procedure is applied sequentially and iteratively. The departure time of the routes is also evaluated at different times, for example, at every 30 min of the planning horizon. The improvement procedure uses either the full route or partial permutations. In the full route permutation, all routes containing up to ν_1 customers are explored, and we determine the best sequence of customers by enumerating all permutations. A partial permutation, however, is applied to a subgroup of ν_2 customers. The position of these customers in this subgroup is modified until the best sequence is obtained. Then, the position of the first customer in the subgroup is fixed, and the same procedure is applied to the next ν_2 remaining customers. This process continues until all possible improvements for the entire route are taken place. A numerical example is presented in Figure 1.

Initial route	**1**	**2**	**3**	4	5
Partial optimal 1	**3**	**1**	**2**	4	5
	3	**1**	**2**	**4**	5
Partial optimal 2	3	**4**	**2**	**1**	5
	3	4	**2**	**1**	**5**
Partial optimal 3	3	4	**5**	**1**	**2**
	3	4	5	**1**	**2**
Partial optimal 4	3	4	5	**1**	2
	3	**4**	5	1	**2**
Local optimal	2	3	5	1	4

Figure 1: A numerical example for the intra-route partial permutations improvement step

In the second phase of the initialization algorithm, new solutions are generated to increase the size of the initial population. For this purpose, large neighborhood search operators are applied to the initial solutions, with the selection of a removal and an insertion operator:

Removal operators:

- **Depot swap:** Applied by Hemmelmayr et al. (2012) and Koç et al. (2016a), this operator randomly selects an open depot and closes it, followed by the opening of a closed depot. All customers assigned to the closed depot are then kept in a removal list L_r.

- **Depot opening:** This operator is an adaptation of the one used by Hemmelmayr et al. (2012) and Koç et al. (2016a). It randomly opens a closed depot. Then n' customers are removed from the current solution and added to the list L_r. The removal criterion is the shortest average travel time between the customer and the newly opened depot.

- **Random removal:** Used by Ropke and Pisinger (2006) and Koç et al. (2016a), this operator randomly selects n' customers and adds them into the removal list L_r.

- **Depot time exchange:** Proposed by Koç et al. (2016a), it is based on the first removal operator but differs in the criterion applied to choose the new open depot. We adapted it to consider the shortest average travel time between the closed and open depots.

Insertion operators:

- **Greedy insertion:** it is an adaptation of an operator from Ropke and Pisinger (2006). A customer is randomly selected from the list L_r, and a route to insert it is also randomly selected. Its insertion is tested into all possible positions, choosing the one that minimizes the travel cost of the current route. If the insertion is not feasible in the existing routes, the insertion is also evaluated for new routes from already opened depots.
- **Travel cost – sequential greedy insertion:** This operator is based on the previous one but the route is not chosen randomly. We test the position in all routes to minimize the travel cost.
- **Travel cost – greedy insertion:** Similar to the previous two, this is a greedy operator that inserts the customer to the position with the lowest increase in travel cost. All customers are evaluated, and the one with the best insertion cost is selected. After each insertion, the remaining non-assigned customers are reevaluated.
- **Regret insertion:** it is also an adaptation of the operator applied by Ropke and Pisinger (2006), and it is performed to avoid a customer being assigned to a bad position in a route. This operator is more complicated than the previous ones and requires a higher number of operations. For each customer, the regret is calculated based on the cost difference between the insertion in its best and second best positions. Then, the customer with the highest regret is selected. All possibilities are tested on existing and new routes, as long as they remain feasible. After an insertion, all customers from the removal list are reevaluated.

The number n' of removed customers is randomly chosen from an interval calculated as percentages of the total number of customers. To create a new solution, we randomly select one of the solutions generated in the first phase of the initialization. The complete solution is added to the initial population. This process is repeated until the initial population reaches the size of η.

4.2. Evolution phase

In this phase, we improve and generate new solutions by using crossover and mutation operators. The goal is to diversify the search and improve the quality of the solutions.

To generate solutions by the crossover operator we apply the *Partially Mapped Crossover* (Goldberg and Lingle, 1985). Initially, two solutions (parents) are randomly selected, P_1 and P_2. Two cut-off points indicating the number of genes to be crossed are determined, referring to the number of customers to be removed from the sequence. This number is randomly selected based on percentages of the number of customers in the instance. Once the position of the first cut-off point is determined, the genes that are between the two points are crossed for the offspring generation. It is probable that during the genetic material exchange, chromosomes end up having repeated genes. In this case, all those outside the cut-off region are replaced with those on the same *locus* but in another chromosome. Having generated the sequence of customers for each child solution, following the configurations of their parents with respect to the order of the sequence and the number of customers at the respective routes, these children are placed in routes. Finally, we check the feasibility of every solution and discard all the infeasible ones.

Mutation involves creating new children solutions by copying another solution to change the current chromosome genes. In order to improve the quality of each new generation, we use directed mutation in the form of an improvement heuristic. The procedure consists of two steps: a removal followed by an insertion. In the first step, n' customers are allocated to a removal list L_r. Again, n' is randomly selected based on percentages of the number of customers. During the insertion step, the removed customers are relocated in the incomplete solution. Both operators are randomly selected from the following ones.

Removal operators:

- **Neighborhood removal:** This operator is inspired by Ropke and Pisinger (2006); Demir et al. (2012), Koç et al. (2016a). The general idea is to remove the n' customers that are "extreme" with respect to the travel time. We identify these customers by calculating the sum of the time required to arrive at each customer coming from its previous node and the time required to go from this customer to the subsequent node in the route. For each customer, we consider the corresponding interval h associated with each inbound and outbound arc.

- **Worst travel time removal:** This operator removes the n' customers that are "extreme" with respect to the insertion cost. By inserting a customer into a route, we change the costs associated with the new arc added to the route and the costs of all the subsequent arcs. Therefore, we define the insertion cost as the difference between the total execution time of a route with and without adding each customer. A distance-based version of this operator can be found in Ropke and Pisinger (2006); Demir et al. (2012), and Koç et al. (2016a).

- **Depot removal – efficiency:** Proposed by Koç et al. (2016a), this operator aims to calculate the utilization efficiency of each open depot, expressed by the ratio of the total demand allocated to the depot over its capacity. The depot with the lowest efficiency is closed, and its customers are placed in the removal list L_r.
- **Vehicle removal:** Similar to the previous one, this operator computes the utilization efficiency of each vehicle. This value is expressed by the ratio of the total demand associated with a vehicle to its capacity. All customers associated with the least efficient vehicle are placed in the removal list L_r.

Insertion operators:

- **Greedy insertion – lowest cost:** This operator iteratively assigns customers to routes in the position that minimizes the insertion cost. In this operator, before adding customers to routes, we try to improve the existing solution. Given a solution, we check the possibility of serving the routes with a small vehicle (i.e., lower fixed cost) or allocating the routes to an already open depot with no customers. Then for every customer in the L_r list, we compute its insertion cost, including additional fixed and opening costs. We identify the customer with the lowest insertion cost at each iteration and add it to the route.
- **Greedy insertion – highest cost:** This operator is similar to the previous one, but here we insert the customer with the highest cost into a route.

The evolution and selection procedures are described next.

From the initial population $\Gamma_{initial}$, which has a size η, we randomly select two solutions, P_1 and P_2 and apply the crossover operator. If at least one of the two created children is feasible and new, its cost is computed, and the solution is added to the current generation $\Gamma_{generation}$. We repeat this process until ω new individuals are created. Then, the mutation procedure is iteratively applied to the initial population solutions $\Gamma_{initial}$. If this procedure creates a new solution, we apply the route improvement procedure to these new routes: the permutation of customers and departure time optimization in fixed intervals, say, 30 min. The improved solution is added to the current generation $\Gamma_{generation}$. This procedure is applied either to the entire initial population or until η new individuals are created. Then, we combine $\Gamma_{initial}$ and $\Gamma_{generation}$ which will make the current population Γ containing up to 2η solutions. This population is ordered by cost, and a set $\Gamma_{parents}$ is selected to constitute the next generation. We select the χ_1 best solutions, the χ_3 worst, and among the remaining solutions, we also randomly select χ_2 solutions, so that $\chi_1+\chi_2+\chi_3=\eta$. The process terminates when there is no further improvement to the σ best solutions for δ consecutive iterations.

4.3. Improvement phase

The last phase of our matheuristic aims to improve the quality of the solution. It involves a method to restrict our mathematical model from Section 3, taking into account a set of best solutions from the evolution phase. To do so, we fix many binary variables f_i^d and x_{ij}^h such that only a few variables can be optimized. We take the ϑ best solutions from the evolution phase and compare information about the values of variables f_i^d and x_{ij}^h. We set to zero all these variables that have never been used in any of these ϑ solutions. This allows for some customers to be potentially exchanged, as all used parts of the best ϑ solutions can be combined, and the model can optimize and merge pieces of different solutions.

5. Computational experiments

This section provides details of our instances, the parameter setting of the algorithms, and extensive results, along with an elaborated analysis. The algorithms are coded in C++, and we use Gurobi Optimizer 9.1 with default settings as the mixed-integer programming (MIP) solver with default settings. All computational experiments are conducted on an Intel Core i7 processor running at 3.4 GHz with 64 GB of RAM installed, with the Ubuntu Linux operating system. The solver uses two threads, and a total time limit of 10800 seconds is imposed for each execution. Section 5.1 describes the instances used. The results of detailed computational experiments are provided in Section 5.2.

The initial population sizeis defined as $\eta = 30\sqrt{|N_c|}$. This relation is defined to avoid overpopulating generations in large instances. The number of best solutions used in each generation for the evolution phase (Section 4.2) is a function of the initial population. Therefore, we set $\sigma = 0.1\eta$ for instances with 10 and 20 customers, $\sigma = 0.15\eta$ for instances with 50 customers, and $\sigma = 0.2\eta$ for all the other instances. For the intra-route improvement heuristic, we set ν_1 equal to 10, 9, 8, 7, and 6, respectively, for instances with 10, 20, 50, 80 and 100 or more customers; ν_2 equals to 3 for all configurations. The composition of set $\Gamma_{parents}$ is $\chi_1 = 0.5\eta, \chi_2 = 0.4\eta$ and $\chi_3 = 0.1\eta$.

The range defined for n', the number of customers to remove, differs for each phase of the algorithm. In the initialization phase, it is set between $n_l^{initial} = 0.3|N_c|$ and $n_u^{initial} = 0.6|N_c|$, while in the evolution phase, the range is defined as $n_l^{cross} = 0.2|N_c|$ and $n_u^{cross} = 0.4|N_c|$. Finally, for the mutation operator, these bounds are defined by $n_l^{mutation} = 0.2|N_c|$ and $n_u^{mutation} = 0.5|N_c|$. The number of individuals created at each iteration (Section 4.2) is set to $\omega = 0.02\eta$.

We consider that all parameters have integer values. Hence, if necessary, we round the result to the nearest positive integer.

5.1. Instance generation and benchmarks from the literature

We modify the instances used in Schmidt et al. (2019) to fit the fleet size and mix problem studied here. The instances are based on geographical information from the real road network and traffic of Quebec City. A planning horizon of 15 hours (from 6:00 to 21:00) is divided into intervals of either 3600, 5400, or 10800 seconds. These three intervals then differentiate the *large*, *medium*, or *small* instances. The number of customers is 10, 20, 50, 80, or 100, and the number of available depots for each instance is 3 or 5. Each customer's demand and service time are random numbers chosen from $[50, 750]$ units and $[1000, 10800]$ seconds, respectively. We set the number of different types of vehicles in the fleet, $|K|$, to 3 for all instances. For the capacity Q_k of these vehicles, we consider 2000, 4000, and 6000 units, and the fixed cost, F_k, 1000, 1500, and 2000 monetary units, respectively. For each unit of travel time, we set the cost of 0.03 monetary units. We randomly generate the depot capacity, W_i, from a discrete uniform distribution from the interval $[w_l^i, w_u^i]$. We define the lower and upper bounds as percentages of total customer demand, respectively, set at 50% and 85%. All instances, solutions, and detailed results are available at `https://www.leandro-coelho.com/time-dependent-location-routing-problem/`.

To compare the performance of our models and algorithms, we have used a set of 14 instances proposed by Salhi and Sari (1997) for the FSM-MDVRP, inspired from older benchmarks for other VRPs. These instances are commonly used in the VRP literature, as for the multi-level composite heuristic of Salhi and Sari (1997), the variable neighborhood search of Salhi et al. (2014), the hybrid genetic search with advanced diversity control of Vidal et al. (2014), and the alternative formulations and improved bounds for the FSM-MDVRP of Lahyani et al. (2018).

These instances contain between 50 and 360 customers and between two and nine depots. There are five vehicle types whose capacities are generated centered around the value of the vehicle capacity

Q^* of the original instances designed for the MDVRP. The vehicle capacities Q_k along with the vehicle variable cost C_k and the vehicle fixed cost F_k are derived based on the following formulas: $Q_k = (0.4 + 0.2k)Q^*$, $C_k = 70 + 10k$, and $F_k = 0.7 + 0.1k$, with $k \in K$.

We have also used a set of 10 smaller instances generated by Lahyani et al. (2018). These instances were created by randomly selecting subsets of customers from the smaller instances of Salhi and Sari (1997), namely instances 4-55-100 and 4-50-80. They contain two and three depots, from 10 to 30 customers, five vehicle types, and different demands distribution.

5.2. Computational results

We now present the results of our extensive computational experiments. We start our analysis by showing in Section 5.2.1 the results from the mathematical model of Section 3. Then, we compare these results with the ones in which all valid inequalities are added. We continue the analysis by presenting in Section 5.2.2 the results of our matheuristic algorithm. Finally, we evaluate the performance of our matheuristic algorithm to solve two sets of FSM-MDVRP instances from the literature in Section 5.2.3.

5.2.1. Results of the mathematical model

We now present the average results obtained by solving the proposed mathematical formulation of Section 3, when provided with an initial solution as follows. We save the best solution obtained at the initialization phase as input for the MIP model.

Table 1 presents the results for instances with 3 and 5 potential depots. In the first two columns of this table, we provide information about the instance. Then, for each different number of depots, we report information about the initial solution (IS), upper bound (UB), lower bound (LB), gap calculated as $100(UB - LB)/UB$, execution time, and finally, the improvement over the initial solution.

Results in Table 1 show how the solver can improve the initial solution, but also that this improvement, typically, decreases as the number of customers increases. The average improvements are 17.56% for instances with 3 and 22.52% for cases with 5 potential depots. Even with these improvements, this table shows a significant difference between the upper and lower bounds, even for instances with a few customers, with an average gap of 31.66% and 27.90% for sets with 3 and 5 depots, respectively.

Table 1: Average results for the mathematical model with an initial solution and without valid inequalities

Size	$\|N_c\|$	3 depots						5 depots					
		IS	Upper Bound	Lower Bound	Gap (%)	Time (s)	Improvement over IS (%)	IS	Upper Bound	Lower Bound	Gap (%)	Time (s)	Improvement over IS (%)
Small	10	301.83	217.51	217.51	0.00	695	27.92	306.84	198.01	198.01	0.00	318	35.40
	20	504.43	421.57	326.04	22.01	10800	16.03	477.64	360.11	294.35	17.66	10800	24.17
	50	952.95	792.52	495.05	37.40	10804	16.28	1011.72	754.93	482.01	34.82	10802	25.16
	80	1371.53	1160.96	525.37	54.43	10801	14.81	1305.50	1090.18	540.60	49.91	10803	15.93
	100	1538.09	1421.86	626.82	55.54	10800	7.65	1317.82	1113.02	551.10	50.95	10801	14.49
Average		933.77	802.88	438.16	33.88	8780	16.54	883.90	703.25	413.21	30.67	8705	23.03
Medium	10	294.38	217.90	217.90	0.00	259	26.17	288.96	211.51	211.51	0.00	567	26.20
	20	498.70	386.27	333.87	12.50	10800	22.57	459.50	341.20	298.74	11.62	10800	25.86
	50	962.77	753.80	500.14	33.21	10800	20.88	903.07	659.76	486.98	25.68	10800	26.86
	80	1423.83	1252.11	565.85	54.46	10800	12.08	1223.73	1109.04	561.10	48.84	10800	9.62
	100	1667.55	1478.35	649.80	54.99	10801	11.28	1368.77	1190.55	572.49	51.71	10800	12.31
Average		969.45	817.69	453.51	31.03	8692	18.60	848.81	702.41	426.16	27.57	8754	20.17
Large	10	290.72	215.99	215.99	0.00	154	25.67	314.93	204.90	204.90	0.00	127	34.25
	20	488.79	365.72	337.14	7.05	9334	25.18	473.58	324.00	316.80	2.18	9371	31.49
	50	957.80	730.62	493.63	32.11	10800	23.72	983.38	653.47	486.36	25.17	10801	33.61
	80	1352.73	1241.69	548.31	54.98	10801	8.55	1254.15	1126.19	571.20	48.66	10802	10.16
	100	1626.71	1558.96	660.89	56.21	10800	4.67	1387.37	1214.49	585.03	51.28	10805	12.30
Average		943.35	822.59	451.19	30.07	8378	17.56	882.68	704.61	432.86	25.46	8381	24.36
Global average		948.85	814.39	447.62	31.66	8617	17.56	871.80	703.42	424.08	27.90	8613	22.52

In what follows, we analyze the average results from our mathematical model when it is fed the initial solution and with all valid inequalities. Table 2 presents the results, showing that using valid inequalities does not significantly affect most upper bounds, compared to the ones reported in Table 1. An average improvement of 3.51% is obtained for 3-depots instances, whereas it is 3.29% for 5-depots instances. However, as expected, adding valid inequalities improves the lower bounds and consequently reduces the gap. As indicated by the LB gap (%) in Table 2, the percentage difference between these lower bounds and those presented in Table 1 is, on average, greater than 7.21% for 3-depots instances. In contrast, it is 5.46 % for instances with 5 depots. For almost all instances with 80 or more customers, the lower bound improvement is greater than 10%. On columns Gap (%), we can observe average gap values of 25.68% and 22.65%, for instances with 3 and 5 depots, respectively. These results highlight the importance of valid inequalities to improve lower bounds. Detailed results (available online) also show that for instances with 10 customers, the solver can prove optimality for all 30 test instances in an average execution time of less than 6 minutes. This time decreases by 16% when the set of valid inequalities are included. However, it is evident that even by providing an initial solution and strengthening the mathematical model by a set of valid inequalities, the average gap remains high after three hours of execution. This reflects the complexity of the problem and the need to apply approximate methods.

5.2.2. Results of the proposed matheuristic

Table 3 shows the average results from our proposed matheuristic algorithm. We detail the best solution obtained in each phase (initialization, evolution, and improvement), followed by the total execution time. Finally, on the last two columns for each depot configuration, we present the improvement of our matheuristic over solutions from the mathematical model with an initial solution and valid inequalities (results from Table 2).

Several interesting observations can be drawn from the analysis of the results in Table 3. First, we observe that our metaheuristic can find, on average, better solutions than the MIP. The improvements are remarkable: 12.03% for instances with 3 depots and 11.07% for those with 5 depots. Except for some instances with 10 or 20 customers in which the average obtained solution slightly worsens (less than 0.27%), we usually can significantly improve the solutions. Compared with the solutions obtained by the mathematical formulation with valid inequalities, we can observe that our method can improve the results by more than 20% in most instances with 80 and 100 customers.

Table 2: Average results for the mathematical model with initial solutions and valid inequalities

Size	$\|N_c\|$	3 depots						5 depots					
		Upper Bound	Lower Bound	Gap (%)	Time (s)	Improvement over results from Table 1 UB gap (%)	LB gap (%)	Upper Bound	Lower Bound	Gap (%)	Time (s)	Improvement over results from Table 1 UB gap (%)	LB gap (%)
Small	10	217.51	217.51	0.00	267	0.00	0.00	198.01	198.01	0.00	140	0.00	0.00
	20	407.74	335.57	17.35	10800	3.52	2.91	348.77	303.38	12.61	10800	3.41	3.10
	50	780.67	521.12	33.22	10800	1.52	5.14	698.74	507.51	26.21	10800	7.97	4.95
	80	1152.12	620.48	45.81	10800	1.07	15.14	1066.31	607.08	42.89	10800	2.11	10.98
	100	1367.86	728.62	46.13	10800	4.24	13.85	1087.75	631.37	42.31	10800	2.46	12.81
Average		785.18	484.66	28.50	8694	2.07	7.41	679.92	449.47	24.81	8668	3.19	6.37
Medium	10	217.90	217.90	0.00	276	0.00	0.00	211.51	211.51	0.00	814	0.00	0.00
	20	382.48	347.52	8.62	10078	0.78	3.91	346.21	306.91	9.55	10800	-0.55	2.56
	50	755.68	528.35	29.21	10800	0.32	5.36	628.95	511.65	18.26	10800	4.87	4.80
	80	1165.55	646.86	44.26	10803	7.56	12.32	1062.24	615.76	41.43	10800	4.36	8.90
	100	1349.02	758.87	42.69	10800	9.81	14.17	1105.45	652.02	41.24	10806	7.49	12.18
Average		774.13	499.90	24.96	8552	3.69	7.15	670.87	459.57	22.10	8804	3.23	5.69
Large	10	215.99	215.99	0.00	71	0.00	0.00	204.90	204.90	0.00	99	0.00	0.00
	20	364.07	350.06	3.30	5390	0.44	3.59	325.41	315.03	3.03	10317	-0.39	-0.52
	50	723.49	520.67	27.85	10800	0.91	5.27	626.84	507.92	18.80	10800	4.14	4.27
	80	1153.70	636.45	44.13	10800	7.70	13.58	1079.50	615.83	42.89	10800	4.36	7.24
	100	1352.66	760.06	42.58	10800	14.82	12.98	1103.97	652.13	40.50	10802	9.19	10.67
Average		761.98	496.65	23.57	7572	4.77	7.08	668.12	459.16	21.04	8564	3.46	4.33
Global average		773.76	493.74	25.68	8272	3.51	7.21	672.97	456.07	22.65	8679	3.29	5.46

Table 3: Average results from our matheuristic algorithm

Size	$\|N_c\|$	3 depots						5 depots					
		Best solution for each phase			Total	Improv. over results Table 2		Best solution for each phase			Total	Improv. over results Table 2	
		Initial.	Evolution	Improv.	time (s)	Best solution	Time	Initial.	Evolution	Improv.	time (s)	Best solution	Time
Small	10	219.77	219.77	217.84	5	-0.17	97.49	200.46	200.46	198.05	4	-0.02	92.42
	20	387.11	380.36	377.90	219	7.21	97.97	330.60	325.78	323.40	92	7.09	99.15
	50	670.96	648.34	643.05	1757	17.63	83.73	623.21	593.51	587.16	680	14.77	93.71
	80	920.57	892.24	884.69	5277	23.03	51.14	894.00	856.61	847.60	3748	20.35	65.30
	100	1090.03	1061.10	1056.05	8171	22.43	24.35	900.44	868.62	860.74	4566	21.39	57.72
Average		657.69	640.36	635.91	3086	14.02	70.94	589.74	569.00	563.39	1818	12.72	81.66
Medium	10	220.47	219.37	218.41	3	-0.26	97.08	212.28	212.03	211.67	4	-0.06	96.40
	20	382.63	372.30	369.77	170	3.17	98.33	335.56	327.93	324.79	68	4.74	99.37
	50	662.54	641.83	639.16	838	14.69	92.24	614.99	587.17	582.59	530	7.12	95.09
	80	978.49	939.50	931.31	3114	20.15	71.18	876.43	844.37	835.91	2027	20.66	81.23
	100	1090.84	1061.38	1052.38	5402	20.95	49.98	897.72	864.05	859.73	4324	22.44	59.98
Average		666.99	646.88	642.21	1905	11.74	81.76	587.40	567.11	562.94	1391	10.98	86.42
Large	10	216.41	216.41	215.99	3	0.00	82.26	205.82	205.81	205.42	2	-0.27	76.02
	20	375.77	365.91	363.77	116	0.06	94.91	327.52	325.33	324.13	83	0.38	99.20
	50	680.65	646.50	643.79	1028	11.01	90.41	626.43	598.68	594.67	859	5.04	92.05
	80	937.02	903.91	898.81	3643	21.57	66.27	881.06	851.67	845.58	1361	21.62	87.40
	100	1113.22	1083.82	1078.44	2937	19.05	72.81	903.48	878.79	871.13	2885	20.79	73.29
Average		664.61	643.31	640.16	1545	10.34	81.33	588.86	572.06	568.18	1038	9.51	85.59
Global average		663.10	643.52	639.42	2179	12.03	78.01	588.67	569.39	564.84	1416	11.07	84.56

Considering the time needed to achieve these improvements, the effectiveness of the proposed method is even better highlighted. The results from Table 3 show that the matheuristic approach can provide better solutions considerably faster. For instances with 3 depots, the average execution time is 2179 seconds, whereas, for cases with 5 potential depots, it equals 1416 seconds, against more than 8000 seconds on average for the MIP, which represents a reduction of more than 81% for instances with 3 depots and more than 84% for 5-depots instances. These results highlight the importance of the matheuristic approach in both aspects: to find high-quality solutions significantly faster.

Moreover, we can combine the strengths of both methods: quality solutions from our matheuristic and lower bounds of the MIP, improved by our valid inequalities. We present in Table 4 an aggregated gap by comparing these values. As presented, the resulting average gap for instances with 3 potential depots is 16.80%, and it is 14.11% for instances with 5 potential depots.

The detailed solutions (available online) indicate that all depots and all vehicles types are used in the final solutions, particularly for large instances. While medium-size vehicles are used more often, these solutions properly exploit the benefits of smaller and larger vehicles, at different costs. In most instances, the solution also serves customers using all available facilities, indicating that in dense urban areas, it is beneficial to use small facilities that are geographically dispersed. This also helps create routes using smaller vehicles, which is also positive for congestion issues within city boundaries.

5.2.3. Matheuristic algorithm performance on the FSM-MDVRP

We now evaluate the performance of our matheuristic on the FSM-MDVRP by using the sets of instances from the literature described in Section 5.1. These instances use different variable costs for each vehicle type, but our problem and methods do not consider this parameter. To this end, we compare our results with the ones obtained by Vidal et al. (2014) and Lahyani et al. (2018). To adapt our proposed method for these instances, we use the same cost for all vehicles during the optimization and update it at the end, according to type of the vehicle used in the final solution.

Table 5 reports and compares the results for the set of instances from the literature. We compare the performance of our proposed algorithm against the results reported by Vidal et al. (2014) and Lahyani et al. (2018) for the same sets. In the first two columns, we provide information about the instances. They indicate the number of depots, customers, vehicle capacity Q^* of the original

Table 4: Average results of the matheuristic solution and lower bound of the MIP

Size	$\|N_c\|$	3 depots			5 depots		
		Matheuristic solution	MIP Lower bound	Gap (%)	Matheuristic solution	MIP Lower bound	Gap (%)
Small	10	215.99	215.99	0.17	205.42	204.90	0.02
	20	364.51	350.03	11.02	324.08	315.10	6.04
	50	641.64	520.71	18.93	594.13	507.93	13.53
	80	894.42	636.43	29.57	846.13	615.92	28.23
	100	1070.55	759.87	30.73	865.20	652.06	26.64
Average		637.42	496.61	18.08	566.99	459.18	14.89
Medium	10	218.41	217.90	0.26	211.67	211.51	0.06
	20	369.78	347.97	5.66	325.14	306.88	5.29
	50	636.52	528.35	17.33	584.36	511.72	12.06
	80	934.61	647.85	30.19	833.43	616.25	26.24
	100	1053.66	755.70	27.76	861.79	651.60	24.23
Average		642.60	499.55	16.24	563.28	459.59	13.58
Large	10	217.84	217.51	0.00	198.05	198.01	0.27
	20	377.70	336.05	3.27	323.64	303.36	2.67
	50	643.21	521.12	18.97	587.90	507.53	14.49
	80	888.79	620.90	28.89	849.05	606.92	27.06
	100	1046.69	728.82	29.32	858.47	632.02	24.86
Average		634.85	484.88	16.09	563.42	449.57	13.87
Global average		638.29	493.68	16.80	564.56	456.11	14.11

instances, and who adapted the instances to this problem. We then show their best solution and the last two columns contain the results of our proposed metaheuristic.

The results of Table 5 show the quality of our matheuristic approach on the instances when we compare it with results obtained by Lahyani et al. (2018). The average time requested to run all of them is 1785 seconds against 8590 seconds as reported by Lahyani et al. (2018). Moreover, our average best solution is equal to 1971.61 against an average upper bound of 2211.04, which represents a reduction of 19.27%. Our metaheuristic also found a new best know solution for instance 3-30-80 (932.14).

In summary, the results show how our proposed matheuristic algorithm, adapted to the FSM-MDVRP, can produce good-quality solutions requiring less processing time compared with the runtime related by Lahyani et al. (2018) for the same set of instances.

6. Conclusions

This paper studies the time-dependent fleet size and mix multi-depot vehicle routing problem. Its main contribution is to extend the time-dependent literature to include more real-world features to this very practical problem. This paper also contributes to the integrated optimization literature as

Table 5: Results from our metaheuristic on the FSM-MDVRP instances

Instance	References	Literature: Vidal et al. (2014) UB	Literature: Lahyani et al. (2018) UB	Proposed metaheuristic: Best solution	Proposed metaheuristic: Time (s)
4-55-100	Salhi and Sari (1997)	–	1398.30	1398.33	170
3-85-100	Salhi and Sari (1997)	–	2131.46	2138.14	948
3-85-160	Salhi and Sari (1997)	–	1483.51	1486.86	1219
4-50-80	Salhi and Sari (1997)	1477.73	1565.27	1504.21	399
4-50-160	Salhi and Sari (1997)	957.73	1021.59	970.13	250
5-75-140	Salhi and Sari (1997)	1569.67	1828.73	1585.55	1044
2-100-100	Salhi and Sari (1997)	2292.64	2660.50	2378.02	1982
2-100-200	Salhi and Sari (1997)	1453.64	1818.69	1531.90	1415
3-100-100	Salhi and Sari (1997)	2208.66	2648.17	2252.62	1370
4-100-100	Salhi and Sari (1997)	2198.91	2626.33	2260.84	737
2-80-60	Salhi and Sari (1997)	2072.18	2565.53	2158.94	1314
4-160-60	Salhi and Sari (1997)	3973.47	5157.49	4124.87	10236
6-240-60	Salhi and Sari (1997)	5887.43	7758.97	6436.88	10800
9-360-60	Salhi and Sari (1997)	8709.26	11638.50	10335.10	10765
2-10-60	Lahyani et al. (2018)	–	441.59	441.59	1
2-15-60	Lahyani et al. (2018)	–	674.32	674.32	1
3-20-80	Lahyani et al. (2018)	–	666.02	666.02	12
3-25-80	Lahyani et al. (2018)	–	787.37	787.37	41
3-30-80	Lahyani et al. (2018)	–	940.49	**932.14**	84
2-10-60*	Lahyani et al. (2018)	–	482.09	482.09	1
2-15-60*	Lahyani et al. (2018)	–	690.38	690.38	1
3-20-100	Lahyani et al. (2018)	–	563.19	563.19	6
3-25-100	Lahyani et al. (2018)	–	676.59	679.12	11
3-30-100	Lahyani et al. (2018)	–	839.98	839.98	29

it presents the first mathematical formulation for the TD-FSM-MDVRP. Using a commercial solver to solve instances generated from Quebec city's real traffic data, we evaluate instances with up to 5 potential depots, 100 customers, and 15 time intervals. We show that adding a pool of initial solutions and considering several problem-specific valid inequalities are essential to obtain and improve the lower bounds for the TD-FSM-MDVRP. However, to achieve high-quality solutions and to reduce computational time, we propose a matheuristic algorithm based on exploring a population of solutions. We have compared the performance of our proposed algorithm against the exact method and shown the importance of a powerful approximate algorithm to solve complex problems as the one studied in this paper. Our matheuristic can find good solutions for large instances and significantly reduce the execution time. We have also evaluated our method on a set of FSM-MDVRP instances from the literature, showing the solution quality and run time, and providing a new best known solution.

This research demonstrates how companies can have financial gains by properly exploring several facilities and vehicle fleets of different sizes. It also indicates that policies restricting the access of large vehicles to dense urban areas do not necessarily impose any financial limitation on logistics operators, as they can adapt and diversify the origins of their routes and the types of vehicles used.

Acknowledgments

This project was partly funded by the Canadian Natural Sciences and Engineering Research Council (NSERC) under grants 2020-00401 and 2019-00094. This support is greatly acknowledged. We thank Hamza Heni and Khaled Belhassine for their help in creating the instances.

References

R. Baldacci and A. Mingozzi. A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming*, 120:347–380, 2009.

G. Clarke and J.W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12:568–581, 1964.

C. Contardo and R. Martinelli. A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. *Discrete Optimization*, 12:129–146, 2014.

E. Demir, T. Bektaş, and G. Laporte. A comparative analysis of several vehicle emission models for road freight transportation. *Transportation Research Part D: Transport and Environment*, 16(5):347–357, 2011.

E. Demir, T. Bektaş, and G. Laporte. An adaptative large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, 223:346–359, 2012.

M. A. Figliozzi. The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics. *Transportation Research Part E: Logistics and Transportation Review*, 48(3):616–636, 2012.

Y. Fu, D. Wu, Y. Wang, and H. Wang. Facility location and capacity planning considering policy preference and uncertain demand under the one belt one road initiative. *Transportation Research Part A: Policy and Practice*, 138:172–186, 2020.

M. Gendreau, G. Ghiani, and E. Guerriero. Time-dependent routing problems: a review. *Computers & Operations Research*, 64:189–197, 2015.

D.E. Goldberg and R. Lingle. Alleles, loci and the traveling salesman problem. In *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, pages 154–159, Pittsburg, US, 1985.

B.L. Golden, A. Assad, L. Levy, and F. Gheysens. The fleet size and mix vehicle routing problem. *Computers & Operations Research*, 11(1):49–66, 1984.

V.C. Hemmelmayr, J.-F. Cordeau, and T.G. Crainic. An adaptive large neighborhood search heuristic for the two-echelon vehicle routing problems arising in city logistics. *Computers & Operations Research*, 39: 3215–3228, 2012.

H. Heni, L.C. Coelho, and J. Renaud. Determining time-dependent minimum cost paths under several objectives. *Computers & Operations Research*, 105:102–117, 2019.

W. Ho, G.T.S. Ho, P. Ji, and H.C.W. Lau. A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence*, 21:548–557, 2008.

R. Jaballah, M. Veenstra, L. C. Coelho, and J. Renaud. The time-dependent shortest path and vehicle routing problem. *INFOR: Information Systems and Operational Research*, 59(4):592–622, 2021.

İ. Kara, G. Laporte, and T. Bektaş. A note on the lifted Miller–Tucker–Zemlin subtour elimination constraints for the capacitated vehicle routing problem. *European Journal of Operational Research*, 158(3): 793–795, 2004.

Ç. Koç, T. Bektaş, O. Jabali, and G. Laporte. The fleet size and mix location-routing problem with time windows: formulations and a heuristic algorithm. *European Journal of Operational Research*, 248(1): 33–53, 2016a.

Ç. Koç, T. Bektaş, O. Jabali, and G. Laporte. The impact of depot location, fleet composition and routing on emissions in city logistics. *Transportation Research Part B: Methodological*, 84:81–102, 2016b.

R. Lahyani, L.C. Coelho, and J. Renaud. Alternative formulations and improved bound for the multi-depot fleet size and mix vehicle routing problem. *OR Spectrum*, 40:125–157, 2018.

J. Li, R. Wang, T. Li, Z. Lu, and P.M. Pardalos. A literature review on the vehicle routing problem with multiple depots. *Transportation Research Part D: Transport and Environment*, 59:417–432, 2018.

C. Malandraki and M.S. Daskin. Time dependent vehicle routing problems: formulations, properties and heuristic algorithms. *Transportation Science*, 26(3):185–200, 1992.

S. Mancini. A real-life multi depot multi period vehicle routing problem with a heterogeneous fleet: formulation and adaptive large neighborhood search based matheuristic. *Transportation Research Part C: Emerging Technologies*, 70:100–112, 2016.

J.R. Montoya-Torres, J. L. Franco, S. N. Isaza, H. F. Jimenez, and N. Herazo-Padilla. A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, 79:115–129, 2015.

J. Renaud and F.F. Boctor. A sweep-based algorithm for the fleet size and mix vehicle routing problem. *European Journal of Operational Research*, 140(3):618–628, 2002.

N. Rincon-Garcia, B. Waterson, T. J. Cherrett, and F. Salazar-Arrieta. A metaheuristic for the time-dependent vehicle routing problem considering driving hours regulations – an application in city logistics. *Transportation Research Part A: Policy and Practice*, 137:429–446, 2020.

S. Ropke and D. Pisinger. An adaptative large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40:455–472, 2006.

S. Salhi and M. Sari. A multi-level composite heuristic for the multi-depot vehicle fleet mix problem. *European Journal of Operational Research*, 103(1):95–112, 1997.

S. Salhi, A. Imran, and N.A. Wassan. The multi-depot vehicle routing problem with heterogeneous vehicle fleet: formulation and a variable neighborhood search implementation. *Computers & Operations Research*, 52:315–325, 2014.

M.W.P. Savelsbergh and T. Van Woensel. 50th anniversary invited article—city logistics: Challenges and opportunities. *Transportation Science*, 50(2):579–590, 2016.

C.E. Schmidt, A.C.L. Silva, M. Darvish, and L.C. Coelho. The time-dependent location-routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 128:193–315, 2019.

A. Snoeck and M. Winkenbach. The value of physical distribution flexibility in serving dense and uncertain urban markets. *Transportation Research Part A: Policy and Practice*, 136:151–177, 2020.

M.G. Speranza. Trends in transportation and logistics. *European Journal of Operational Research*, 264(3): 830–836, 2018.

D. Taş, N. Dellaert, T. van Woensel, and T. De Kok. The time-dependent vehicle routing problem with soft time windows and stochastic travel times. *Transportation Research Part C: Emerging Technologies*, 48: 66–83, 2014.

J. E. Tyworth and A. Z. Zeng. Estimating the effects of carrier transit-time performance on logistics cost and service. *Transportation Research Part A: Policy and Practice*, 32(2):89–97, 1998.

T. Vidal, T.G. Crainic, M. Gendreau, N. Lahrichi, and W. Rei. A hybrid genetic algorithm for multi-depot and periodic vehicle routing problems. *Operations Research*, 60:611–624, 2013.

T. Vidal, T.G. Crainic, M. Gendreau, and C. Prins. Implicit depot assignments and rotations in vehicle routing heuristics. *European Journal of Operational Research*, 237(1):15–28, 2014.

Y. Xiao and A. Konak. The heterogeneous green vehicle routing and scheduling problem with time-varying traffic congestion. *Transportation Research Part E: Logistics and Transportation Review*, 88:146–166, 2016.

B. Yu, Z.Z. Yang, and J.X. Xie. A parallel improved ant colony optimization for multi depot vehicle routing problem. *Journal of the Operational Research Society*, 62(1):183–188, 2011.