



A POPMUSIC approach for the Multi-Depot Cumulative Capacitated Vehicle Routing Problem

Eduardo Lalla-Ruiz¹ · Stefan Voß^{2,3}

Received: 14 March 2018 / Accepted: 18 December 2018
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract

The Multi-Depot Cumulative Capacitated Vehicle Routing Problem is a variation of the recently proposed Capacitated Cumulative Vehicle Routing Problem, where several depots can be considered as starting points of routes. Its objective aims at minimizing the sum of arrival times at customers for providing service. Practical considerations imply to address the delivery of customers from multiple depots where the service quality level depends on the customer waiting time and the delivering vehicles may be able to depart from different points. Those scenarios require theoretical models to support the decision-making process as well as for measuring the quality of the solutions provided by approximate approaches. In the present work, we formalize this new problem variant by means of a mathematical formulation and propose a matheuristic approach (POPMUSIC) for solving it.

Keywords Multi-Depot Cumulative Vehicle Routing Problem · POPMUSIC · Matheuristic · Disaster logistics · Customer-oriented applications

1 Introduction

The Cumulative Capacitated Vehicle Routing Problem (CCVRP) is a quite recent routing problem that differs from the well-known Vehicle Routing Problem (VRP) [5] in that the required objective aims to minimize the total arrival time (i.e. the sum of arrival times) or the sum of the (estimated) time of finishing services at customers

✉ Eduardo Lalla-Ruiz
e.a.lalla@utwente.nl
Stefan Voß
stefan.voss@uni-hamburg.de

¹ Department of Industrial Engineering and Business Information Systems, University of Twente, Enschede, The Netherlands

² Institute of Information Systems, University of Hamburg, Hamburg, Germany

³ Escuela de Ingeniería Industrial, Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile

ensuring that every customer is visited and fully serviced. Considering this type of problem setting, a straightforward (and optimal) way of solving it is having each customer served by a separate vehicle. Nevertheless, that requires to have as many vehicles as customers, which usually does not happen in reality where the number of customer jobs normally exceeds the number of available vehicles. That consideration makes the problem interesting (and harder). Even further, when considering scenarios with multiple depots, this problem becomes more complex as the decision range of the problem is extended to also determine from which depot a customer has to be served.

The first work that formally investigates the CCVRP is from Nogueu et al. [16]. Their aim for proposing the CCVRP is to address problems where fast services characterizing the transport of supplies are required; practical applications include distribution, machine scheduling and even power-control and receiver optimization in wireless telecommunication systems (see [1]). The authors compare the contribution of the problem with already proposed routing problems and its similarities with the Traveling Repairmen Problem (TRP) also known as Minimum Latency Problem (MLP) and the Delivery Man Problem (DMP). For solving the CCVRP they propose a Memetic Algorithm (MA). The algorithm is assessed using instances whose size ranges from 50 to 199 nodes. Ribeiro and Laporte [18] propose an Adaptive Large Neighborhood Search (ALNS) for this problem. ALNS is able to improve the computational results obtained by means of the MA from [16]. Chen et al. [3] propose an Iterated Local Search (ILS) algorithm and provide a comparison with MA and ALNS. The comparison shows that they are able to improve the computational time to reach some best-known solutions; nevertheless, on average ALNS exhibits a better performance. Lysgaard and Wøhlk [13] investigate a branch-and-cut-and-price (BCP) algorithm. In their computational experiments, they consider the case of having an extra route. That is, they increase the instance feature concerning the number of available vehicles by one unit. The computational results show that the BCP is able to solve the CCVRP problem instances with up to 70 customers within reasonable computational times without considering an extra vehicle. Moreover, when an additional vehicle is added, the authors observe that the instances seem to be easier allowing to solve instances with more nodes for some of the instance sets they consider. A CCVRP with a modified objective function (minimizing the maximum arrival time) is proposed in [22].

In several logistic contexts, a large number of scenarios considers vehicles departing from different depots when starting their routes. This issue is addressed by the Multi-Depot Vehicle Routing Problem (MD-VRP); see Sumichras and Markham [21], Renaud et al. [17] and Montoya et al. [15]. This can be extended to the CCVRP for addressing scenarios where the service level to customers and timing is important from a customer viewpoint and where a set of vehicles can depart from different depots. That is, the vehicles can depart from one of the available depots and provide the requested deliveries or service to customers as soon as possible. Hence, including this additional decision level enables to also decide from which depot each vehicle has to depart and to which customer provide service. Thus, this extension allows to capture different real-world scenarios from distribution services centered in the customer satisfaction to humanitarian response or city services such as ambulance routing.

The importance of considering multiple depots can be contextualized by means of some related works in the literature. For instance, Talarico et al. [25] address the ambulance routing in disaster response where ambulances departing from different hospitals have to aid patient groups depending on the severity of the health problem. In that problem, the objective aims to minimize the weighted sum of the latest service completion time among red code patients and the latest service time among all green code patients. Salazar et al. [14] propose a routing problem allowing multiple uses of a single vehicle with the aim of minimizing the sum of the customer waiting times aiming to aid disaster-stricken communities. Similarly, Rivera et al. [19] propose the Multi-Trip Cumulative Capacitated Vehicle Routing Problem (MT-CCVRP) in which, inspired by disaster logistics, a single vehicle can perform successive trips for serving a set of sites or individuals with the objective of minimizing the sum of arrival times. It has to be noted that in that problem the authors consider a single vehicle which can perform more than one trip.

As discussed below, extending the CCVRP to consider multiple depots is hard to solve through the optimization model implemented in a general purpose solver. Thus, to improve the solution quality, in this work, we propose a Partial Optimization Metaheuristic under Special Intensification Conditions (POPMUSIC [23,24]) approach. In particular, the matheuristic version of this template proposed by Lalla-Ruiz and Voß [10], where sub-problems are solved to optimality by using exact approaches instead of approximate ones, is used for solving the MD-CCVRP. The underlying idea behind POPMUSIC is to address large problems by firstly decomposing them into parts and, subsequently, bundle those parts in order to build sub-problems. Those sub-problems are later solved using either an approximate or an exact approach. Similar to [10–12], we solve a reduced version of the problem instance and solve it by means of its optimization problem. The results in those works indicate that solving the sub-problems following that procedure reports a better performance than solving the complete problem at once.

Considering the aforementioned discussion, the goal of this paper is to propose the extension of the CCVRP for considering multiple depots by means of an integer programming formulation. This new problem is referred to as the Multi-Depot Cumulative Capacitated Vehicle Routing Problem (MD-CCVRP). Moreover, as indicated in [16], the CCVRP generalizes the TRP by adding capacities while considering a homogeneous fleet. Therefore, the MD-CCVRP allows addressing the TRP/MLP/DMP with multiple-depots and vehicles if the capacity constraints are relaxed and a maximum of one vehicle per depot is permitted. Thus, although not being the main goal of this paper, an adaptation of the developed mathematical model for tackling those problems is provided. Additionally, to enhance the proposed mathematical formulation, a lower bound on the minimum number of vehicles considering the capacities and requirements of the customers is proposed and assessed in the context of this problem. Finally, matheuristic POPMUSIC approaches are proposed and assessed to solve this problem.

The remainder of this paper is organized as follows. The proposed optimization problem, the MD-CCVRP, is introduced in Sect. 2. For solving this problem, a POPMUSIC matheuristic approach is developed and presented in Sect. 3. The computational experience carried out in this work is presented in Sect. 4. Finally, some conclusions are drawn in Sect. 5.

2 Mathematical formulation

The Multi-Depot Capacitated Vehicle Routing Problem (MD-CCVRP) can be defined as follows. Let $G = (V, A)$ be a directed graph, where $V = N \cup D$ is the vertex or node set that contains all customer and depot nodes, and A is the arc set. Vertex set $D = \{1, 2, \dots, m\}$ represents the set of m uncapacitated depots, whereas vertex set $N = \{1, 2, \dots, m+n\}$ represents the customers to be served. A travelling cost, $c_{ij} > 0$, is defined for each arc between each pair of vertices (i, j) , $i, j \in V, i \neq j$. The traveling cost can represent, according to the application environment, the distance, time, fuel consumption, etc., between each pair of vertices. It is assumed that travel times are symmetric and satisfy the triangle inequality. Moreover, each customer $i \in N$ has demand q_i and, each depot $d \in D$ stores and supplies enough goods to serve all the customers. In the MD-CCVRP, the number of vehicles is limited, whereas R represents the set of identical vehicles with the same positive capacity, denoted as Q . The MD-CCVRP aims to minimize the total arrival times at customers while satisfying the following conditions: (a) each customer must be visited on exactly one route, (b) each customer has to be fully served when visited, (c) each vehicle departs from one of the available depots and finishes at one of them, (d) the total demand of the customers on any route does not exceed the capacity of the vehicles, i.e., Q .

In the following, we introduce a mathematical formulation for solving the MD-CCVRP that extends CCVRP [16] to consider multiple depots. The decision variables are the following:

x_{ij}^k 1 if vehicle k travels directly from node $i \in V$ to node $j \in V$, 0 otherwise.

t_i^k the time at which vehicle k arrives at node $i \in V$. If a customer is not serviced by a given vehicle, then this value is 0.

The mathematical model we propose is defined as follows:

$$(MD-CCVRP) \text{ minimize } \sum_{k \in R} \sum_{i \in N} t_i^k \quad (1)$$

subject to:

$$\sum_{i \in V} \sum_{k \in R} x_{ij}^k = 1, \quad \forall j \in N \quad (2)$$

$$\sum_{i \in V} x_{ij}^k - \sum_{i \in V} x_{ji}^k = 0, \quad \forall j \in N, \forall k \in R \quad (3)$$

$$\sum_{i \in D} \sum_{j \in V} x_{ij}^k = 1, \quad \forall k \in R \quad (4)$$

$$\sum_{i \in D} \sum_{j \in V} x_{ji}^k = 1, \quad \forall k \in R \quad (5)$$

$$\sum_{i \in V} \sum_{j \in V} q_j x_{ij}^k \leq Q, \quad \forall k \in R \quad (6)$$

$$t_i^k + c_{ij} - t_j^k \leq (1 - x_{ij}^k)M, \quad \forall i \in V, \forall j \in N, k \in R \quad (7)$$

$$t_i^k \geq 0, \quad \forall i \in V, k \in R \tag{8}$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall i, j \in V, k \in R \tag{9}$$

The objective function (1) minimizes the sum of arrival times at the customers. Constraints (2) guarantee that each customer is visited exactly once. Constraints (3) impose the degree balance of each node, including both customers and depots. Constraints (4) and (5) establish that each vehicle starts and finishes at a depot $i \in D$. Constraints (6) represent the capacity restriction of each vehicle. Constraints (7) ensure that if customer j is served after customer i by vehicle k , then t_j^k must be greater than or equal to t_i^k plus the travel time c_{ij} , and M is a sufficiently large positive constant. Note that (7) also guarantees that subtours are avoided. Finally, Constraints (8) and (9) are the integrality and non-negativity constraints for the different kinds of variables.

Furthermore, it should be mentioned that MD-CCVRP is a generalization of the CCVRP for considering multiple depots. The special case of having one depot ($|D| = 1$) is the standard CCVRP. Therefore, since the MD-CCVRP can be reduced to the CCVRP, this problem is \mathcal{NP} -hard.

Property 1 *Optimal MD-CCVRP solutions use exactly $\min(R, n)$ vehicles.* This is derived from the properties of the CCVRP (see [16]). The objective function value of the solution decreases as R increases. Moreover, the solution having $|R|$ larger than n is a set of back and forth routes between depots and customers, with $R - n$ vehicles unused. Note that this applies in the case where the triangle inequality is maintained as well as having the initial position of the vehicles not fixed. In this case, a vehicle visiting a customer $j \in N$ will depart from that depot yielding the minimum travelling cost to the customer, i.e., $\min(c_{ij})_{i \in D}$. The objective function value of that lower bound is equal to $\sum_{j \in N} c_{\min(c_{ij})_{i \in D} j}$.

Property 2 *A CCVRP solution using exactly $\min(R, n)$ vehicles for a scenario with fixed starting depot is a valid upper-bound for MD-CCVRP.* Since the MD-CCVRP is a generalization where several depots are added, the solution of the CCVRP is valid and constitutes a valid upper bound.

2.1 Illustrative example

In order to improve the understanding of the MD-CCVRP, Fig. 1 shows an example of a solution of the MD-CCVRP composed of 13 customers $\{c_1, c_2, \dots, c_{13}\}$ and 3 depots $\{\text{depot}_1, \text{depot}_2, \text{and } \text{depot}_3\}$. The dashed lines represent the routes. In this solution a vehicle departs from each depot. To ease the following of this example, the identifier of each vehicle matches with its departing depot and the solution satisfies the capacity constraints. Thus, the route followed by vehicle 1 is $(c_4, c_5, c_6, c_7, c_8, c_9)$, by vehicle 2 is (c_{12}, c_{11}, c_{10}) , and by vehicle 3 is (c_1, c_2, c_3, c_{13}) .

In the example vehicle 1 is departing from depot₁ and visits a total of six customers along its route; the time that the first customer of this route, c_4 , waits for its service is 14. Customer c_5 is served after 24 time units. The objective function value of this solution is 290, composed of the following values: $(t_{c_1}^3 = 3) + (t_{c_2}^3 = 15) + (t_{c_3}^3 = 17) + (t_{c_4}^1 = 14) + (t_{c_5}^1 = 24) + (t_{c_6}^1 = 32) + (t_{c_7}^1 = 38) + (t_{c_8}^1 = 41) + (t_{c_9}^1 = 43) + (t_{c_{10}}^2 = 15)$

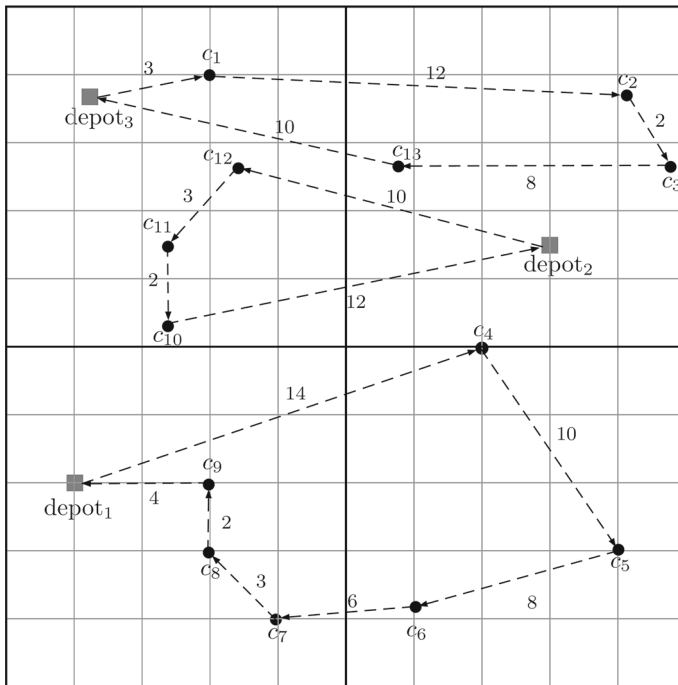


Fig. 1 Example of a solution for the MD-CCVRP. This example is composed of 13 customers and 3 depots

+ ($t_{c_{11}}^2 = 13$) + ($t_{c_{12}}^2 = 10$) + ($t_{c_{13}}^3 = 25$). It is worth to mention that this problem can be categorized as an open problem since the last arc is not considered in the objective function. The above highlights the focus given by this problem to customer oriented applications as mentioned in Sect. 1. For the same solution drawn in Fig. 1, the objective function value of the MD-VRP considering the traveled arcs is 109.

2.2 A multi-depot multi-vehicle generalization for the traveling repairman problem

As indicated in [16], the CCVRP generalizes the Traveling Repairman Problem (TRP [20]), the Minimum Latency Problem (MLP [2]), and the Delivery Man Problem (MDP [6]), which at the same time are special cases of the time-dependent traveling salesman problem [8] by adding capacities and considering a homogeneous fleet. Therefore, the MD-CCVRP allows addressing those problems with multiple depots and vehicles if the capacity constraints are relaxed and a maximum of one vehicle per depot is permitted. In order to adapt the previous mathematical model, Constraints (6) have to be removed for avoiding the capacity restrictions and Constraints (4) and (5) have to be redefined as follows.

$$\sum_{j \in V} x_{ij}^i = 1, \forall i \in D \quad (10)$$

$$\sum_{j \in V} x_{ji}^i = 1, \forall i \in D \tag{11}$$

Constraints (10) and (11) establish that one vehicle departs and returns to each depot. It should be noted that this problem requires that $|R| = |D|$.

Furthermore, it should be mentioned that the formulation for the TRP/MLP/MDP can be improved by adapting and using the strengthening constraints already proposed for those problems.

Lemma 1 *The TRP/MDP/MLP with multiple depots is a special case of the MD-CCVRP when the capacities are not considered and one vehicle is only able to depart from each depot.*

Proof Consider the formulation of MD-CCVRP. Define $q_j = 0, \forall j \in V, |R| = |D|$, and replace Constraints (4) and (5) by Constraints (10) and (11). The resulting formulation leads to the corresponding of TRP/MDP/MLP with multiple depots. \square

Lemma 2 *The TRP/MDP/MLP is a special of the MD-CCVRP when one depot is considered.*

Proof Consider Lemma 1 while setting $|D| = |R| = 1$. \square

2.3 Lower bound on the number of vehicles

A constraint is proposed to set a lower bound on the number of vehicles required to serve customers.

$$\sum_{i \in D} \sum_{j \in N} \sum_{k \in R} x_{ij}^k \geq \frac{\sum_{i \in N} q_i}{Q} \tag{12}$$

Proposition 1 *The following constraint is valid for the MD-CCVRP:*

$$\sum_{i \in D} \sum_{j \in N} \sum_{k \in R} x_{ij}^k \geq \frac{\sum_{i \in N} q_i}{Q}$$

Proof By contradiction. Let us consider a feasible solution where Constraint (12) is not satisfied. Therefore,

$$\sum_{i \in D} \sum_{j \in N} \sum_{k \in R} x_{ij}^k < \frac{\sum_{i \in N} q_i}{Q}.$$

This means that the amount of goods delivered by the vehicles is lower than the requirement of the customers. That is,

$$Q \sum_{i \in D} \sum_{j \in N} \sum_{k \in R} x_{ij}^k < \sum_{i \in N} q_i,$$

which leads to not satisfying Constraints (2) and (6), and henceforth the solution is not feasible. \square

3 A POPMUSIC approach for the MD-CCVRP

The POPMUSIC framework was firstly proposed by Taillard and Voß [23]. Its basic idea is to split a solution of the problem at hand, S , into t parts $part_1, part_2, \dots, part_t$ and joining some of them to build a sub-problem, T . To form the sub-problem, one of the t parts is selected, $part_{seed}$, termed as *seed-part*. With that, a number r of the closest parts to $part_{seed}$ are aggregated to the latter to form the sub-problem T . In order to determine the closeness of the parts, a distance measure among them is defined. Once a sub-problem is constructed, it is solved by using an approximate or an exact solution approach (leading to the matheuristic version, see [10]). If parts and sub-problems are defined in an appropriate way, each improvement of a sub-problem corresponds to an improvement of the whole solution S . This process of parts selection is repeated until the solution does not contain a sub-problem that can be improved.

Algorithm 1: POPMUSIC framework

```

1 Generate an initial solution  $S$ 
2 Decompose  $S$  into  $t$  parts,  $H = \{part_1, \dots, part_t\}$ 
3 Set  $O = \emptyset$ 
4 while ( $O \neq \{part_1, \dots, part_t\}$ ) do
5   | Select a seed part  $part_{seed} \notin O$ 
6   | Build a sub-problem  $T$  composed of the  $r$  parts of  $S$  which are the closest to  $part_{seed}$ 
7   | Optimize  $T$  by using an approximate or exact solution approach
8   | if ( $T$  has been improved) then
9   |   | Update solution  $S$ 
10  |   |  $O \leftarrow \emptyset$ 
11  | end
12  | else
13  |   | Include  $part_{seed}$  in  $O$ 
14  | end
15 end
16 return improved solution  $S$ 

```

Algorithm 1 depicts the POPMUSIC framework. An initial solution is generated, S (line 1). Once it is generated, the next step is to divide the solution into t parts (line 2). Then, a seed part, $part_{seed}$, is selected (line 5). A sub-problem, T , is constructed by considering its r nearest parts according to a relatedness function (line 6). In this regard, the unique parameter of this framework, r , is used for delimiting the size of the sub-problems. The sub-problem T is then solved by an approximate or exact procedure (line 7). In this framework, the set of parts O corresponds precisely to seed parts that have been used to define sub-problems that have been unsuccessfully optimized. Once O contains all the parts of the complete solution (line 4), the process stops as all sub-problems have been examined without success.

3.1 Solution structure

In the context of POPMUSIC, we first define a solution structure for the MD-CCVRP. It is defined as a sequence of routes starting at each depot. The visiting order of each customer is determined by its position in the sequence. That is, for example, if we consider the solution of Fig. 1 for three vehicles and thirteen customers, the solution is as follows:

$$S = \{d_1, c_4, c_5, c_6, c_7, c_8, c_9, d_2, c_{12}, c_{11}, c_{10}, d_3, c_1, c_2, c_3, c_{13}\}.$$

In the solution, the parts are delimited by the depots, so in this case, the first part is defined as $part_1 = \{d_1, c_4, c_5, c_6, c_7, c_8, c_9\}$. Moreover, as indicated above, the POPMUSIC framework uses the information from the solution and problem instance to delimit the parts that will be subsequently used to build the sub-problems. In the MD-CCVRP, the number of parts, t , is a parameter depending on the instance. Thus, a solution is divided into $part_1, part_2, \dots, part_t$ parts, where t is equal to the number of depots (m), i.e. $t = m$. At this point, it should be noted that some parts (i.e. vehicle routes) can be empty, meaning that the vehicle does not visit any customer. In that case, the corresponding part starts and finishes at the same depot without any customer in-between.

3.2 Initial solution generation procedure

In order to properly start the POPMUSIC search process, a starting solution is required. In this sense, different ways can be considered, such as running the optimization model until an incumbent solution is obtained or using heuristics for its generation. In this work we propose a greedy clustering algorithm that takes into consideration the distance of each customer to each depot. By means of this method, we are able to generate a fast initial and feasible solution.

Algorithm 2 depicts the initial solution generation procedure. Firstly, we determine the nearest depot to each customer (lines 1–3). Afterward, as shown in lines 5–19, for each depot and customer, we route a vehicle until its capacity does not enable to serve a customer. In such case, a new vehicle is assigned to that customer (see lines 12–16).

3.3 Sub-problem generation strategy

In order to form a sub-problem, T , a part, termed as $part_{seed}$, is selected and its r neighbor parts are joint together. Thus, a sub-problem contains all the customers and depots associated with any of its parts. That is, those customers served in the joined parts are the ones considered in the sub-problem. The information related to the depots and customers remains the same as in the complete problem while the information regarding the parts not belonging to the sub-problem is not considered for solving the sub-problem.

For building a sub-problem, the way the parts are bundled can be conducted by using different strategies. In the current paper, two strategies are proposed:

- *Lexicographic* The sub-problem is constructed according to the indexes of the parts. That is, for a given part $part_o$ and $r = 1$ all the customers belonging to that

Algorithm 2: Initial solution generation procedure

```

1 for (each  $i \in N$ ) do
2   |  $\min[i] = \operatorname{argmin}(c_{ij})_{j \in D}$ 
3 end
4  $\text{vehicle}_{i,d} = 1$ 
5 for (each  $j \in D$ ) do
6   | for (each  $i \in N$ ) do
7     | if ( $\min[i] = j$ ) then
8       | if ( $Q_{\text{vehicle}_{i,d}} \geq d_i$ ) then
9         |  $Q_{\text{vehicle}_{i,d}} = Q_{\text{vehicle}_{i,d}} - q_i$ 
10        | customer  $i$  added to  $\text{route}_{\text{vehicle}_{i,d}}$ 
11        | end
12        | else
13          |  $\text{vehicle}_{i,d}++$ 
14          |  $Q_{\text{vehicle}_{i,d}} = Q_{\text{vehicle}_{i,d}} - q_i$ 
15          | customer  $i$  added to  $\text{route}_{\text{vehicle}_{i,d}}$ 
16          | end
17        | end
18      | end
19 end

```

part are grouped to those belonging to the next increasing index neighbor part, i.e. part_{o+1} . If $r = 2$, the part part_{o+2} is also included in the sub-problem. Thus, this can be generalized to $\cup_{i=0}^{o+r} \text{part}_i$. For the example shown in Fig. 1 with the solution divided into 3 parts, if we consider $r = 1$, we can have the following sub-problems, $T = \{\text{part}_1, \text{part}_2\}$, $T = \{\text{part}_2, \text{part}_3\}$, and $T = \{\text{part}_3, \text{part}_1\}$. At this point, it has to be noted that if $o + r$ is larger than T , the consideration of the parts also includes starting from the first part.

- *Distance* This strategy takes into account the minimum distance between the depots. That is, the distance between two parts, for example, part_i and part_j , is given by the distance between the two depots as $\bar{d}_{ij} = c_{ij}$. Thus, the construction of the sub-problem is then performed in a greedy way. Once the seed $\text{part}_o \in H$ has been selected, those r parts with the minimum distance to that part are selected to build the sub-problem. For example, for $r = 1$ and being the seed part part_o , the part chosen to form the sub-problem is the one given by $\text{part}_j = \operatorname{argmin}(\bar{d}_{oj})_{\text{part}_j \in H, j \neq o}$.

Returning to the example shown in Fig. 1, and let us suppose our seed part $\text{part}_{\text{seed}}$ is the third part associated to the third depot (i.e. d_3), and let us suppose a value r equal to 1. Then, the sub-problem T , using the *Lexicographic* strategy is constructed with part_3 and part_1 . Notice that part_1 is bundled as the next part of part_3 . At first glance, it would be part_4 , but since we only have three parts we restart and use part_1 instead. On the other hand, by means of the *Distance* strategy, the resulting sub-problem is composed of part_2 and part_3 .

4 Computational results

This section is devoted to present the computational experiments carried out for assessing the performance of the mathematical model proposed in this work (see Sect. 2), the provided lower bound on the number of vehicles, and our proposed POPMUSIC approach. The computational experiments related to our mathematical model were conducted on a computer equipped with an Intel Dual Core 3.5 GHz and 16 GB of RAM. The model was implemented in a general-purpose solver, CPLEX version 12.6, with a maximum computational time of 7200 s and the number of threads restricted to 1. The problem instances used in this work are those proposed by Cordeau et al. [4] for the MD-VRP. Additionally, a new set of small and medium size problem instances¹ generated based on [4] is proposed. Namely, for generating an instance of n customers and m depots, we randomly extract from a larger instance of [4] $n + m$ node locations as well as n customer demands that are later associated at random to the n customers of the new instance. The capacity of the vehicles remains the same as the original instance of Cordeau et al. [4]. The rationale behind this is to provide problem instances where CPLEX can provide a feasible solution within the given time limit to better assess and tune our proposed algorithm. By means of this new set, we aim to provide the possibility of comparing future approximate and exact approaches for this problem. The new instances identifier format is as follows: $n \times m \times |R|$ -id. This way, instance $10 \times 4 \times 5$ -1 is the first instance considering 10 customers, 4 depots, and 5 vehicles. Finally, for the instances of Cordeau et al. [4] we fix the number of vehicles to 35.

4.1 Assessment of lower bound contribution on the number of vehicles

This subsection compares the performance of both formulations, with and without a lower bound on the number of vehicles, i.e. MD-CCVRP w/o (12) and MD-CCVRP w/(12), respectively. In doing so, first, we evaluate their performance on small and medium size instances as shown in Table 1 to investigate the performance and contribution of the lower bound when the size of the problems is not so large. Table 2 reports the results of both optimization models for large instances. In the tables, we report the calculated upper and lower bounds, UB and LB, respectively, the relative error [gap (%)], and the time measured in seconds [t (s)].

As can be checked in Table 1, while the performance of the model without the lower bound exhibits a better performance in terms of best-known solutions, the quality of the lower bounds is better when considering a lower bound on the number of vehicles. Moreover, in those cases where an optimal solution is reached, the time performance of both optimization models remains similar. Although MD-CCVRP without (12) provides better lower bounds, in some cases (i.e. 25×4 -3, 50×4 -3) MD-CCVRP with (12) shows a better performance. The weak influence of the lower bound on the number of vehicles finds its rationale in the fact that such lower bound considers the requested quantity and the capacity of the vehicle but the objective seeks to reduce the waiting time. Moreover, this bound may be redundant for the general purpose solver, since it is a preprocessing requisite for starting to solve a given problem. Note

¹ The new instances are online available at: <https://github.com/elalla/MDCCVRP>.

Table 1 Computational results of the optimization models for small and medium size instances (best UB values are given in bold face)

Instance	MD-CCVRP w/o (12)			MD-CCVRP w/(12)				
	UB	LB	Gap (%)	t (s)	UB	LB	Gap (%)	t (s)
10 × 4 × 5-1	545.69	545.69	0.00	2.20	545.69	545.69	0.00	2.20
10 × 4 × 5-2	832.69	832.69	0.00	2.98	832.69	832.69	0.00	3.00
10 × 4 × 5-3	832.78	832.78	0.00	7.83	832.78	832.78	0.00	7.86
25 × 4 × 10-1	2135.52	10.00	99.53	7200	2195.40	30.00	98.63	7200
25 × 4 × 10-2	1854.19	10.00	99.46	7200	1993.38	40.00	97.99	7200
25 × 4 × 10-3	1863.77	20.11	98.92	7200	1786.95	40.00	97.76	7200
50 × 4 × 20-1	6589.67	0.00	100.00	7200	8093.72	123.73	98.47	7200
50 × 4 × 20-2	5098.24	0.00	100.00	7200	9582.61	51.44	99.46	7200
50 × 4 × 20-3	6522.25	0.00	100.00	7200	6130.96	56.50	99.08	7200
50 × 6 × 20-1	3905.96	0.00	100.00	7200	5657.76	40.00	99.29	7200
50 × 6 × 20-2	4005.96	0.00	100.00	7200	5483.76	64.14	98.83	7200
50 × 6 × 20-3	5100.17	0.00	100.00	7200	6707.37	44.54	99.34	7200
100 × 4 × 25-1	53,332.63	0.00	100.00	7200	105,440.07	77.79	99.93	7200
100 × 4 × 25-2	90,525.46	0.00	100.00	7200	90,525.46	98.50	99.89	7200
100 × 4 × 25-3	72,261.54	0.00	100.00	7200	78,622.25	107.30	99.86	7200
100 × 6 × 25-1	56,831.23	0.00	100.00	7200	94,304.49	65.72	99.93	7200
100 × 6 × 25-2	73,249.00	0.00	100.00	7200	86,076.03	75.11	99.91	7200
100 × 6 × 25-3	95,593.84	0.00	100.00	7200	104,913.93	60.22	99.94	7200
Avg.	26,726.70	125.07	83.22	6000.72	33,873.63	177.01	82.68	6000.73

Table 2 Computational results of the optimization models for the instances proposed in [4] considering $|R|=35$ vehicles (best UB values are given in bold face)

Instance	MD-CCVRP w/o (12)				MD-CCVRP w/(12)			
	UB	LB	Gap (%)	t (s)	UB	LB	Gap (%)	t (s)
Cordeau_p01	916.93	0.00	100.00	7200	1573.59	48.24	96.93	7200
Cordeau_p02	5377.49	0.00	100.00	7200	1814.32	11.50	99.37	7200
Cordeau_p03	7247.32	0.00	100.00	7200	8952.15	27.37	99.69	7200
Cordeau_p04	10,252.10	0.00	100.00	7200	10,602.68	95.35	99.08	7200
Cordeau_p05	16,626.06	0.00	100.00	7200	16,785.58	26.49	99.84	7200
Cordeau_p06	8350.04	0.00	100.00	7200	11,187.35	58.65	99.48	7200
Cordeau_p07	9601.76	0.00	100.00	7200	10,891.65	57.49	99.47	7200
Cordeau_p08	–	–	–	7200	–	–	–	7200
Cordeau_p09	–	–	–	7200	–	–	–	7200
Cordeau_p10	–	–	–	7200	–	–	–	7200
Cordeau_p11	–	–	–	7200	–	–	–	7200
Cordeau_p12	23,670.55	0.00	100.00	7200	31,989.96	65.00	99.80	7200
Cordeau_p15	–	–	–	7200	–	–	–	7200
Cordeau_p18	–	–	–	7200	–	–	–	7200
Cordeau_pr01	8088.03	0.00	100.00	7200	5339.00	11.93	99.78	7200
Cordeau_pr02	46,714.67	0.00	100.00	7200	47,051.32	20.13	99.96	7200
Cordeau_pr03	89,167.09	0.00	100.00	7200	89,167.09	36.47	99.96	7200
Cordeau_pr04	–	–	–	7200	–	–	–	7200
Cordeau_pr05	–	–	–	7200	–	–	–	7200
Cordeau_pr06	–	–	–	7200	–	–	–	7200
Cordeau_pr07	22,940.81	0.00	100.00	7200	38,705.41	17.45	99.96	7200
Cordeau_pr08	74,941.48	0.00	100.00	7200	74,941.48	27.95	99.96	7200
Cordeau_pr09	–	–	–	7200	–	–	–	7200
Cordeau_pr10	–	–	–	7200	–	–	–	7200
Avg.	24,914.95	0.00	100.00	7200	26,846.28	38.77	99.48	7200

that this is related to the influence of erraticism (see [7]) and could be exploited by a more sophisticated framework as that proposed in [9]. Moreover, for the large size instances reported in Table 2, similar insights can be extracted. Clearly, MD-CCVRP without (12) provides better upper bounds while MD-CCVRP with (12) does the same for the lower bounds. In this case, MD-CCVRP with (12) complements the results of the other formulation by providing better results for the instances Cordeau_p02 and Cordeau_pr01.

As discussed in Sect. 2, the number of available vehicles is a feature of the problem instance that has an influence on the temporal performance of the models (see [13]). Due to this, we have investigated the influence of such feature of the problem for three problem instances where the optimal solution can be reached within the given time limit (i.e. $10 \times 4-1$, $10 \times 4-2$, and $10 \times 4-3$). That is, we have incrementally varied that problem instance feature from 2 to the number of nodes of those instances, i.e.

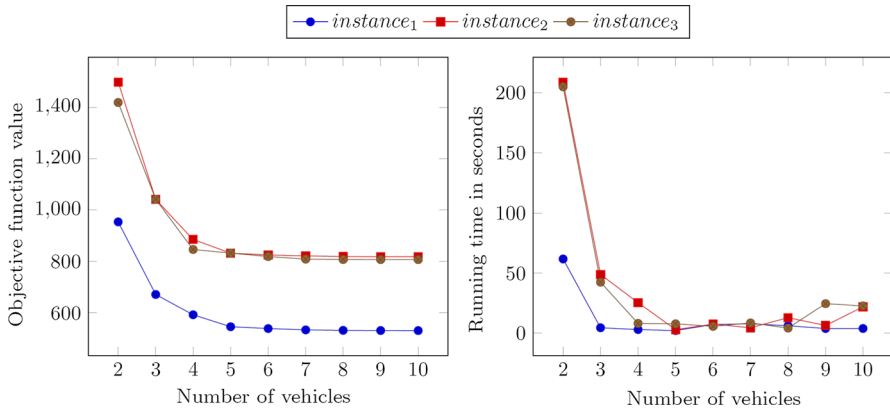


Fig. 2 Performance of the optimization model for different number of vehicles

$|R| = 2, 3, \dots, |N|$. In this regard, it has to be noticed, considering the objective function value and Property 2 (see Sect. 2), that all available vehicles will always be used when their number is less than the number of customers. Figure 2 shows the variation of the optimal solutions and solving time when the number of vehicles varies. As expected, the lower the number of vehicles the harder to solve the problem while in the range of 5 to 10 vehicles the temporal behavior keeps a similar trend.

4.2 Parameter setting of the POPMUSIC approach

In order to determine the best parameters for our POPMUSIC approaches, we have conducted a proper selection of parameter values by executing them on a representative set of problem instances. The parameter values assessed in this subsection are presented in Table 3. The corresponding table thus reports the time limit t to solve in an exact way each sub-problem, the different values of r for building the sub-problems, the strategy to build the sub-problems (*closeness*), and if the used optimization model for solving the sub-problem considers the lower bound on the number of vehicles (*mvc*). We have run a Friedman parameter setting test considering the average values. The test indicates that the parameter tuple $(300, 2, \text{lex}, 0)$ is the best performing one. Moreover, with the aim of illustrating the behavior of POPMUSIC for the different parameters, we provide in Fig. 3 the trade-off between the average of best objective function value and computational time for each parameter setting. As can be observed in the plots, both lexicographic approaches behave better than those using the distance closeness measure.

4.3 POPMUSIC computational results

POPMUSIC ruling parameters have been determined in the previous subsection. We assess the performance of POPMUSIC on all the problem instances in terms of objective function value and computational time of the selected POPMUSIC approach

Table 3 POPMUSIC Parameter values considered

Parameter	Values
t	$\in \{50, 100, 200, 300\}$
r	$\in \{2, 3\}$
$closeness$	$\in \{dist, lex\}$
mvc	$\in \{0, 1\}$

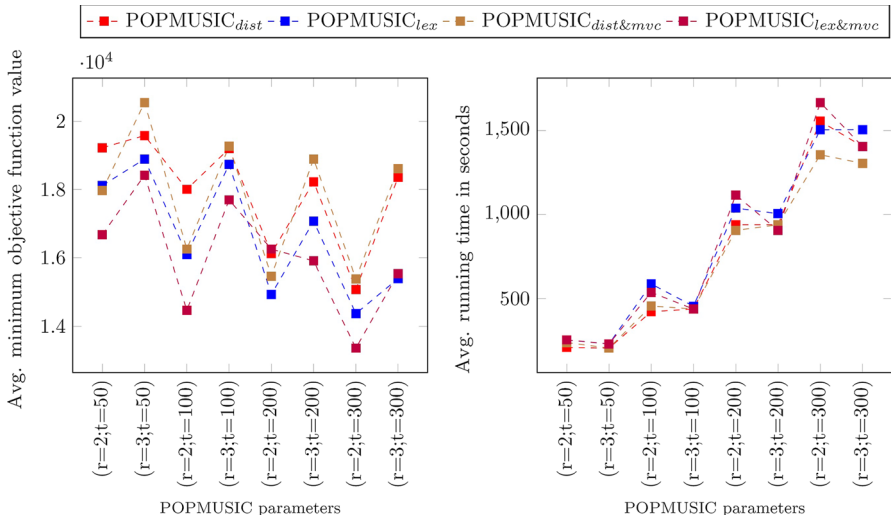


Fig. 3 Overall performance in terms of objective function value and time of POPMUSIC for different parameter values

embedding both models, namely, with and without a lower bound on the number of vehicles. First, we analyze the performance of POPMUSIC on the small and medium size instances in order to determine the quality of our decomposition strategy contemplated in POPMUSIC for such sizes. Those results are reported in Table 4. Similarly, in Table 5, we provide the results for the same experiment on large size instances. In the tables, we report the best-known solution provided by the optimization models implemented in CPLEX (BKS_{CPLEX}) which includes the information concerning the upper bound (UB) and the relative error [gap (%)] for each case. It has to be noted that for these instances CPLEX reaches the time limit in all cases. Moreover, for the POPMUSIC approach with and without the lower bound on the number of vehicles, i.e. POPMUSIC_{w/(12)} and POPMUSIC_{w/o(12)}, we provide the best (Min), average (Avg), and worst (Max) objective function value provided per each problem instance along ten replications. In addition, the gap with respect to the UB provided by CPLEX is reported in column $gap_2(\%)$.

In the results reported in Table 4, it can be seen that in those cases where the optimal solution is known, both POPMUSIC approaches are able to reach it. As the size of the problem instances increases, CPLEX provides better solutions than POPMUSIC at the expense of a higher computational time. This points out that, up to a given problem

Table 4 Assessment of POPMUSIC performance for small and medium size instances (best UB values are given in bold face)

Instance	BKSCPLEX				POPMUSIC _{w/o} (12)				POPMUSIC _w (12)					
	UB	Gap (%)	t (s)		Min	Max	Avg	Gap ₂ (%)	t (s)	Min	Max	Avg	Gap ₂ (%)	t (s)
10 × 4 × 5-1		0.00	2.20		545.69	583.62	556.64	0.00	3.41	545.69	545.69	545.69	0.00	3.41
10 × 4 × 5-2		0.00	2.98		832.69	878.77	841.90	0.00	2.23	832.69	862.31	838.61	0.00	2.23
10 × 4 × 5-3		0.00	7.83		832.78	836.91	834.74	0.00	3.72	832.78	836.91	836.09	0.00	3.48
25 × 4 × 10-1		99.53	7200		2180.54	2550.93	2420.91	2.06	1703	2260.74	2537.46	2398.34	5.54	2101
25 × 4 × 10-2		99.46	7200		2054.77	2309.05	2188.40	9.76	2401	2241.22	2523.04	2373.87	17.27	1501
25 × 4 × 10-3		97.76	7200		1953.68	2261.42	2075.25	8.53	2401	2134.46	2464.10	2247.30	16.28	2102
50 × 4 × 20-1		0.00	7200		15,791.78	26,400.03	21,666.61	58.27	1201	11,145.20	26,400.03	20,832.29	40.87	1201
50 × 4 × 20-2		0.00	7200		18,312.43	18,556.43	18,507.63	72.16	1197	18,556.43	18,556.43	18,556.43	72.53	1201
50 × 4 × 20-3		99.08	7200		15,190.85	20,363.59	19,329.04	59.64	1201	17,281.14	20,363.59	18,331.51	64.52	1201
50 × 6 × 25-1		100.00	7200		5202.64	16,260.08	10,991.71	24.92	2402	5752.04	9649.41	6987.33	32.09	3604
50 × 6 × 25-2		100.00	7200		9780.76	10,632.35	10,402.21	59.04	2102	8367.53	13,521.89	10,959.34	52.12	2102
50 × 6 × 25-3		100.00	7200		8485.06	11,080.64	9304.31	39.89	2103	8239.45	9263.11	8559.20	38.10	5712
100 × 4 × 25-1		53,332.63	7200		45,659.96	45,659.96	45,659.96	-16.80	1205	45,659.96	45,659.96	45,659.96	-16.80	1205
100 × 4 × 25-2		90,525.46	7200		46,682.08	46,682.08	46,682.08	-93.92	1205	46,682.08	46,682.08	46,682.08	-93.92	1205
100 × 4 × 25-3		72,261.54	7200		38,390.17	38,390.17	38,390.17	-88.23	1205	38,390.17	38,390.17	38,390.17	-88.23	1205
100 × 6 × 25-1		56,831.23	7200		23,005.64	29,465.80	27,017.76	-147.03	1807	28,535.58	29,465.80	29,279.76	-99.16	1807
100 × 6 × 25-2		73,249.00	7200		32,661.61	40,070.40	38,297.44	-124.27	1807	34,318.08	40,070.40	38,670.84	-113.44	1807
100 × 6 × 25-3		95,593.84	7200		29,813.45	40,453.45	36,719.67	-220.64	1808	37,230.88	40,453.45	39,562.78	-156.76	1807
Avg:		26,700.69	6000.72		16,520.92	19,635.32	18,438.13	-19.81	1430.90	17,167.01	19,346.99	18,428.42	-12.72	1653.79

Table 5 Assessment of POPMUSIC performance for the problem instances proposed by [4] (best UB values are given in bold face)

Instance	BKS_CPLEX				POPMUSIC _{w/ol(12)}				POPMUSIC _{w/(12)}					
	UB	Gap (%)	t (s)		Min	Avg	Max	Gap2 (%)	t (s)	Min	Avg	Max	Gap2 (%)	t (s)
Cordeau_p01	916.93	100.00	7200		1680.93	1952.91	2278.94	45.45	1502.40	1469.90	1852.09	2294.56	37.62	1201.85
Cordeau_p02	1814.32	99.37	7200		2102.48	2502.41	2744.59	13.71	1502.70	2084.70	2625.83	2872.52	12.97	2103.17
Cordeau_p03	7247.32	100.00	7200		3988.72	3988.72	3988.72	-81.70	1504.54	3820.91	3955.16	3988.72	-89.68	1504.80
Cordeau_p04	10,252.10	100.00	7200		6908.25	6908.25	6908.25	-48.40	604.74	6908.25	6908.25	6908.25	-48.40	604.90
Cordeau_p05	16,626.06	100.00	7200		15,394.49	15,394.49	15,394.49	-8.00	606.23	15,394.49	15,394.49	15,394.49	-8.00	604.81
Cordeau_p06	8,350.04	100.00	7200		6201.52	6201.52	6201.52	-34.64	906.41	6201.52	6201.52	6201.52	-34.65	906.36
Cordeau_p07	9,601.76	100.00	7200		5882.11	5882.11	5882.11	-63.24	1208.18	5882.11	5882.11	5882.11	-63.24	1207.36
Cordeau_p08	-	-	7200		105,289.18	105,289.18	105,289.18	-	846.57	105,289.18	105,289.18	105,289.18	-	944.82
Cordeau_p09	-	-	7200		78,484.63	78,484.63	78,484.63	-	946.96	78,484.63	78,484.63	78,484.63	-	953.47
Cordeau_p10	-	-	7200		74,570.66	74,570.66	74,570.66	-	1256.04	74,570.66	74,570.66	74,570.66	-	1260.00
Cordeau_p11	-	-	7200		64,744.95	64,744.95	64,744.95	-	1567.30	64,744.95	64,744.95	64,744.95	-	1566.57
Cordeau_p12	23,670.55	100.00	7200		27,202.35	27,502.91	28,164.72	12.98	1243.83	27,379.94	27,579.06	28,164.72	13.55	1807.42
Cordeau_p15	-	-	7200		56,329.44	56,329.44	56,329.44	-	1222.19	56,249.30	56,309.40	56,329.44	-	2435.71
Cordeau_p18	-	-	7200		84,494.16	84,494.16	84,494.16	-	1869.93	84,494.16	84,494.16	84,494.16	-	1867.21
Cordeau_pr01	5339.00	99.78	7200		4120.95	5380.99	6834.80	-29.56	2404.22	5105.67	6719.07	8637.19	-4.57	1501.83
Cordeau_pr02	46,714.67	100.00	7200		26,102.43	26,102.43	26,102.43	-78.97	1207.31	21,969.60	24,270.99	26,102.43	-112.63	1206.95
Cordeau_pr03	89,167.09	99.96	7200		40,930.50	40,930.50	40,930.50	-117.85	1218.09	40,930.50	40,930.50	40,930.50	-117.85	1216.86
Cordeau_pr04	-	-	7200		50,580.46	50,580.46	50,580.46	-	1230.46	50,580.46	50,580.46	50,580.46	-	1230.68
Cordeau_pr05	-	-	7200		50,022.89	50,022.89	50,022.89	-	1250.64	50,022.89	50,022.89	50,022.89	-	1253.73
Cordeau_pr06	-	-	7200		68,366.90	68,366.90	68,366.90	-	1276.87	68,366.90	68,366.90	68,366.90	-	1277.04
Cordeau_pr07	22,940.81	100.00	7200		9387.03	9996.68	10,649.54	-144.39	2107.87	9577.79	10,422.29	11,418.10	-139.52	2105.87

Table 5 continued

Instance	BKS_CPLEX				POPMUSIC _{w/o} (12)				POPMUSIC _w (12)					
	UB	Gap (%)	t (s)		Min	Avg	Max	Gap2 (%)	t (s)	Min	Avg	Max	Gap2 (%)	t (s)
Cordeau_pr08	74,941.48	100.00	7200		32,611.42	32,611.42	32,611.42	-129.80	1822.31	32,020.00	32,463.56	32,611.42	-134.05	1821.83
Cordeau_pr09	-	-	7200		42,748.00	42,748.00	42,748.00	-	1852.31	42,748.00	42,748.00	42,748.00	-	1853.91
Cordeau_pr10	-	-	7200		61,794.38	61,794.38	61,794.38	-	1903.14	61,794.38	61,794.38	61,794.38	-	1902.27
Avg.	-	-	-		47,772.88	47,800.21	47,860.37	-28.26	1116.22	47,781.74	47,805.31	47,860.37	-28.15	1287.15
Avg_vseplex	24,429.40	99.93	7200.00		14,039.48	14,258.10	14,514.77		1372.22	13,749.64	14,246.53	14,723.58		1368.77

size, solving a problem as a whole by means of a general purpose solver such as CPLEX may be advisable rather than partitioning it. That tendency is reversed when the size of the problem instances reaches 100 customers. In such cases POPMUSIC outperforms those results provided by the optimization models. Thus, addressing large problem instances, as in the latter cases, general purpose solvers as CPLEX require quite some amount of memory and time. Furthermore, it can be observed that the time performance and the overall quality of the solutions are better when POPMUSIC uses the optimization model without a lower bound on the number of vehicles.

For those problem instances proposed by Cordeau et al. [4], it can be clearly observed that besides three instances (i.e. Cordeau_p01, Cordeau_p02, and Cordeau_p12), in the rest of the instances POPMUSIC provides better quality results within reasonably less computational time. The above is summarized in row *Avg-vsCPLEX* which only considers those instances where CPLEX is able to provide a feasible solution. Furthermore, for these instances, both POPMUSIC approaches exhibit a similar performance in terms of quality of the solution and computational times. It is worth to point out that these results and those large scenarios of the previous table shed light about the capability of POPMUSIC to tackle large-scale scenarios. Finally, the performance variance among the different approaches highlights what could be observed in advance in Table 2. Depending on the problem instance, one optimization model within POPMUSIC may perform better than the other and not coincide in all cases with the best optimization model reported in Table 2. For instance, in Cordeau_pr02 the best performing optimization model was MD-CCVRP w/o (12) but the best performing POPMUSIC approach was the one embedding MD-CCVRP w/(12). This directly relates to the way POPMUSIC builds the sub-problems which is based on a seed part selected at random as well as the solver performance of sub-instances different than solving the complete problem. The aforementioned issue also points out a future research direction.

5 Conclusions

In this work, we have proposed and addressed the Multi-Depot Cumulative Capacitated Vehicle Routing Problem by means of a mathematical formulation and a POPMUSIC matheuristic approach that uses reduced versions of the problem instance at hand as a sub-problem to solve the overall problem. Different configurations of our approach have been investigated in order to determine a suitable one. Additionally, by considering the definition and characteristics of the problem, we have also proposed a lower bound on the minimum number of vehicles that considers the capacity of the vehicles and the amount required by the customers. The assessment of the contribution of such bound does not lead to an overall improvement of the quality of the solutions but tighter lower bounds. On the other hand, we have found that the optimization model without such constraint provides better results in terms of upper bounds. Nevertheless, both models have shown to be complementary as, in some cases, one model provides best-known solutions when the other cannot. Due to the size of the problem instances, CPLEX is not able to provide a feasible solution for large-sized instances. Thus, by means of our decomposition approach based on POPMUSIC, we have addressed

them. The obtained results indicate that it allows to improve the majority of the best solutions provided by the optimization model while reducing the computational effort considerably. The advantage of POPMUSIC over the optimization model becomes more clear as the size of the instances increases. Moreover, concerning the way of constructing the sub-problems, we have observed a better performance by building the parts in a lexicographic way than considering the distance between the depots. We have also detected that POPMUSIC exhibits a similar performance with and without the additional lower bound on the number of vehicles; this may indicate the good search capability of POPMUSIC, especially bearing in mind the contribution of such bound when solving the complete problem by means of a general purpose solver.

As a future research topic, we plan to investigate the use of more advanced heuristics and metaheuristics within POPMUSIC for solving this and related vehicle routing problems as well as study different ways to decompose the solutions into parts for building the sub-problems. Moreover, we aim to study this problem when backhauling is considered. Another issue concerns the consideration of matches and conflicts, i.e., building tours where several customers must be served one after each other from the same vehicle or case, where customers are not allowed to be served from the same vehicle at all.

References

1. Bertacco, L., Brunetta, L., Fischetti, M.: The linear ordering problem with cumulative costs. *Eur. J. Oper. Res.* **189**(3), 1345–1357 (2008)
2. Blum, A., Chalasani, P., Coppersmith, D., Pulleyblank, B., Raghavan, P., Sudan, M.: The minimum latency problem. In: *Proceedings of the 26th Annual ACM Symposium on Theory of computing*, pp. 163–171. ACM (1994)
3. Chen, P., Dong, X., Niu, Y.: An iterated local search algorithm for the cumulative capacitated vehicle routing problem. In: Tan, H. (ed.) *Technology for Education and Learning*, pp. 575–581. Springer, Berlin (2012)
4. Cordeau, J.F., Gendreau, M., Laporte, G.: A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks* **30**(2), 105–119 (1997)
5. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. *Manag. Sci.* **6**, 80–91 (1959)
6. Fischetti, M., Laporte, G., Martello, S.: The delivery man problem and cumulative matroids. *Oper. Res.* **41**(6), 1055–1064 (1993)
7. Fischetti, M., Monaci, M.: Exploiting erraticism in search. *Oper. Res.* **62**(1), 114–122 (2014)
8. Gouveia, L., Voß, S.: A classification of formulations for the (time-dependent) traveling salesman problem. *Eur. J. Oper. Res.* **83**(1), 69–82 (1995)
9. Lalla-Ruiz, E., Voß, S.: Improving solver performance through redundancy. *J. Syst. Sci. Syst. Eng.* **25**(3), 303–325 (2016)
10. Lalla-Ruiz, E., Voß, S.: Popmusic as a matheuristic for the berth allocation problem. *Ann. Math. Artif. Intell.* **76**(1–2), 173–189 (2016)
11. Lalla-Ruiz, E., Voß, S., Expósito-Izquierdo, C., Melián-Batista, B., Moreno-Vega, J.M.: A POPMUSIC-based approach for the berth allocation problem under time-dependent limitations. *Ann. Oper. Res.* **253**(2), 871–897 (2017)
12. Lalla-Ruiz, E., Schwarze, S., Voß, S.: A matheuristic approach for the p-cable trench problem. In: Festa, P., Sellmann, M., Vanschoren, J. (eds.) *Learning and Intelligent Optimization*, pp. 247–252. Springer (2016)
13. Lysgaard, J., Wøhlk, S.: A branch-and-cut-and-price algorithm for the cumulative capacitated vehicle routing problem. *Eur. J. Oper. Res.* **236**(3), 800–810 (2014)
14. Martínez-Salazar, I., Angel-Bello, F., Alvarez, A.: A customer-centric routing problem with multiple trips of a single vehicle. *J. Oper. Res. Soc.* **66**(8), 1312–1323 (2015)

15. Montoya-Torres, J.R., Franco, J.L., Isaza, S.N., Jiménez, H.F., Herazo-Padilla, N.: A literature review on the vehicle routing problem with multiple depots. *Comput. Ind. Eng.* **79**, 115–129 (2015)
16. Ngueveu, S., Prins, C., Wolfler-Calvo, R.: An effective memetic algorithm for the cumulative capacitated vehicle routing problem. *Comput. Oper. Res.* **37**(11), 1877–1885 (2010)
17. Renaud, J., Laporte, G., Boctor, F.: A tabu search heuristic for the multi-depot vehicle routing problem. *Comput. Oper. Res.* **23**(3), 229–235 (1996)
18. Ribeiro, G., Laporte, G.: An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Comput. Oper. Res.* **39**(3), 728–735 (2012)
19. Rivera, J.C., Afsar, H.M., Prins, C.: Mathematical formulations and exact algorithm for the multitrip cumulative capacitated single-vehicle routing problem. *Eur. J. Oper. Res.* **249**(1), 93–104 (2016)
20. Salehipour, A., Sörensen, K., Goos, P., Bräysy, O.: Efficient GRASP + VND and GRASP + VNS metaheuristics for the traveling repairman problem. *4OR* **9**(2), 189–209 (2011)
21. Sumichras, R., Markham, I.: A heuristic and lower bound for a multi-depot routing problem. *Comput. Oper. Res.* **22**(10), 1047–1056 (1995)
22. Sze, J.F., Salhi, S., Wassan, N.: The cumulative capacitated vehicle routing problem with min–sum and min–max objectives: an effective hybridisation of adaptive variable neighbourhood search and large neighbourhood search. *Transp. Res. Part B Methodol.* **101**, 162–184 (2017)
23. Taillard, É.D., Voß, S.: POPMUSIC-partial optimization metaheuristic under special intensification conditions. In: Ribeiro, C.C., Hansen, P. (eds.) *Essays and Surveys in Metaheuristics*, pp. 613–629. Springer, Berlin (2002)
24. Taillard, É.D., Voß, S.: Popmusic. In: Martí, R., Pardalos, P.M., Resende, M.G.C. (eds.) *Handbook of Heuristics*, pp. 687–701. Springer, Berlin (2017). https://doi.org/10.1007/978-3-319-07124-4_31
25. Talarico, L., Meisel, F., Sörensen, K.: Ambulance routing for disaster response with patient groups. *Comput. Oper. Res.* **56**, 120–133 (2015)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.