Computers & Operations Research 158 (2023) 106293

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

An iterated local search algorithm for latency vehicle routing problems with multiple depots

Alan Osorio-Mora[a], John Willmer Escobar[b], Paolo Toth[a,*]

[a] DEI, Department of Electric, Electronic and Information Engineering, Alma Mater Studiorum, Università di Bologna, Bologna, Italy
[b] Department of Accounting and Finance, Universidad del Valle, Cali, Colombia

ARTICLE INFO

Keywords:
Cumulative routing
ILS
Latency
MDCCVRP
MD*k*-TRP
LLRP

ABSTRACT

The Multi-depot Cumulative Capacitated Vehicle Routing Problem (MDCCVRP) extends the recently proposed Cumulative Capacitated Vehicle Routing Problem (CCVRP). The aim is to minimize the sum of the arrival times at the customers considering a fleet of N_v capacitated vehicles and a set of N_d uncapacitated depots. This paper proposes valid lower bounds and a novel metaheuristic algorithm for the solution of the MDCCVRP. The initial solution is obtained by combining different heuristic approaches, while the improving phase consists of an iterated local search algorithm (ILS). Computational experiments on 78 MDCCVRP benchmark instances show that the proposed algorithm is able to find, within reasonable computing times, solution values globally better than those obtained by the state-of-the-art heuristic algorithms. For challenging instances (having a large number of customers and a small fleet size), the algorithm can find, within short computing times, solutions globally better than those obtained by the published exact algorithms. The proposed algorithm has also been applied to the recently introduced Multi-depot k-traveling Repairman Problem (MDk-TRP) and the Latency Location Routing Problem (LLRP). The MDk-TRP is a particular case of the MDCCVRP arising when the vehicles are uncapacitated, while the LLRP is a generalization of the MDCCVRP in which, at most, p of the N_d available depots can be used. The computational experiments performed on 87 MDk-TRP benchmark instances and 76 LLRP benchmark instances show that the proposed algorithm globally outperforms the state-of-the-art metaheuristic algorithms for what concerns both the solution quality and the computing time. For large-size instances, the computing time required to provide a good quality solution is considerably smaller than that required by the previously published heuristic and exact algorithms. For all the problems, the proposed algorithm is able to find better solution values than those obtained by the respective state-of-the-art metaheuristic algorithms when it is executed for the same computing time as the respective competitor.

1. Introduction

The Multi-depot Cumulative Capacitated Vehicle Routing Problem (MDCCVRP) is a variant of the well-known Multi-depot Vehicle Routing Problem (MDVRP), in which, instead of minimizing the total travel time of the system, the aim is to minimize its global latency. The latency can be defined as the sum of the arrival times at the customers, and it is a metric used for defining customer satisfaction. Although the MDVRP is a well-known variant of the vehicle routing problems (a survey on MDVRPs can be found in Montoya-Torres et al. (2015)), the classical objective function is not appropriate for solving real-world cases concerning customer satisfaction. Indeed, it has been proved that the optimal solutions for classical routing problems lead to sub-optimal solutions for cumulative (latency) routing problems (Sze et al., 2017). This work suggests that new methods must be developed for solving effectively cumulative routing problems.

Research on cumulative capacitated vehicle routing problems (CCVRP) dates from 2010 with the seminal work of Ngueveu et al. (2010). The CCVRP is a particular case of the MDCCVRP considering only one depot. For a recent survey paper on cumulative-based vehicle routing problems, focusing on the CCVRP, the reader is referred to Corona-Gutiérrez et al. (2022). Although the CCVRP was relatively recently introduced, the research on latency routing problems started in the early 90 s with the traveling repairman problem (TRP) (Tsitsiklis, 1992), also known as delivery man problem (Fischetti et al., 1993) and minimum latency problem (MLP) (Blum et al., 1994). The TRP consists of finding the best sequence for visiting a set of customers considering a single vehicle, such that the latency is minimized.

Some natural extensions of the TRP have been studied by different researchers. The k-TRP (Fakcharoenphol et al., 2007) corresponds to a generalization of the TRP by considering k uncapacitated vehicles.

[*] Corresponding author.
E-mail addresses: alan.osorio2@unibo.it (A. Osorio-Mora), john.wilmer.escobar@correounivalle.edu.co (J.W. Escobar), paolo.toth@unibo.it (P. Toth).

https://doi.org/10.1016/j.cor.2023.106293
Received 1 February 2023; Received in revised form 22 April 2023; Accepted 29 May 2023
Available online 1 June 2023

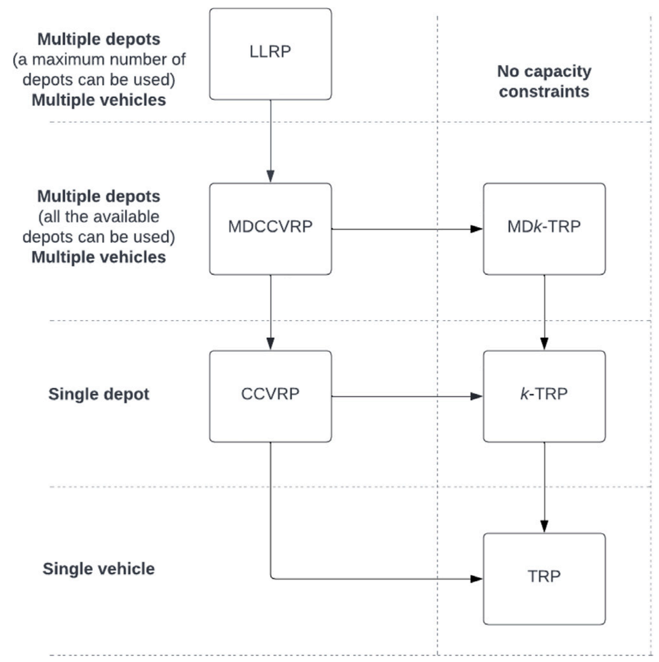


Fig. 1. Relations between the different latency-based routing problems.

This problem has also been called m-MLP (Ángel-Bello et al., 2019). The MDk-TRP (Bruni et al., 2022a) is a generalization of the k-TRP by considering multiple depots. The MDk-TRP is also a particular case of the MDCCVRP in which the vehicles have a capacity large enough to serve the demand of the customers. A generalization of the MDCCVRP is the latency location routing problem (LLRP) (Moshref-Javadi and Lee, 2016), arising when at most a fixed number of the available depots can be used. Fig. 1 shows a diagram pointing out the relations among the previously mentioned problems. All these problems are NP-hard since they can be reduced to the TRP, which has been proved to be NP-hard (Fakcharoenphol et al., 2007; Ngueveu et al., 2010; Moshref-Javadi and Lee, 2016; Lalla-Ruiz and Voß, 2020; Bruni et al., 2022a).

The MDCCVRP can be defined by considering a complete undirected graph $G = (V, E)$, where V corresponds to the set of nodes and E to the set of edges. The set V equals $V' \cup D$, with V' representing the set of N_c customers and D representing the set of N_d uncapacitated depots. Let also K be the set of N_v homogeneous vehicles, each with capacity Q. Each customer $i \in V'$ has a non-negative demand q_i (with $q_i \leq Q$). Each edge $(i, j) \in E$, with $i \neq j$, has an associated non-negative travel time c_{ij}, which satisfies the triangular inequality. The problem consists of defining the routes to be performed by the vehicles, minimizing the sum of the arrival times at the customers. Each customer must be visited once. Each vehicle can perform one route. Each route starts from a depot and visits a subset of customers whose global demand cannot exceed the vehicle capacity Q. As in the classical MDVRP, the MDCCVRP does not consider the availability of the vehicles at each depot. Note that it is not mandatory to use all the depots. The objective of the cumulative routing problems is to minimize the sum of the arrival times at the customers. Hence the last edge of each route (connecting the last customer of the route with a depot) has not to be considered in evaluating the objective function (Ngueveu et al., 2010). Therefore, the routes can be considered as "open routes".

The notation of the problem is summarized in Table 1.

The MDCCVRP was introduced in Lalla-Ruiz and Voß (2020), where a mixed-integer linear programming (MILP) formulation and a POPMUSIC matheuristic algorithm are proposed to solve the problem. The POPMUSIC method begins generating an initial solution through a greedy clustering algorithm. The idea is to improve the solution by partitioning the MDCCVRP into smaller sub-problems. The sub-problems are solved separately with the commercial solver CPLEX, and then included within the global matheuristic solution. Some relevant features of the MDCCVRP have been discussed in that work; in particular, the proof that the number of vehicles in the optimal solution is equal to $min\{N_v, N_c\}$, and the relations of the MDCCVRP with the TRPs.

In Wang et al. (2020), a perturb-based local search (PLS) metaheuristic algorithm has been proposed to solve the MDCCVRP. A constructive heuristic based on the k-regrets insertion criterion (Mattos Ribeiro and Laporte, 2012) is used for finding the initial solution. Then, a local search procedure is applied by exploring six different moves under the first improvement criterion. Once no improvement can be reached, 2-opt and 2-exchange moves are applied to perturb the solution and explore a new search space. The efficiency of this approach is compared with the results previously presented in Lalla-Ruiz and Voß (2020).

A branch-cut-and-price algorithm for solving the CCVRP and the MDCCVRP is proposed in Damião et al. (2021). The algorithm is able to provide the optimal solution for many small/medium size instances and high-quality solutions for large-size instances by fixing the maximum number of customers in each route. Two MILP formulations have been proposed in Nucamendi-Guillén et al. (2022) to solve the LLRP. These formulations were also adapted to solve the MDCCVRP, and were able to find some optimal solutions for small/medium-size instances (with up to 50 customers) for both the LLRP and the MDCCVRP. The formulations were able to solve large MDCCVRP instances (with up to 192 customers) when a large number of vehicles is considered (in Lalla-Ruiz and Voß (2020) it has been proved that these instances are easier than the same instances with a relative small number of vehicles).

Although the problem considered in Wang et al. (2016) does not correspond to an MDCCVRP, it is important to point out the differences between these two problems. Indeed, the former problem has been defined by its authors as a cumulative multi-depot vehicle routing problem (Cu-MDVRP), considering it as a generalization of the cumulative vehicle routing problem (Cu-VRP) proposed in Kara et al. (2008). The Cu-VRP seeks to minimize the sum of the travel times of the traversed edges weighted by the load on the vehicle at the moment of traversing the edges. Nevertheless, the Cu-MDVRP presented in Wang et al. (2016) seeks to minimize the sum of the arrival times at the customers weighted by their demands. Thus, it can be considered as a weighted version of the MDCCVRP. In addition, Cu-MDVRP considers a certain number of vehicles available at each depot, which is another feature that makes it different from the MDCCVRP. For more details about these two families of problems (Cu-VRPs and CCVRPs) the reader is referred to the literature review presented in Corona-Gutiérrez et al. (2022).

The MDk-TRP was recently introduced in Bruni et al. (2022a). For its solution, the authors proposed two MILP models and a genetic algorithm (GA) under two different configurations. The formulations,

Table 1
Notation of the multi-depot cumulative capacitated vehicle routing problem.

Sets	
V':	Set of N_c customers.
D:	Set of N_d identical uncapacitated depots.
V:	Set of nodes, $V = V' \cup D$.
K:	Set of N_v identical vehicles.
Parameters	
Q:	Capacity of the vehicles.
q_i:	Demand of customer i ($i \in V'$).
c_{ij}:	Travel time between nodes i and j ($i, j \in V', i \neq j$).
p:	Maximum number of used depots for the LLRP.

which can solve to proven optimally several large-size instances (with over 200 customers), are based on the multi-level network approach. This approach was previously used to solve other related latency routing problems as the MLP (Ángel-Bello et al., 2013), the k-TRP/m-MLP (Nucamendi-Guillén et al., 2016; Ángel-Bello et al., 2019), and the CCVRP (Nucamendi-Guillén et al., 2018). More recently, as we already mentioned, it was also used to solve the LLRP and the MDCCVRP (Nucamendi-Guillén et al., 2022). On the other hand, the GA is able to solve large-size instances within short computing times.

The LLRP, which is a combination of the facility location problem (FLP) and the CCVRP, was introduced in Moshref-Javadi and Lee (2016). The LLRP can be considered as an extension of the MDCCVRP in which, at most, p of the N_d available depots can be used (i.e. opened). Two heuristic algorithms to solve efficiently the LLRP (a memetic algorithm (MA) and a recursive granular algorithm (RGA)) have been proposed in Moshref-Javadi and Lee (2016). According to the reported computational experiments, MA performs better than RGA for the complete set of the considered instances. Recently, two MILP models, three enumerative algorithms, and a GRASP-based iterated local search algorithm (GBILS) have been proposed in Nucamendi-Guillén et al. (2022) to solve the LLRP. The authors provide the optimal solution for several instances with up to 50 customers using the five exact methods, while the metaheuristic algorithm GBILS found globally better quality solutions than those obtained by the algorithms RGA and MA within short computing times. The best results on the benchmark instances currently considered for the LLRP have been reported for the three metaheuristic algorithms presented in Osorio-Mora et al. (2023). These algorithms combine simulated annealing (SA) and variable neighborhood descent (VND) procedures. The main difference between the three proposed algorithms is the VND used strategy. The proposed approaches outperform the state-of-the-art algorithms for the LLRP, being able to find better quality solutions within comparable computing times.

Applications of multi-depot latency routing problems have also been studied, especially in post-disaster and customer-centric contexts. A problem in which the visit to affected areas must be planned after a natural disaster is studied in Ajam et al. (2022), where also the possibility of restoration of blocked paths is considered. The authors proposed a mixed-integer programming model and two heuristic algorithms based on the cluster-first-route-second approach for solving the problem. A bi-objective location routing problem under uncertainty applied to humanitarian logistics is studied in Zhong et al. (2020). The problem considers time windows and a heterogeneous fleet of vehicles, while a risk-averse approach is used for minimizing the total cost and the latency of the system. The problem was solved with a hybrid genetic algorithm. Bruni and Khodaparasti (2022) propose a VND matheuristic for the Drone Routing Problem in the context of last-mile delivery. The problem is formulated as a deterministic location-routing model and derives its robust counterpart under the travel time uncertainty.

This paper proposes an iterated local search metaheuristic algorithm called M-ILS to solve effectively the MDCCVRP, the MDk-TRP, and the LLRP. The reported computational experiments on benchmark instances from the literature show that the proposed algorithm finds several current and new best-known solutions within computing times comparable with those required by the state-of-the-art algorithms proposed for the considered problems. Furthermore, the optimal solution values reported in Bruni et al. (2022a) are rectified for several MDk-TRP instances. Finally, valid lower bounds for the MDCCVRP and the MDk-TRP are presented.

The paper is organized as follows. Section 2 describes the proposed metaheuristic and presents lower bounds for the MDCCVRP and the MDk-TRP. The computational results are reported and analyzed in Section 3. Finally, in Section 4, a summary of our findings and future directions are drawn.

Table 2
Algorithm's parameters.

it_{max}	:	Number of iterations of the iterated local search.
it_{soft}	:	Frequency of the $Route-Relocation$ perturbation.
it_{hard}	:	Frequency of the $Configuration-Swap$ perturbation.
a	:	Percentage for the penalization for each unit exceeding the capacity of the vehicles.
t_0	:	Initial temperature in the SA-VND procedure.
t_f	:	Minimum temperature in the SA-VND procedure.
α	:	Cooling factor in the SA-VND procedure.

2. Description of the proposed approach

This section presents an M-ILS metaheuristic approach for solving the MDCCVRP, the MDk-TRP, and the LLRP. Besides, valid lower bounds are proposed for the MDCCVRP and the MDk-TRP. The main body of the proposed approach (M-ILS algorithm) consists of two major phases: the construction phase and the improvement phase. The goal of the construction phase is to build an initial feasible solution s_0 (see Section 2.1). In the improvement phase, an Iterated Local Search (ILS) scheme, considering several diversification and local search procedures, is applied to improve the quality of the current solution. The ILS procedure starts by setting the current solution s_c, and the best feasible solution s_{bf} equal to s_0. It consists of three procedures executed for it_{max} iterations: a perturbation procedure, a local search procedure called LS, and a procedure combining the simulated annealing (SA) and the variable neighborhood descent (VND) frameworks (this procedure is called SA-VND). After the execution of the ILS, the well-known Lin–Kernighan–Helsgaun heuristic (LKH-3) (Helsgaun, 2017) is used to solve a CCVRP (for the MDCCVRP and the LLRP) or a k-TRP (for the MDk-TRP) for each open depot, considering the best feasible solution. The local search procedure LS is applied to the solution obtained by the LKH-3 algorithm. Finally, for each route, a procedure called checking is applied. This procedure verifies if each route's first customer is assigned to its closest depot. If this is not the case, the closest depot is assigned to the corresponding route. The details of the ILS procedure are described in Section 2.2, and a summarized representation of the overall algorithm is presented in Algorithm 1. Table 2 shows the parameters used for the proposed approach.

The key points for the success of the proposed approach are the correct selection of the depots using the heuristic procedure of the construction phase. Besides, the local search and diversification procedures within the improvement phase and the Lin–Kernighan–Helsgaun heuristic (LKH-3) allow an efficient exploration of the search space. Since the most critical decisions of the multi-depot variants of the single-depot vehicle routing problems are initially those concerning the use and assignment of the depots, a correct selection of the depots can reduce the search space for the improvement phase (avoiding local search procedures between depots). Also, starting from a good initial feasible solution allows for improving the current solution by applying the correct local search procedures. The previously mentioned procedures are described in more detail in the following subsections.

2.1. Construction phase: Initial solution

In this phase, we propose an efficient procedure to construct an initial feasible solution. The procedure is based on an approach that combines different heuristic procedures, including the LKH-3 heuristic. Besides, a cluster-based method is considered as starting point within the initial iterative framework. The initial solution s_0 is obtained by the following Constructive procedure, which generally finds good feasible solutions within short computing times:

-Step 1: Considering all the customers, construct the corresponding giant Traveling Salesman Problem (TSP) tour using the LKH-3 heuristic.

```
Algorithm 1: Main Scheme
   Input: A MDCCVRP/MDk-TRP/LLRP instance, Algorithm
          parameters
   Output: $s_{bf}$ ($Best\ feasible\ solution$)
1  Constructive procedure — return: $s_0$ (initial solution)
2  Iterated local search:
3  $it = 0$
4  while ($it < it_{max}$) do
5      step 1: Perturbation
6      step 2: Local search (LS)
7      step 3: SA-VND search procedure
8      it=it+1
9  end
10 For each used depot solve a CCVRP/k-TRP applying the LKH-3
     heuristic
11 Local search (LS)
12 Checking
   return: $s_{bf}$
```

Note that for the giant TSP, the global travel time to visit all the customers is minimized.

-Step 2: A good initial solution can be obtained by identifying clusters of customers. To this end, the giant tour is split into N_v clusters, as described below. We apply a clustering procedure by considering each customer as a "starting point", and by splitting the giant tour into groups of consecutive customers. For the MDCCVRP and the MDk-TRP, the splitting aims to balance the solution. The first ($N_c\ mod\ N_v$) clusters are composed of $\lceil \frac{N_c}{N_v} \rceil$ customers, while the remaining clusters are composed by $\lfloor \frac{N_c}{N_v} \rfloor$ customers. The idea of considering balanced solutions is based on the valid lower bound LB2 presented in Section 2.3. For the LLRP, a different clustering procedure is applied. It consists of splitting the giant tour into groups of consecutive customers such that the total load of each cluster does not exceed the capacity of the vehicles. If the number of clusters created is larger than N_v, a repair procedure is applied to delete the least-loaded clusters until the number of clusters equals N_v. The customers are removed from the least-loaded clusters according to the order given in the clusters. Each customer belonging to a least-loaded cluster is removed from its current position and inserted in its best position in a different cluster, so as to minimize the score defined in Eq. (1):

$$ScoreIN_{ik}^{j} = \Delta instime_{j}^{ik} + \theta[max\{0,(dc_j + q_i) - Q\}] \tag{1}$$

where: $\Delta instime_j^{ik}$ represents the variation of the travel time of cluster j caused by the insertion of customer i in position k, dc_j represents the current load of cluster j, and θ represents a penalization parameter (large positive value). The process is repeated until all the least-loaded clusters are deleted.

-Step 3: If the total load of a cluster (say cluster j) exceeds the vehicle capacity, a swapping procedure is applied to two customers (say customers k and i, with $q_i < q_k$) with respect to their current clusters (say clusters j and l, respectively, with $j \neq l$), so as to minimize the following score:

$$ScoreSW_{ik}^{jl} = \Delta time_{j}^{ki} + \Delta time_{l}^{ik} + \theta[max\{0,(dc_j - q_k + q_i) - Q\} + max\{0,(dc_l + q_k - q_i) - Q\}] \tag{2}$$

where: $\Delta time_j^{ki}$ (resp. $\Delta time_l^{ik}$) represents the variation of the travel time of cluster j (resp. cluster l) caused by the exchange of the customers k and i. If no feasible splitting of the customers into N_v clusters is found by this swapping procedure, the exact algorithm MTP proposed in Martello and Toth (1990) is applied to the Bin Packing Problem (BPP) instance corresponding to the given MDCCVRP instance to obtain a set of at most N_v feasible clusters.

-Step 4: Let CL be the clusters set created in the previous step. For each depot $i \in D$ and for each cluster $j \in CL$, we define an allocation cost l_{ij}, which represents the total latency of the route composed by the customers in cluster j and depot i. This allocation cost is obtained by applying an intra-route local search ($IntraLS$) procedure to the path generated for the cluster (more details about this local search procedure are presented in Section 2.2.2).

Step 5: The best assignment of the clusters to the depots for the MDCCVRP and the MDk-TRP is obtained by assigning each cluster $j (j \in CL)$ to the depot i such that $l_{ij} = min\{l_{hj} : h \in D\}$. Thus, the latency of the solution is given by $\sum_{j \in CL} \min_{i \in D}\{l_{ij}\}$. We define a depot $i \in D$ as "used" if at least one cluster is assigned to i. A depot configuration corresponds to a binary vector $Configuration$ of size N_d that indicates if depot $i \in D$ is used ($Configuration_i = 1$) or not ($Configuration_i = 0$). All the previously mentioned information is stored in this *Step*.

For the LLRP, the best assignment of the clusters to the depots is obtained by solving the integer linear programming (ILP) model (3)–(8). We introduce two sets of binary variables, where A_{ij} is equal to 1 if cluster j is assigned to depot i ($i \in D, j \in CL$), and y_i is equal to 1 if depot i ($i \in D$) is opened.

$$min \sum_{i \in D} \sum_{j \in CL} l_{ij} A_{ij} \tag{3}$$

$$\sum_{i \in D} A_{ij} = 1 \qquad \forall j \in CL \tag{4}$$

$$A_{ij} \leq y_i \qquad \forall i \in D, \forall j \in CL \tag{5}$$

$$\sum_{i \in D} y_i \leq p \tag{6}$$

$$A_{ij} \in \{0,1\} \qquad \forall i \in D, \forall j \in CL \tag{7}$$

$$y_i \in \{0,1\} \qquad \forall i \in D \tag{8}$$

where l_{ij} is the latency of the route composed by the customers in cluster j and depot i (see *Step 4*), dc_j is the global demand of cluster j, and p is the maximum number of depots to be opened.

The objective function (3) seeks to minimize the total latency. Constraints (4) ensure that each cluster is allocated to exactly one depot. Constraints (5) impose that the clusters can be allocated only to open depots. Eq. (6) ensures that the maximum number of open depots is at most p. Finally, constraints (7)–(8) define the domain of the variables.

-Step 6: Note that there are exactly N_c different possibilities to split the giant tour, since in the clustering procedure, the definition of the clusters depends only on the choice of the first customer (starting point), and the clustering procedure is applied N_c times, by considering each of the N_c customers as the starting point. *Steps 2–5* are repeated until all the solutions corresponding to the possible splittings of the customers into clusters are evaluated.

A list of promising depot configurations stores all the configurations obtained at *Step 5*, the number of times that each configuration is selected, and the allocation of the clusters to the depots that provides the minimum latency for that configuration. At each of the N_c iterations, if the current $Configuration$ has not yet been stored in the list, it is stored with the respective latency and the allocation of the clusters to the depots. On the other hand, if the current $Configuration$ has been already stored in the list, the number of times that the corresponding depot configuration has been selected is updated, and if the latency associated with the new allocation is smaller than that previously stored, the best latency, and the respective allocation are updated. At the end of the N_c iterations, the solution corresponding to the depot configuration with the minimum latency is selected. Each cluster is considered an open route, which starts from the assigned depot, and the sequence of the customers is as done at *Step 4*. Finally, the list of the promising depot configurations is sorted according to the number of times that each configuration was selected, putting in the first positions those configurations that were selected more times. This list will be used in the perturbation procedure described in Section 2.2.2. For the LLRP, a splitting procedure is applied to add new vehicles if the number of open routes currently created is smaller than N_v. The splitting procedure is performed based on the idea that the total latency decreases by adding new routes for the same or different depots. It consists of a local search procedure with the following steps.

- Select the route r containing the longest edge (i, j) (where i and $j \in V'$).
- Split the route r (starting from depot h) by removing edge (i, j). Two new sub-routes ($r1$ and $r2$) are created. $r1$ is the sub-route starting from depot h and composed of the customers belonging to route r until customer i. $r2$ is the sub-route composed by the customers of route r from customer j to the final customer of route r.
- Assign sub-route $r2$ to the best depot by considering its current or its reverse sequence, so as to minimize the corresponding latency.
- The procedure is performed until the number of routes of the current solution equals N_v.

-*Step 7*: Apply the LKH-3 heuristic for each depot with its assigned routes, solving a CCVRP or a k-TRP depending if the problem to solve is a MDCCVRP/LLRP or a MDk-TRP, respectively.

-*Step 8*: Apply the local search procedure LS until no improvement is found (for more details about the procedure LS, see Section 2.2.1). This procedure allows infeasible solutions in terms of vehicle capacity. These solutions are penalized by using a factor a, defined as a percentage of the solution value provided at the end of *Step 7*.

Although there are similarities between the algorithm proposed in this paper and those proposed in Escobar et al. (2013, 2014b,a), substantial differences are pointed out in the following. The major difference is given by the improvement phase of the proposed algorithm (which will be described in the next section) since all the mentioned works presented a tabu-search-based approach, while this paper proposes an iterated local search algorithm. Furthermore, due to the differences in the cumulative and classical vehicle routing problems, there are several differences concerning the construction of the initial solution and the diversification/intensification strategies between this paper and the mentioned works. Regarding the construction of the initial solution, in *Step 4*, the mentioned works use the LKH heuristic to solve a TSP and to calculate the values l_{ij}, while in this paper, the $IntraLS$ procedure was specifically designed for solving a TRP. In preliminary computational experiments, an approach where the TRPs were solved using the LKH-3 heuristic was tested, but it led to extremely high computing times for large-size instances. In addition, the construction of the initial solution presented in the mentioned papers does not ensure that $min\{N_c, N_v\}$ vehicles will be used, while the procedure proposed in this paper does it. Another important difference is the inclusion of the binary vector storing the promising depot configurations. It is noted that in the previously mentioned works, the best configuration of the depots is first selected, and then no change of the used depots is performed. This situation may lead to skipping promising parts of the search space.

2.2. Improvement phase: Iterated local search algorithm (ILS)

In this phase, the algorithm tries to improve the initial solution s_0 by applying an Iterated Local Search (ILS) procedure. The ILS algorithms have been successfully applied to a wide number of combinatorial optimization problems, and the main idea is to explore new regions of the solution space by applying a perturbation when a local optimum is reached. For further details about the ILS algorithms the reader is referred to Lourenço et al. (2019).

For the proposed ILS, the current solution s_c and the best feasible solution s_{bf} are initially equal to the initial solution s_0. The three steps of the ILS are described in this section. Note that the local search procedure LS (step 2) is explained before the perturbation step since the neighborhoods and the local search procedures are used in the three steps of the algorithm.

2.2.1. Local search

In this section, the search space and the neighborhoods are described. The proposed algorithm accepts solutions infeasible with respect to the vehicle capacity to avoid local optima and extend the search space. Thus, the value of the objective function $f(s_c)$ of a solution s_c, feasible or not, is given by the following formula:

$$f(s_c) = \bar{f}(s_c) + af(s_0)\Delta QV \tag{9}$$

where $\bar{f}(s_c)$ is the sum of the arrival times at the customers, $f(s_0)$ is the value of the objective function corresponding to the initial solution s_0 found at *Step 8* of the constructive procedure, and ΔQV is the total amount of load violating the capacities of the vehicles. It is noted that for the feasible solutions the second term of (9) is equal to zero, and for the case of the MDk-TRP, this term is always equal to zero. The best improvement strategy is used in the local search-based procedures.

The proposed algorithm executes the following five types of moves: $insertion$, $swap$, $2-opt$, $arc-swap$, and $shift_{2-1}$. All the neighborhoods, except $shift_{2-1}$, can be applied for the intra-route and the inter-route cases. For the inter-route case, the moves can be applied for routes starting from the same or from different depots. The neighborhoods are the following ones:

- $insertion$: A customer i is transferred from its current position to another position just after node j. Note that the selected customer can be moved to a different position in the same or in different routes.
- $swap$: Two customers (i and j) exchange their positions, either in the same route or between different routes.
- $2-opt$: This move is a classical version of the well-known $2-opt$ move for the TSP, in which two non-consecutive edges are removed, and the routes are reconnected differently. Note that if the two selected edges are in the same route, the two opt move is equivalent to that described by Lin and Kernighan (1973). In particular, two edges (i, j) and (k, l) are deleted, and two new edges are created. When the move is applied to edges belonging to the same route, the edges (i, j) and (k, l) are deleted, then the edges (i, k) and (j, l) are created, and the connection from k to j is reversed. On the other hand, when the move is related to two different routes, a crossing is applied: the edges (i, j) and (k, l), with the edge (i, j) in route 1 and the edge (k, l) in route 2, are deleted, then the edges (i, l) and (k, j) are created, and the initial customers of routes 1 and 2 (until nodes i and k, respectively) are merged with the final customers of routes 2 and 1 (from customers l and j, respectively).
- $arc-swap$: Two pairs of consecutive customers (i, j) and (k, l) are swapped with respect to their current positions. The two pairs of customers can belong to the same or to different routes.
- $shift_{2-1}$: Two consecutive customers (i, j) assigned to route $r1$ exchange their current positions with that of the customer k in the route $r2$, with $r1 \neq r2$.

The local search procedure LS (step 2), calls for exploring all the mentioned neighborhoods and applying the move which improves the most the current solution. The procedure stops when no improvement move is found.

2.2.2. Perturbation procedure

Since the ILS procedure can fail in finding a move to improve the current solution, the algorithm tries to escape from a local optimum by perturbing the current solution. The perturbation procedure considers three possible perturbations applied with different frequencies. The "less-aggressive" perturbations are called $Route-Swap$ and $Route-Relocation$, and the "most aggressive" one is called $Configuration-Swap$. $Route-Relocation$ is applied every $iter_{soft}$ iterations, $Configuration-Swap$ every $iter_{hard}$ iterations, while the $Route-Swap$ is applied at each iteration if no other perturbation is applied. The descriptions of the perturbations are given in the following paragraphs (where it is assumed that each random choice is performed by considering the same probability with respect to the possible choices):

- $Route - Swap$: We use an exchange scheme involving two routes. The procedure randomly selects two routes $r1$ and $r2$ belonging to two different depots i and j, respectively. A new solution s is obtained by considering the following move: remove the route $r1$ from the depot i and assign it to the depot j; remove route $r2$ from the depot j and assign it to the depot i. Since the new routes may not generate good solutions, an intra-route local search procedure ($IntraLS$) is applied to $r1$ and $r2$. The $IntraLS$ procedure sequentially explores each $insertion$, $swap$, and $2-opt$ neighborhood for the considered route until no improvement is found for the considered neighborhood.
- $Route - Relocation$: This perturbation procedure randomly selects a depot i with more than one route assigned. Then, the procedure randomly selects a route $r1$ belonging to the depot i. A new solution s is obtained by considering the following move: remove the route $r1$ from the depot i and assign it to a different depot j, which is randomly selected from all the remaining used depots. Then, the $IntraLS$ procedure is applied to the route $r1$.
- $Configuration - Swap$: This perturbation procedure swaps the current depot configuration (called $C1$) with the first one in the list of promising configurations (called $C2$). The list is sorted according to the number of times each configuration has been selected for the initial solution. Each time a configuration is evaluated, it is removed from the list. After picking $C2$, *Steps 7 - 8* of the constructive procedure are applied to construct the new solution. It is to note that, before applying the swapping, the LKH-3 heuristic and the LS procedure are applied to the best solution found during the exploration of the configuration $C1$.

For the exceptional cases in which the depot configuration considers only one available depot (with assigned routes), the perturbations $Route - Swap$ and $Route - Relocation$ are replaced by random moves applied under a simulated annealing framework described in Section 2.2.3. These random moves are applied to the current solution. If there are no more promising configurations to evaluate, $Configuration-Swap$ is skipped, and $Route - Relocation$ is applied to the best feasible solution. In this case, the perturbation procedure is not applied to the current solution. Note that $Route - Relocation$ cannot be applied when all the available depots in the current configuration have only one associated route.

The idea of applying different levels of aggressiveness in the perturbations is based on the fact that if only the "less-aggressive" perturbations are applied, the algorithm stacks into local-optimum solutions. After the application of $Route-Relocation$ or $Configuration-Swap$, the proposed local search operators cannot find the same local-optimum solution found previously. Indeed, these perturbation procedures change the allocation of routes to depots.

2.2.3. The SA-VND search procedure

The SA-VND procedure is presented in Algorithm 2.

This procedure starts by applying a simulated annealing framework (see Steps 1 to 11 of Algorithm 2). RandomMove(s_c) denotes the solution obtained by generating random moves with the same probability using the following neighborhoods: $insertion$, $swap$, and $2-opt$. The current temperature $temp$ is set to an initial temperature t_0. The SA procedure is applied until the minimum temperature t_f is reached ($temp \leq t_f$). A new solution s_p is generated by applying to the current solution s_c one of the mentioned random moves, and it is accepted as the new s_c if one of the following conditions holds (i.e., $AcceptanceCriteria(temp, s_p, s_c)$ is true): (i) $\Delta f = f(s_c) - f(s_p) > 0$, where $f(s_c)$ and $f(s_p)$ are the objective function values of the solutions s_c and s_p, respectively; or (ii) if $\Delta f \leq 0$ and $r < \exp^{(\Delta f/temp)}$ where r is a uniform random number in the interval $[0,1]$. If $f(s_c) < f(s_{bf})$, and s_c is feasible (i.e., $IsFeasible(s_c)$ is $true$), the current solution is updated as the best feasible solution s_{bf} found so far. Then, the value of $temp$ is reduced according to a cooling factor α.

After the SA framework, a variable neighborhood descent (VND) procedure is applied to the current solution s_c (see Steps 12 to 30 of Algorithm 2). Denote by $Neigh = \{insertion,\ swap,\ 2-opt,\ arc-swap,\ shift_{2-1}\}$ the set of the five previously described neighborhoods, and consider $ng \in Neigh$ as the ngth neighborhood of the current solution s_c. In addition, $sol(ng, s_c)$ denotes the solution obtained by exploring the neighborhood ng starting from the solution s_c. The neighborhoods are explored according to the order in which they are listed in the set $Neigh$. In the VND procedure, the exploration starts from the first neighborhood, which is explored until no improvement is found. Then, the search moves to the next neighborhood, and the process is repeated until the last neighborhood does not improve the current solution s_c. Otherwise, the search is restarted from the first neighborhood. If no improvement is found for all the neighborhoods, the VND procedure ends. Each time that a move is applied to the current solution s_c, if $f(s_c) < f(s_{bf})$, and s_c is feasible, the current solution s_c is updated as the best feasible solution s_{bf} found so far.

Algorithm 2: The SA-VND search procedure.

```
Input: $t_0$, $t_f$, $\alpha$, $s_c$, $s_{bf}$, $Neigh$
Output: $s_c$ ($New\ current\ solution$), $s_{bf}$ ($New\ current\ best\ feasible\ solution$)
/* simulated annealing procedure                          */
1  $temp = t_0$
2  while ($temp > t_f$) do
3      $s_p$ = RandomMove($s_c$)
4      if ($AcceptanceCriteria(temp, s_p, s_c)$) then
5          $s_c = s_p$
6          if ($f(s_c) < f(s_{bf})$ and $IsFeasible(s_c)$) then
7              $s_{bf} = s_c$
8          end
9      end
10     $temp = \alpha * temp$
11 end
/* variable neighborhood descent procedure                */
12 $flag_{vnd} = true$
13 while ($flag_{vnd} = true$) do
14     $flag_{vnd} = false$
15     for ($each\ ng \in Neigh$) do
16         $flag_{neigh} = true$
17         while ($flag_{neigh} = true$) do
18             $s_{vnd} = sol(ng, s_c)$
19             if ($f(s_{vnd}) < f(s_c)$) then
20                 $s_c = s_{vnd}$
21                 $flag_{vnd} = true$
22                 if ($f(s_c) < f(s_{bf})$ and $IsFeasible(s_c)$) then
23                     $s_{bf} = s_c$
24                 end
25             else
26                 $flag_{neigh} = false$
27             end
28         end
29     end
30 end
return: $s_c$, $s_{bf}$
```

2.3. Lower bounds

This section describes two lower bounds, $LB1$ and $LB2$, proposed for both the MDCCVRP and the MDk-TRP. Note that lower bounds for the LLRP have been already proposed in Moshref-Javadi and Lee (2016). The lower bounds for the MDCCVRP and the MDk-TRP generalize those proposed in Ngueveu et al. (2010) for the CCVRP.

Lower bound LB1: The first lower bound does not restrict the vehicle fleet size and considers one vehicle (i.e., one route) for each customer. The optimal solution for the unrestricted fleet problem is to assign each customer to its closest depot. The value of this solution is a valid lower bound for the considered problem:

$$LB1 = \sum_{i\in V'} \min_{j\in D}\{c_{ij}\} \tag{10}$$

Lower bound LB2: The second lower bound assumes a cardinality balanced solution, i.e., a solution where the routes visit an equal number of edges (i.e. of customers) or at most one edge of difference between the route with the largest number of edges and that with the smallest number of edges. Let us define NE_k as the number of edges associated with route $k, \forall k \in K$. The first (N_c *mod* N_v) routes are composed of $\lceil \frac{N_c}{N_v} \rceil$ edges, while the last $N_v - (N_c$ *mod* $N_v)$ routes are composed by $\lfloor \frac{N_c}{N_v} \rfloor$ edges.

For the cumulative (latency) routing problems, the edges from the last customer of a given route to the associated depot do not affect the objective function. Therefore, it is unnecessary to consider them in the solution. Thus, as the total number of edges of a given solution equals the number of customers, $LB2$ considers N_c edges. The first N_v edges must correspond to the shortest edges between depots and customers, while the last $(N_c - N_v)$ edges must correspond to the shortest edges between customers. Let us also define the sets $EDC \subset E$ and $ECC \subset E$ as the sets of the edges between depots and customers and between two different customers, respectively. The vectors $CEDC$ and $CECC$ contain the travel time associated with each edge in EDC and ECC, respectively. The edges in the sets EDC and ECC are sorted according to ascending values of $CEDC$ and $CECC$, respectively.

The proposed lower bound can be computed as described in Algorithm 3. $LB2$ corresponds to the sum of the estimated latencies associated with each route. Due to the nature of the latency functions, the edges at the initial positions of the routes have the largest impact on the value of $LB2$. The procedure for the computation of this lower bound sorts the edges of the graph to include the shortest edges at the first positions of each route. Note that the routes must be sorted in descending order according to the value of NE_k. The shortest edges are included within the routes with the largest NE_k values (i.e., those having the largest impact on the value of the objective function). $LB2$ is divided into two parts: $LB2a$, associated with the edges from the depots to the customers, and $LB2b$, associated with the edges between two different customers.

Algorithm 3: LB2

```
   Input: $EDC$, $ECC$, $CEDC$, $CECC$, $NE$, $N_v$, $N_c$
   Output: $LB2$
1  $LB2a = 0$, $LB2b = 0$
   /* Computation of LB2a                                  */
2  for ($i = 1$ to $N_v$) do
3      $k = i$
4      $RE_k = NE_k$
5      $LB2a = LB2a + CEDC_i RE_k$
6      $RE_k = RE_k - 1$
7  end
   /* Computation of LB2b                                  */
8  $k = 1$
9  $i = 1$
10 for ($i = 1$ to $(N_c - N_v)$) do
11     if ($k > N_v$) then
12         k=1
13     end
14     $LB2b = LB2b + CECC_i RE_k$
15     $RE_k = RE_k - 1$
16     $k = k + 1$
17 end
18 $LB2 = LB2a + LB2b$
   return: $LB2$
```

It is important to remark that $LB2$ has been previously proposed in Ngueveu et al. (2010) and then generalized in Moshref-Javadi and Lee (2016). The definition of $LB2$ presented in Ngueveu et al. (2010) and Moshref-Javadi and Lee (2016) is given by Eq. (11), where W_e and W'_e represent the travel time of the eth shortest edge of the graph between depots and customers, and between two different customers, respectively.

$$LB2 = \sum_{e=1}^{N_v} \lceil * \rceil \frac{N_c + N_v - e - (N_c \; mod \; N_v)}{N_v} W_e + \sum_{e=1}^{N_c - N_v} \lceil * \rceil \frac{N_c - e - (N_c \; mod \; N_v)}{N_v} W'_e \tag{11}$$

Considering an instance with $N_c = N_v = 5$, it is possible to show that Eq. (11) does not define a valid lower bound for this instance. The optimal solution is to visit each customer on a different route. Hence, each edge connecting the depots to the customers impacts the objective function value once, while the edges connecting two customers (corresponding to the second summation of (11)) give no contribution; nevertheless, according to the first summation of (11), the first 4 (i.e., $N_v - 1$) shortest edges connecting the depots to the customers impact two times on the objective function value.

3. Computational results

The overall algorithm (M-ILS) has been implemented in C++, and the computational experiments have been performed on an Intel(R) Core(TM) i7-8700K CPU @ 3.70 GHz with 32 GB RAM, under Linux Ubuntu 18.04 operative system (single thread). The travel time matrix for all the considered instances was calculated with double precision. The ILP model (3)–(8) has been optimally solved using the ILP solver CPLEX 20.1 (IBM, 2021) under the default parameters configuration (one thread).

Since the previously published papers used different computers, the corresponding computing times are scaled using a “scaling factor”, which approximates the original computing times reported in each published paper to the expected computing time of the processor used in our experiments. The “scaling factor” is based on the PassMark performance test (https://www.cpubenchmark.net/), which is focused on evaluating the CPU and memory performance. Higher “Single Thread Rating” values indicate that the corresponding CPU is faster (considering one thread). The value of each “scaling factor” is calculated as the ratio between the “Single Thread Rating” value of each computer and the “Single Thread Rating” value of the computer used in our experiments. The details are presented in Table 3.

The tables showing the computational results obtained by the proposed algorithm (M-ILS) and by the state-of-the-art methods for the solution of the MDCCVRP, the MDk-TRP, and of the LLRP are presented in Sections 3.3–3.5, respectively.

For each instance, the following values are given:

- Instance: Name of the instance.
- N_c: Number of customers.
- N_d: Number of depots.
- N_v: Number of vehicles.
- BKS: Best known solution value considering all the algorithms. The underlined values have not been proved to be optimal.

For each algorithm and for each instance, the following values are reported:

- Best: Best solution value found. When the value of Best is equal to the corresponding BKS, it is presented in boldface.

Table 3
Details of the computers used in each published paper.

Author	Wang et al. (2020)	Damião et al. (2021)	Bruni et al. (2022a), and Nucamendi-Guillén et al. (2022)	Us
Computer	Intel Core i5-4210M @ 2.60 GHz	Intel Core i7-3770 @ 3.40 GHz	Intel Core i5-6300U @ 2.40 GHz	Intel Core i7-8700K @ 3.70 GHz
Single thread rating	1679	2071	1676	2750
Scaling factor	0.61	0.75	0.61	1

Table 4
Best configuration of parameters for each problem.

Problem	it_{max}	it_{soft}	it_{hard}	t_0	t_f	α	a
MDCCVRP	200	18	20	500	10	0.98	0.1
MDk-TRP	200	18	30	500	5	0.98	–
LLRP	200	15	20	400	10	0.90	5

- gap_B: Percentage gap between Best and BKS, computed as $gap_B = 100\frac{(Best-BKS)}{BKS}$.
- Avg: Average solution value computed over 30 runs for the PLS heuristic algorithm (see Wang et al. (2020)), and computed over 30, 10, and 5 runs for the M-ILS algorithm.
- $time$: Global computing time for finding the Best value (expressed in seconds).

3.1. Parameter tuning

For selecting the correct parameters of the M-ILS metaheuristic, the *iterated racing for automatic algorithm configuration* IRACE software has been used. IRACE is a well-known calibration tool that has been used successfully for tuning the parameters of different metaheuristic algorithms for several combinatorial optimization problems. Details about the elitist procedures applied by the software can be found in López-Ibáñez et al. (2016).

Because of the significant differences of the instances composing the considered benchmark data sets, the parameter tuning was performed separately for each of the three problems. For each problem, the training set is a sample of 1/3 of the corresponding instances of each data set. The values analyzed for each parameter were the following: it_{max}:{50, 100, 150, 200}, it_{soft}:{5, 10, 15, 18}, it_{hard}:{20, 25, 30, 40} (both it_{soft} and it_{hard} as a percentage of it_{max}), t_0:{100, 200, 300, 400, 500}, t_f:{0.5, 1, 5, 10}, α:{0.90, 0.95, 0.98, 0.99}, and a:{0.1, 0.3, 0.5, 1, 3, 5} as a percentage of the value of the initial solution s_0. The selected configurations for each problem are presented in Table 4.

3.2. An analysis of each ingredient of the M-ILS algorithm

This section presents an analysis regarding the quality of the solution obtained and the computing time required by each ingredient of the proposed algorithm. Furthermore, the efficiency of the local search procedures and the importance of each neighborhood structure are studied. This analysis is performed to evaluate the main contribution of each ingredient of the proposed approach to the quality of the solution concerning the objective function value and the computing time. For each problem, we have considered a set of unique parameters (described in the previous section) for analyzing the behavior of each ingredient of the proposed algorithm. It is to note that the global contribution of each ingredient remains even if the values of the parameters are changed.

Table 5 presents the global average results obtained by removing the different ingredients of the M-ILS algorithm and by executing 5 runs for each instance. The columns of the table correspond to the following values:

- M-ILS: The results obtained by the complete proposed algorithm.
- Initial Solution: The results obtained by the initial Constructive procedure, removing the ILS procedure (see Section 2.1).
- M-ILS (wLS): The results obtained by the proposed algorithm when the local search procedure (step 2 of the algorithm) is removed.
- M-ILS (wSA-VND): The results obtained by the proposed algorithm when the SA-VND procedure (step 3 of the algorithm) is removed.
- M-ILS (wLKH-3): The results obtained by the proposed algorithm when the LKH-3 procedure is removed from the first part of the perturbation $Configuration-Swap$ and from the final part of the Improvement phase.

In order to evaluate, for each problem and for each data set, the quality of the initial solution and the effect of removing step 2, step 3, and the LKH-3 procedure from the M-ILS algorithm, the following values (with the averages computed over all the corresponding instances) are considered.

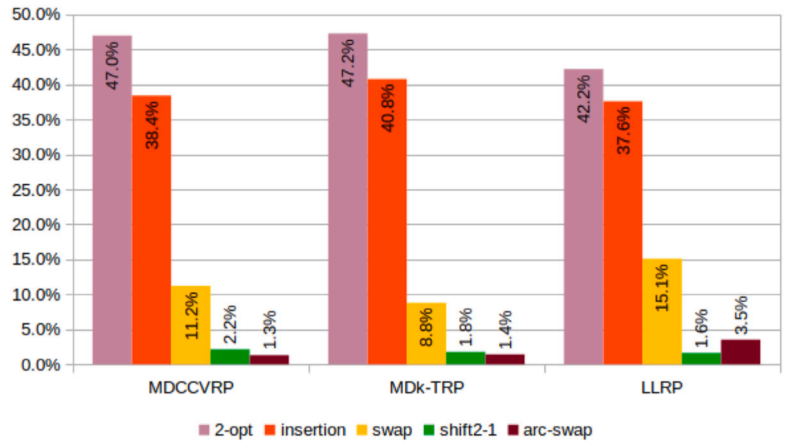
- $A-Best$: Average of the best solution values ($Best$) found by the considered algorithm.
- $A-Avg$: Average of the average solution values found for each run by the considered algorithm.
- $A-time$ (avg): Average of the average computing times required for each run by the considered algorithm.
- $A-gap_{B0}$: Average of the percentage gaps gap_{B0} between the values of $Best$ found by the considered algorithm and BKS_0, where BKS_0 represents the currently published best known solution value for the respective instance, with $gap_{B0} = 100\frac{(Best-BKS_0)}{BKS_0}$.
- $leq\ BKS_0$: The number of instances for which the best solution value $Best$ found by the considered algorithm is better than or equal to BKS_0.

The results reported in Table 5 show that the largest reduction of the computing time is achieved when the SA-VND procedure is removed. However, this also implies a considerable reduction of the solution quality. On the other hand, the results indicate that when the local search procedure LS (step 2) is removed, the computing time increases, generally without affecting considerably the quality of the solutions. These results suggest that the LS procedure helps to avoid extensive explorations during the execution of the SA-VND procedure. The results obtained by removing the LKH-3 procedure are worse than those obtained by the complete algorithm for all the data sets but the MDCCVRP data set lr, for which the results obtained by the two versions of the algorithm are similar. In all the cases, there is a reduction of the computing times; nevertheless, by considering the most complex data sets it is clear that by not considering the LKH-3 procedure the quality of the solution is negatively affected. Concerning the initial Constructive procedure, it is possible to note that it can provide reasonably good quality solutions in very short computing times; indeed, it can find solution values that are better than or equal to BKS_0 for 5 instance for the MDCCVRP, for 10 instances for the MDk-TRP, and for 13 instances for the LLRP. Globally, the best results are obtained when all the parts of the M-ILS algorithm are considered. This shows that all the ingredients of the proposed algorithm contribute to the final solution and must be considered.

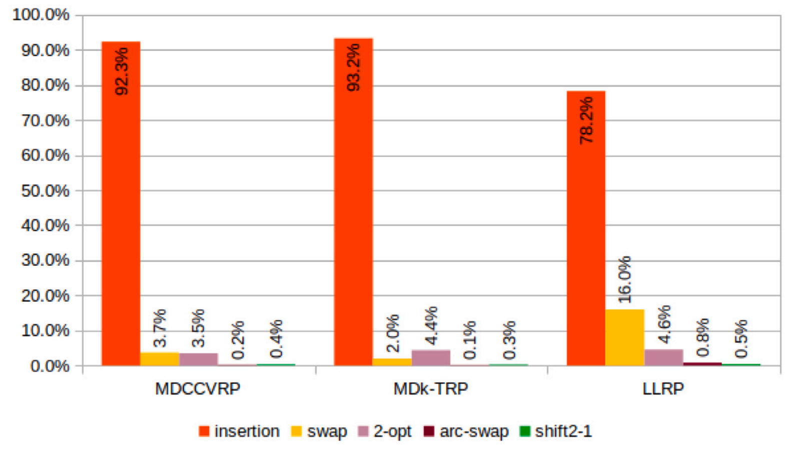
Another interesting analysis regards the importance of each neighborhood in the local search (LS) and VND procedures. Fig. 2 presents the percentage average contribution of each neighborhood for each

Table 5
Average results obtained, for each problem and data set, when different parts of the M-ILS algorithm are removed.

Data set	# Instances	M-ILS					Initial Solution					M-ILS (wLS)					M-ILS (wSA-VND)					M-ILS (wLKH-3)				
		A-Best	A-Avg	A-time (avg)	A-gap_{B0}	#leq BKS_0	A-Best	A-Avg	A-time (avg)	A-gap_{B0}	#leq BKS_0	A-Best	A-Avg	A-time (avg)	A-gap_{B0}	#leq BKS_0	A-Best	A-Avg	A-time (avg)	A-gap_{B0}	#leq BKS_0	A-Best	A-Avg	A-time (avg)	A-gap_{B0}	#leq BKS_0
MDCCVRP data sets																										
p-pr	33	**9705.07**	**9762.53**	138.5	**0.27**	**13**	9913.22	10 052.91	**15.7**	2.61	2	9714.97	9789.11	154.9	0.43	11	9773.40	9861.13	74.0	0.89	9	9735.35	9808.55	113.8	0.59	9
p-pr with $N_v = 35$	24	**5536.41**	**5547.76**	176.5	**0.17**	4	5564.21	5 588.54	**11.9**	0.66	2	5537.26	5549.31	184.0	0.20	**5**	5541.51	5555.14	61.7	0.26	2	5541.23	5553.11	138.4	0.24	3
lr	21	3830.01	3839.59	32.0	**0.08**	14	3869.47	3 875.47	**4.1**	1.66	1	3832.21	**3837.97**	31.7	0.11	11	3832.11	3844.69	18.1	0.10	10	**3829.87**	3839.45	17.5	**0.08**	**16**
All the MDCCVRP instances	78	**6840.66**	**6871.04**	121.5	**0.19**	**31**	6947.90	7 016.10	**11.4**	1.76	5	6845.70	6882.33	130.7	0.27	27	6871.70	6916.40	55.2	0.48	21	6854.91	6892.12	95.4	0.35	28
MDk-TRP data sets																										
p-pr with reduced fleet	24	**7270.43**	**7300.35**	141.8	**−0.03**	6	7342.18	7 418.98	**10.8**	1.27	2	7285.38	7316.95	156.3	0.14	**8**	7290.91	7330.29	55.1	0.30	4	7278.75	7314.82	121.1	0.08	6
p-pr with $N_v = 35$	24	**5529.99**	**5541.86**	188.2	**0.14**	5	5564.08	5 584.10	**17.1**	0.69	2	5534.07	5547.23	202.3	0.20	5	5539.96	5553.43	78.0	0.27	3	5535.04	5547.80	151.7	0.20	**7**
lr	21	**3830.41**	**3837.34**	39.3	**0.09**	**14**	3868.50	3 875.37	**6.8**	1.65	1	3832.76	3840.14	40.3	0.13	12	3833.68	3845.40	26.6	0.14	12	3831.33	3838.80	21.3	0.10	13
lr with reduced fleet	18	6367.18	**6394.93**	21.4	0.24	10	6526.54	6 575.17	**2.7**	2.30	5	**6359.69**	6397.76	21.6	**0.16**	**12**	6405.57	6450.80	14.9	0.76	8	6363.67	6398.89	14.8	0.20	10
All the MDk-TRP instances	87	**5773.08**	**5792.02**	108.1	**0.10**	35	5844.44	5 882.87	**9.9**	1.41	10	5777.35	5799.34	113.1	0.16	**37**	5790.21	5816.98	46.2	0.35	27	5776.27	5798.83	83.5	0.14	36
LLRP data sets																										
Tuzun-Burke	36	**3814.16**	**3840.57**	123.4	**−0.22**	22	3855.89	3 902.73	**17.3**	0.85	7	3816.82	3840.98	131.0	−0.16	**23**	3823.51	3849.32	106.6	0.01	15	3824.81	3853.05	76.8	0.04	18
Prodhon	30	1497.73	**1503.61**	76.4	**−0.08**	22	1511.46	1 527.72	**10.6**	1.09	5	**1497.29**	1503.80	77.1	**−0.08**	**23**	1498.71	1507.14	64.0	0.00	16	1501.09	1508.11	43.0	0.11	15
Barreto	10	9639.26	9815.04	39.3	**0.58**	**6**	9872.22	10 432.01	**4.0**	3.54	1	9616.07	**9774.02**	32.5	0.74	5	**9592.40**	9845.78	29.1	0.87	4	9740.23	9916.49	20.9	1.07	**6**
All the LLRP instances	76	3666.24	3704.20	92.8	**−0.06**	50	3722.08	3 824.34	**12.9**	1.30	13	**3664.28**	**3699.08**	96.8	−0.01	**51**	3664.89	3713.79	79.6	0.12	35	3685.89	3725.24	56.1	0.20	39



(a)



(b)

Fig. 2. Percentage average contribution of the neighborhoods. (a) LS. (b) VND.

problem. The contribution is measured in terms of the ratio of the number of times a move was applied, improving the current solution, over all the applied moves.

Regarding the LS procedure (see Fig. 2.a), the neighborhoods $2-opt$ and *insertion* correspond to those which produce the largest impact on the effectiveness of the proposed algorithm. On the other hand, due to the descent design of the VND procedure, the neighborhoods explored at the beginning give the largest contribution to the final solution (see Fig. 2.b), with the neighborhood *insertion* (which is the first neighborhood executed in the VND exploration), being the most applied move. As it is possible to note, all the neighborhoods contribute to the final solution in both procedures, even if some of them are not intensively applied.

The perturbations play a crucial role for the success of M-ILS. In order to evaluate the importance of the criterion based on the level of aggressiveness used in the perturbation step, three different versions of the M-ILS algorithm are compared with the original one. The considered versions are the following:
-P1: $Route - Swap$ is removed and replaced by $Route - Relocation$.
-P2: $Route - Relocation$ is removed and replaced by $Route - Swap$.
-P3: $Configuration - Swap$ is removed and replaced by $Route - Relocation$.

The comparison presented in Table 6 considers the same values defined at the beginning of this subsection. The results show that the original algorithm leads to the best global results regarding solution quality for the three problems. The version P1 leads to results similar to those obtained by the original version of M-ILS; nevertheless, P1 is clearly outperformed by the original M-ILS when the most complex instances are analyzed. Furthermore, in general, P1 requires larger computing times than the original version of the algorithm. The reason for the similar results obtained by the two versions is that the solution space does not change when the perturbation $Route - Swap$ is applied. With respect to P2, it is possible to note that also this version obtains results similar to those obtained by the original algorithm for the three problems in similar computing times; nevertheless, for the three problems, the original version is clearly more stable (see columns Avg and gap_{B0}). It is to note that the perturbation $Route - Relocation$ modifies the search space since the number of routes assigned to each used depot is changed; nevertheless, this search space can be potentially explored when new depot configurations are evaluated by applying $Configuration - Swap$, since the allocation of routes to depots is changed. Finally, the results show that P3 leads to the worst results in terms of solution quality among all the analyzed versions. By removing $Configuration - Swap$ it is possible to achieve a considerable reduction in the computing times; however, by avoiding an important part of the solution space associated with the used depots, worse-quality solutions are obtained.

3.3. The multi-depot cumulative capacitated vehicle routing problem (MDCCVRP)

There are four papers in the literature for the solution of the MDCCVRP: the POPMUSIC matheuristic (Lalla-Ruiz and Voß, 2020),

Table 6
Average results obtained, for each problem and data set, when different perturbations of the M-ILS algorithm are removed.

Data set	# Instances	M-ILS					P1					P2					P3				
		A-Best	A-Avg	A-time (avg)	A-gap_{B0}	#leq BKS_0	A-Best	A-Avg	A-time (avg)	A-gap_{B0}	#leq BKS_0	A-Best	A-Avg	A-time (avg)	A-gap_{B0}	#leq BKS_0	A-Best	A-Avg	A-time (avg)	A-gap_{B0}	#leq BKS_0
MDCCVRP data sets																					
p-pr	33	9705.07	**9762.53**	138.5	0.27	**13**	9743.62	9832.10	205.0	0.48	11	**9702.28**	9770.17	137.9	**0.25**	10	9731.71	9802.57	**123.3**	0.69	11
p-pr with $N_v = 35$	24	5536.41	**5547.76**	176.5	**0.17**	4	**5534.92**	5548.19	213.8	0.18	**5**	5536.53	5549.04	174.0	0.18	4	5540.44	5555.85	**151.2**	0.25	3
lr	21	3830.01	3839.59	32.0	0.08	**14**	**3829.50**	**3835.28**	31.0	**0.06**	**14**	3832.80	3839.30	31.6	0.15	11	3835.07	3844.56	**20.3**	0.19	10
All the MDCCVRP instances	78	6840.66	**6871.04**	121.5	**0.19**	**31**	6856.37	6899.45	160.9	0.27	30	**6840.27**	6874.59	120.4	0.20	25	6854.53	6891.81	**104.2**	0.42	24
MDk-TRP data sets																					
p-pr with reduced fleet	24	**7270.43**	**7300.35**	141.8	**−0.03**	6	7277.45	7309.53	182.8	0.08	6	7273.97	7300.96	142.0	0.02	**8**	7285.04	7313.76	**129.7**	0.19	7
p-pr with $N_v = 35$	24	**5529.99**	**5541.86**	188.2	**0.14**	**5**	5532.96	5545.72	226.9	0.19	5	5535.82	5546.76	193.5	0.24	4	5531.51	5545.77	**170.0**	0.17	4
lr	21	3830.41	**3837.34**	39.3	0.09	**14**	**3829.80**	3838.10	38.8	**0.06**	**14**	3834.46	3840.50	43.3	0.19	9	3834.07	3842.76	**25.6**	0.16	9
lr with reduced fleet	18	6367.18	6394.93	21.4	0.24	10	**6360.22**	**6377.35**	24.1	**0.17**	**11**	6368.72	6398.71	21.7	0.28	**11**	6380.28	6419.18	**13.8**	0.43	9
All the MDk-TRP instances	87	**5773.08**	**5792.02**	108.1	**0.10**	35	5774.25	5792.17	127.4	0.12	**36**	5776.96	5795.09	107.5	0.17	32	5781.12	5803.13	**91.7**	0.22	29
LLRP data sets																					
Tuzun-Burke	36	**3814.16**	**3840.57**	123.4	**−0.22**	22	3818.79	3843.25	142.1	−0.13	19	3815.69	3842.29	121.6	−0.20	**24**	3826.03	3865.87	**63.8**	0.02	17
Prodhon	30	**1497.73**	**1503.61**	76.4	**−0.08**	**22**	1498.19	1504.66	84.0	−0.05	19	1499.70	1505.99	76.3	0.01	21	1501.77	1510.83	**41.0**	0.24	14
Barreto	10	9639.26	9815.04	39.3	0.58	6	9605.93	**9695.45**	34.3	0.61	5	**9524.81**	9846.84	32.0	**0.56**	**7**	9649.27	9809.65	**12.9**	2.05	3
All the LLRP instances	76	3666.24	3704.20	92.8	**−0.06**	50	3664.23	**3690.15**	105.0	0.00	43	**3652.68**	3710.14	91.9	−0.02	**52**	3674.77	3718.32	**48.1**	0.37	34

the PLS heuristic algorithm (Wang et al., 2020), the branch-and-cut-and-price algorithm (BCP) (Damião et al., 2021), and the two MILP formulations presented in Nucamendi-Guillén et al. (2022). For the MDCCVRP, the M-ILS algorithm is executed for each instance with a number of runs equal to 30, 10 and 5.

Since in Wang et al. (2020) the heuristic algorithm PLS has been shown to be more effective than the matheuristic POPMUSIC in terms of both the solution values and the computing times, the latter algorithm is not considered in the following.

In Damião et al. (2021), the authors presented computational results for two configurations of the BCP, one obtained by fixing a small value for the maximum number of customers that can be visited in the same route (BCP$_{fix}$), and the other without fixing this value (BCP$_{nf}$). According to the results reported in Damião et al. (2021), BCP$_{fix}$ dominates the non-fixed version since it can find the optimal solution for all the instances solved to proven optimality by BCP$_{nf}$, but within shorter computing times. In addition, BCP$_{fix}$ can find feasible solutions for 14 instances for which BCP$_{nf}$ runs out of memory without finding a feasible solution. Of course, the solutions found by BCP $_{fix}$ for these 14 instances are not proved to be optimal.

In order to present a fair comparison, the global computing times reported in Tables 7–9 for each instance correspond to: (i) for PLS to the average computing time reported in Wang et al. (2020) multiplied by 30 (number of runs) and by the scaling factor (0.61); (ii) for BCP to the scaled computing times (considering a scaling factor equal to 0.75) reported for BCP$_{fix}$ in Damião et al. (2021), and (iii) for the two formulations to the scaled computing time (considering a scaling factor equal to 0.61) of the fastest between the two MILP models, as the computing time of each separate model was not reported in Nucamendi-Guillén et al. (2022). The computing times reported for M-ILS correspond to the average computing times multiplied by the respective number of runs. Furthermore, for each number of runs of M-ILS the following values are reported:

- gap_{PLS}: Percentage gap between the Best solution value found by M-ILS ($Best$) and the Best solution value found by PLS (Best$_{PLS}$), computed as $gap_{PLS} = 100\frac{(Best-Best_{PLS})}{Best_{PLS}}$.
- gap_{BCP}: Percentage gap between the Best solution value found by M-ILS and the Best solution value found by BCP (Best$_{BCP}$), computed as $gap_{BCP} = 100\frac{(Best-Best_{BCP})}{Best_{BCP}}$.

3.3.1. The p-pr data set

The computational results corresponding to the 33 instances of the data set p-pr are reported in Table 7. This data set contains the most challenging MDCCVRP instances due to the large number N_c of customers and the small number N_v of vehicles (generally, the smaller N_v, the more difficult is the instance Lalla-Ruiz and Voß, 2020; Damião et al., 2021). According to the results reported in Table 7, only the metaheuristic algorithms, i.e., PLS and M-ILS, can find a feasible solution for all the instances in this data set. BCP$_{fix}$ can find the proven optimal solution for 18 instances and the best-known feasible solution for 6 instances (no feasible solution is found for the remaining 9 instances).

For the 9 instances for which BCP$_{fix}$ runs out of memory without a feasible solution, M-ILS provides the new best-known solution value (outperforming the solution value provided by PLS), independently of the number of runs. The corresponding values of LB and gap_{LB} (where gap_{LB} is the percentage gap between BKS and LB, computed as $gap_{LB} = 100(\frac{(BKS-LB)}{LB})$ for these instances are the following: p08: 14 444.3 (19.58%), p09: 11 742.3 (27.045%), p11: 9883.96 (43.27%), p21: 21 397.1 (19.18%), p22: 20 822.2 (16.85%), p23: 20 247.4 (16.67%), pr05: 6436.69 (52.34%), pr06: 7434.05 (46.26%), and pr10: 7645.9 (48.25%). Furthermore, considering the 24 instances for which BCP$_{fix}$ provides the best-known solution value, M-ILS (executed for 30 and 10 runs) finds the optimal solution value for 7 instances, and the average percentage gap between the best solution value provided by M-ILS and BKS is equal to 0.29% when M-ILS is executed for 30 runs. The global average computing time required by BCP$_{fix}$ for solving these 24 instances is 1.6 times larger than that required by M-ILS (executed for 30 runs). No computing time has been reported in Damião et al. (2021) for the 9 instances for which BCP$_{fix}$ runs out of memory without finding a feasible solution. Therefore, it is impossible to compute the global average computing time (considering all the 33 instances) associated with this algorithm. The corresponding values of LB and gap_{LB} for the 6 instances for which BCP provides the best known feasible solution value are the following: p10: 10 478.5 (33.84%), p18: 13 535.5 (16.14%), p19: 13 097.9 (15.33%), p20: 12 672.8 (15.15%), pr04: 5559.22 (63.18%), and pr09: 5586.49 (61.60%). On the other hand, the MILP formulations can solve optimally only two small-size instances (with up to 50 customers) and provide, within the time limit, feasible solutions for 7 additional instances with up to 100 customers, with an average percentage value of gap_B equal to 2.48%. For the 9 instances for which the MILP formulations are able to provide a solution, the average percentage value of gap_F (where gap_F is the percentage gap between the Best solution value found by M-ILS and the Best solution value found by the formulations ($Best_F$), computed as $gap_F = 100(\frac{(Best-Best_F)}{Best_F})$ is equal to −2.18%, −2.13%, and −1.99% when M-ILS is executed for 30, 10 and 5 runs, respectively. Since the MILP formulations are dominated by BCP, the results associated with them are not reported in Table 7.

By comparing the best results provided by the heuristic algorithms PLS and M-ILS (both executed for 30 runs) on the 33 instances of this data set, it is possible to see that M-ILS provides better solutions than PLS for 27 instances, the same solution value for 4 instances and worse solution values only for 2 instances. The final average percentage value of gap_{PLS} equals −0.92%. However, it is to note that, although M-ILS provides better quality solutions than PLS, the computing times required by PLS are clearly smaller than those required by M-ILS.

For all the instances but one, the Avg. solution value provided by M-ILS is better than the Avg. solution value provided by PLS. Furthermore, for 12 instances, the Avg. solution value provided by M-ILS is better than the best solution value reported for PLS. The global average percentage gap between the average solution value provided by M-ILS and the best solution value provided by PLS is equal to 0.16%. This indicates that M-ILS is more stable than PLS, hence it needs fewer runs to provide good-quality solutions. Indeed, reducing the number of runs to 10 and 5, the number of instances for which M-ILS provides better solution values than those found by PLS equals 27 and 24, respectively. For 4 (resp. 3) instances, both heuristic algorithms found the same solution value when M-ILS is executed for 30 (resp. 10 and 5) runs. Globally, the gap_{PLS} value is equal to −0.72% and to −0.57% by considering 10 and 5 runs, respectively; this means that independently of the number of runs, M-ILS overcomes PLS in terms of solution quality. For 11 and 13 instances, the average solution value provided by M-ILS is better than the best value found by PLS when, respectively, 10 and 5 runs are considered for M-ILS. In addition, when M-ILS is executed for 5 runs, the average solution value provided by M-ILS is equal to the best solution value found by PLS for two instances. Thus, the reduction in the number of runs does not significantly affect the quality of the solutions provided by the proposed algorithm. In contrast, the global computing time of M-ILS is drastically reduced to very competitive ones with respect to those of PLS.

3.3.2. The p-pr data set with $N_v = 35$

The p-pr data set with $N_v = 35$ is composed of 24 instances, and the corresponding computational results are reported in Table 8. BCP$_{fix}$ can obtain a proven optimal solution for 16 instances and the best-known feasible solution for the remaining 8 instances. The MILP formulations cannot find a feasible solution for 9 large-size instances, and the largest-size instance that can be solved optimally considers 192 customers. The columns gap$_{BCP}$ and gap$_F$ are not reported in Table 8,

Table 7
Detailed results for the MDCCVRP p-pr data set.

Instance	N_c	N_d	N_v	BKS	PLS				BCP			M-ILS 30 runs						M-ILS 10 runs						M-ILS 5 runs					
					Best	Avg	Time	gap_B	Best	Time	gap_B	Best	Avg	Time	gap_B	gap_{PLS}	gap_{BCP}	Best	Avg	Time	gap_B	gap_{PLS}	gap_{BCP}	Best	Avg	Time	gap_B	gap_{PLS}	gap_{BCP}
p01	50	4	11	1 055.35	**1 055.35**	1 095.44	26.7	0.00	**1 055.35**	2.1	0.00	**1 055.35**	1 071.27	451.5	0.00	0.00	0.00	**1 055.35**	1 070.70	153.9	0.00	0.00	0.00	**1 055.35**	1071.85	77.3	0.00	0.00	0.00
p02	50	4	5	2 016.18	2 055.40	2 149.54	58.0	1.95	**2 016.18**	44.9	0.00	**2 016.18**	2 047.63	612.3	0.00	−1.91	0.00	**2 016.18**	2 047.35	202.8	0.00	−1.91	0.00	2 020.17	2 046.77	103.9	0.20	−1.71	0.20
p03	75	5	11	1 749.29	1 758.70	1 809.51	59.7	0.54	**1 749.29**	25.8	0.00	1 762.49	1 799.85	1 046.3	0.75	0.22	0.75	1 762.49	1 796.47	362.7	0.75	0.22	0.75	1 773.19	1 796.09	182.2	1.37	0.82	1.37
p04	100	2	15	2 537.59	2618.88	2 703.37	153.4	3.20	**2 537.59**	748.5	0.00	2 552.67	2 635.03	2 503.9	0.59	−2.53	0.59	2 591.08	2 640.48	744.9	2.11	−1.06	2.11	2 591.08	2 644.37	335.9	2.11	−1.06	2.11
p05	100	2	8	3 749.92	3 766.20	3 824.49	181.9	0.43	**3 749.92**	939.0	0.00	3 754.39	3 790.63	1 928.3	0.12	−0.31	0.12	3 754.39	3 783.05	672.8	0.12	−0.31	0.12	3 754.39	3 776.43	339.0	0.12	−0.31	0.12
p06	100	3	16	2 131.61	2 160.93	2 188.17	94.8	1.38	**2 131.61**	80.3	0.00	2 132.97	2 164.93	2 402.3	0.06	−1.29	0.06	2 143.65	2 164.89	802.9	0.56	−0.80	0.56	2 143.65	2 166.16	391.6	0.56	−0.80	0.56
p07	100	4	16	2 108.89	2 142.11	2 181.18	109.4	1.58	**2 108.89**	118.5	0.00	2 140.15	2 183.26	2 288.7	1.48	−0.09	1.48	2 141.06	2 180.23	780.4	1.53	−0.05	1.53	2 160.59	2 177.81	391.8	2.45	0.86	2.45
p08	249	2	25	17 272.00	17 393.46	17 862.13	644.2	0.70	–	–	–	**17 272.00**	17 516.46	7 061.0	0.00	−0.70	–	17 302.20	17 495.97	2 411.7	0.17	−0.52	–	17 302.20	17 414.82	1219.9	0.17	−0.52	–
p09	249	3	26	14 918.00	15 041.69	15 448.98	492.8	0.83	–	–	–	**14 918.00**	15 021.94	6719.0	0.00	−0.82	–	**14 918.00**	14 997.55	2 416.8	0.00	−0.82	–	**14 918.00**	14 976.50	1189.1	0.00	−0.82	–
p10	249	4	26	14 024.58	14 265.77	14 619.35	522.5	1.72	**14 024.58**	13 887.8	0.00	14 089.20	14 322.70	6 787.8	0.46	−1.24	0.46	14 237.00	14 366.55	2194.2	1.51	−0.20	1.51	14 237.00	14 341.92	1 141.0	1.51	−0.20	1.51
p11	249	5	26	14 161.20	14 381.43	14 552.90	518.8	1.56	–	–	–	**14 161.20**	14 404.70	6 523.5	0.00	−1.53	–	14 269.20	14 435.03	2 123.2	0.76	−0.78	–	14 289.70	14 409.52	1060.0	0.91	−0.64	–
p12	80	2	8	5 494.36	**5 494.36**	5 536.40	71.4	0.00	**5 494.36**	63.1	0.00	**5 494.36**	5 495.85	1 129.8	0.00	0.00	0.00	**5 494.36**	5 497.45	377.5	0.00	0.00	0.00	**5 494.36**	**5 494.36**	196.6	0.00	0.00	0.00
p13	80	2	9	4 914.66	**4 914.66**	4926.54	74.1	0.00	**4 914.66**	48.0	0.00	**4 914.66**	4 914.83	970.1	0.00	0.00	0.00	**4 914.66**	4 914.83	314.8	0.00	0.00	0.00	**4 914.66**	**4 914.66**	145.6	0.00	0.00	0.00
p14	80	2	10	4 491.64	4 510.12	4 512.28	71.0	0.41	**4 491.64**	44.6	0.00	**4 491.64**	4 492.13	894.3	0.00	−0.41	0.00	**4491.64**	4 492.59	299.3	0.00	−0.41	0.00	**4 491.64**	**4 491.64**	144.4	0.00	−0.41	0.00
p15	160	4	16	10 590.41	10 662.27	10 747.26	230.4	0.68	**10 590.41**	534.0	0.00	10 629.80	10 676.81	2 653.0	0.37	−0.30	0.37	10 646.80	10 683.73	885.2	0.53	−0.15	0.53	10 668.00	10 683.38	435.3	0.73	0.05	0.73
p16	160	4	17	10 008.28	10 086.50	10 122.13	196.0	0.78	**10 008.28**	624.8	0.00	10 016.10	10 070.56	2 548.3	0.08	−0.70	0.08	10 055.40	10 074.89	904.4	0.47	−0.31	0.47	10 055.40	10 073.96	456.8	0.47	−0.31	0.47
p17	160	4	18	9 493.84	9 538.69	9 573.86	194.0	0.47	**9 493.84**	313.5	0.00	9 495.91	9 518.63	2 465.7	0.02	−0.45	0.02	9512.48	9 522.32	841.4	0.20	−0.27	0.20	9 512.48	9 523.62	420.8	0.20	−0.27	0.20
p18	240	6	24	15 720.73	15 912.27	16 072.75	376.2	1.22	**15 720.73**	9 405.8	0.00	15 847.80	15 948.26	4 978.6	0.81	−0.41	0.81	15 847.80	15 940.73	1 670.6	0.81	−0.41	0.81	15 850.30	15 920.10	846.7	0.82	−0.39	0.82
p19	240	6	25	15 105.26	15 255.02	15 370.98	346.8	0.99	**15 105.26**	1 598.3	0.00	15 224.80	15 301.79	4 749.4	0.79	−0.20	0.79	15 224.80	15 293.53	1 522.0	0.79	−0.20	0.79	15 268.80	15 318.90	766.6	1.08	0.09	1.08
p20	240	6	26	14 592.52	14 709.23	14 786.87	333.8	0.80	**14 592.52**	1 582.5	0.00	14 635.70	14 708.73	4 736.2	0.30	−0.50	0.30	14 636.90	14 700.60	1 599.3	0.30	−0.49	0.30	14 649.00	14 699.60	802.0	0.39	−0.41	0.39
p21	360	9	34	25 500.80	25 770.35	26 102.29	725.0	1.06	–	–	–	**25 500.80**	25 892.53	10 982.1	0.00	−1.05	–	25 632.20	25 943.27	3 724.9	0.52	−0.54	–	25 632.20	25 878.44	1795.3	0.52	−0.54	–
p22	360	9	35	24 330.70	24 451.01	24 816.85	674.5	0.49	–	–	–	**24 330.70**	24 608.60	10 342.0	0.00	−0.49	–	**24 330.70**	24 606.22	3 433.2	0.00	−0.49	–	24 448.70	24 531.56	1676.4	0.48	−0.01	–
p23	360	9	36	23 622.80	23 656.13	23 925.16	598.0	0.14	–	–	–	**23 622.80**	23 784.41	9 638.3	0.00	−0.14	–	**23 622.80**	23 750.51	3 255.1	0.00	−0.14	–	23 636.50	23 754.00	1609.1	0.06	−0.08	–
pr01	48	4	4	3 748.11	**3 748.11**	3 773.27	61.7	0.00	**3 748.11**	38.8	0.00	3 768.69	3 768.69	444.5	0.55	0.55	0.55	3 768.69	3 768.69	149.8	0.55	0.55	0.55	3 768.69	3768.69	73.0	0.55	0.55	0.55
pr02	96	4	8	4 834.46	4 973.36	5 000.74	169.8	2.87	**4 834.46**	318.0	0.00	**4 834.46**	4 854.38	983.3	0.00	−2.79	0.00	**4 834.46**	4859.87	325.2	0.00	−2.79	0.00	4 846.47	4 864.04	168.2	0.25	−2.55	0.25
pr03	144	4	11	8 353.05	8 357.54	8 470.56	339.8	0.05	**8 353.05**	1 401.8	0.00	8 357.54	8 424.14	1 955.7	0.05	0.00	0.05	8383.72	8426.87	657.1	0.37	0.31	0.37	8 401.06	8 422.54	322.9	0.57	0.52	0.57
pr04	192	4	14	9 071.44	9 274.00	9 585.47	728.2	2.23	**9 071.44**	41 308.5	0.00	9156.38	9 324.65	4 431.6	0.94	−1.27	0.94	9 161.21	9 308.96	1 519.6	0.99	−1.22	0.99	9 161.21	9 256.52	717.8	0.99	−1.22	0.99
pr05	240	4	19	9 805.38	10 075.01	10 283.54	967.3	2.75	–	–	–	**9 805.38**	10 025.37	7 502.8	0.00	−2.68	–	9 861.20	10 078.00	2 434.5	0.57	−2.12	–	9 861.20	10 051.80	1221.3	0.57	−2.12	–
pr06	288	4	23	10 873.00	11 071.00	11 234.57	1043.1	1.82	–	–	–	**10 873.00**	10 996.31	8 934.9	0.00	−1.79	–	**10 873.00**	10 971.04	3 052.6	0.00	−1.79	–	**10 873.00**	10 993.04	1659.3	0.00	−1.79	–
pr07	72	6	6	4 760.65	4 877.86	4 906.62	120.4	2.46	**4 760.65**	207.0	0.00	**4 760.65**	4 787.68	731.8	0.00	−2.40	0.00	**4 760.65**	4 790.07	240.3	0.00	−2.40	0.00	4 770.53	4 795.37	119.3	0.21	−2.20	0.21
pr08	144	6	12	6 997.11	7 141.88	7 265.17	343.1	2.07	**6 997.11**	1 395.0	0.00	7 049.50	7 131.65	2 961.0	0.75	−1.29	0.75	7 049.50	7 116.86	971.2	0.75	−1.29	0.75	7 049.50	7 102.81	488.6	0.75	−1.29	0.75
pr09	216	6	17	9 027.82	9 219.95	9 350.34	675.8	2.13	**9 027.82**	21 477.0	0.00	9 147.07	9 327.58	5 598.4	1.32	−0.79	1.32	9 191.35	9 305.49	1 837.2	1.81	−0.31	1.81	9 191.35	9 295.98	922.4	1.81	−0.31	1.81
pr10	288	6	24	11 335.10	11 693.45	11 811.29	989.1	3.16	–	–	–	**11 335.10**	11 493.69	9 038.6	0.00	−3.06	–	**11 335.10**	11 531.23	3 040.6	0.00	−3.06	–	11 476.80	11 506.30	1438.4	1.25	−1.85	–
Global avg				9 648.39	9 758.57	9 897.27	**369.4**	1.23	–	–	–	**9 671.13**	**9772.90**	4151.4	**0.29**	**−0.92**	–	9 691.52	9 774.43	1391.6	0.49	−0.72	–	9 705.07	9762.53	692.7	0.64	−0.57	–
Global avg BCP				6 940.74	7 020.80	7 107.18	**230.8**	1.17	**6 940.74**	4 008.6	**0.00**	6 972.02	**7 031.74**	2 510.5	0.39	−0.76	0.39	6 986.48	7 031.13	834.6	0.59	−0.56	0.59	6 992.87	7 026.98	416.2	0.71	−0.44	0.71

Table 8
Detailed results for the MDCCVRP p-pr data set with $N_v = 35$.

Instance	N_c	N_d	BKS	PLS				BCP			M-ILS 30 runs					M-ILS 10 runs					M-ILS 5 runs				
				Best	Avg	Time	gap_B	Best	Time	gap_B	Best	Avg	Time	gap_B	gap_{PLS}	Best	Avg	Time	gap_B	gap_{PLS}	Best	Avg	Time	gap_B	gap_{PLS}
p01	50	4	712.50	713.18	717.44	41.0	0.10	**712.50**	1.9	0.00	**712.50**	714.18	1 930.2	0.00	−0.10	**712.50**	713.73	640.6	0.00	−0.10	712.74	714.09	319.3	0.03	−0.06
p02	50	4	712.50	713.59	716.78	40.6	0.15	**712.50**	3.8	0.00	**712.50**	713.65	1 875.5	0.00	−0.15	712.74	713.75	578.5	0.03	−0.12	712.74	713.73	251.5	0.03	−0.12
p03	75	5	950.25	952.17	959.03	61.7	0.20	**950.25**	11.6	0.00	**950.25**	953.90	1 859.2	0.00	−0.20	**950.25**	953.76	591.9	0.00	−0.20	**950.25**	953.49	292.3	0.00	−0.20
p04	100	2	1 955.31	1 955.82	1 959.14	95.5	0.03	**1 955.31**	38.0	0.00	**1 955.31**	1 955.97	3 646.5	0.00	−0.03	1 955.48	1 955.88	1238.7	0.01	−0.02	1 955.48	1 956.07	623.0	0.01	−0.02
p05	100	2	1 982.33	1 985.03	1 988.46	93.7	0.14	**1 982.33**	58.5	0.00	1 982.35	1 984.17	3 177.4	0.00	−0.14	1 982.35	1 983.94	1114.8	0.00	−0.14	1 982.35	1 983.57	600.1	0.00	−0.14
p06	100	3	1 552.13	1 553.88	1 563.22	83.6	0.11	**1 552.13**	28.0	0.00	**1 552.13**	1 553.88	3 164.8	0.00	−0.11	**1 552.13**	1 554.15	1122.5	0.00	−0.11	1 553.64	1 555.39	550.8	0.10	−0.02
p07	100	4	1 520.46	1 522.68	1 528.43	97.0	0.15	**1 520.46**	26.6	0.00	1 520.97	1 524.03	3 161.4	0.03	−0.11	1 520.97	1 522.96	1103.9	0.03	−0.11	1 520.97	1 521.73	523.4	0.03	−0.11
p08	249	2	15 372.60	15 410.92	15 458.51	374.4	0.25	**15 372.60**	937.5	0.00	**15 372.60**	15 400.51	7 956.0	0.00	−0.25	**15 372.60**	15 394.63	2640.7	0.00	−0.25	15 372.80	15 397.68	1422.1	0.00	−0.25
p09	249	3	13 070.74	13 136.24	13 335.06	340.4	0.50	**13 070.74**	812.3	0.00	13 071.60	13 105.79	7 521.8	0.01	−0.49	13 078.60	13 112.53	2446.3	0.06	−0.44	13 078.60	13 108.02	1172.1	0.06	−0.44
p10	249	4	12 052.56	12 096.90	12 432.72	341.1	0.37	**12 052.56**	735.0	0.00	12 070.50	12 155.06	7 577.5	0.15	−0.22	12 071.50	12 186.16	2581.9	0.16	−0.21	12 180.00	12 211.26	1315.3	1.06	0.69
p11	249	5	11 955.58	12 033.48	12 228.64	317.3	0.65	**11 955.58**	935.3	0.00	11 995.90	12 051.88	7 297.0	0.34	−0.31	12 007.20	12 052.14	2456.6	0.43	−0.22	12 007.20	12 057.88	1247.2	0.43	−0.22
p12	80	2	2 897.06	**2 897.06**	2 897.06	37.0	0.00	**2 897.06**	17.3	0.00	**2 897.06**	**2 897.06**	1 426.2	0.00	0.00	**2 897.06**	**2 897.06**	471.6	0.00	0.00	**2 897.06**	**2 897.06**	238.3	0.00	0.00
p15	160	4	5 794.11	**5794.11**	5 794.11	95.2	0.00	**5794.11**	171.8	0.00	**5794.11**	**5794.11**	2 987.3	0.00	0.00	**5794.11**	**5794.11**	990.4	0.00	0.00	**5794.11**	**5794.11**	484.8	0.00	0.00
p18	240	6	11 433.91	11 469.49	11 546.91	225.6	0.31	**11 433.91**	895.5	0.00	11 453.50	11 476.90	5 023.9	0.17	−0.14	11 454.00	11 476.24	1674.0	0.18	−0.14	11 454.00	11 471.48	829.0	0.18	−0.14
pr01	48	4	1 261.53	1 261.81	1 264.74	37.5	0.02	**1 261.53**	3.1	0.00	1 262.43	1 266.58	1 315.9	0.07	0.05	1 263.67	1 266.62	445.9	0.17	0.15	1 263.67	1 267.06	221.1	0.17	0.15
pr02	96	4	2 572.84	**2 572.84**	2 580.94	88.2	0.00	**2 572.84**	37.7	0.00	**2 572.84**	2 574.88	2 739.4	0.00	0.00	**2 572.84**	2 573.91	985.4	0.00	0.00	**2 572.84**	2 574.64	443.8	0.00	0.00
pr03	144	4	4 462.50	4 466.10	4 511.37	137.6	0.08	**4 462.50**	122.3	0.00	4 464.91	4 475.65	4 596.5	0.05	−0.03	4 464.91	4 473.63	1539.2	0.05	−0.03	4 464.91	4 474.82	765.4	0.05	−0.03
pr04	192	4	5 804.15	5 813.87	5 863.56	233.7	0.17	**5 804.15**	405.8	0.00	5 813.33	5 825.20	7 706.8	0.16	−0.01	5 813.87	5 825.33	2618.1	0.17	0.00	5 813.87	5 826.20	1309.2	0.17	0.00
pr05	240	4	7 120.22	7 157.06	7 225.66	372.0	0.52	**7 120.22**	946.5	0.00	7 122.06	7 146.94	11 480.7	0.03	−0.49	7 123.63	7 149.97	3848.5	0.05	−0.47	7 123.63	7 149.70	1954.0	0.05	−0.47
pr06	288	4	8 603.85	8 685.68	8 874.53	499.4	0.95	**8 603.85**	1732.5	0.00	8 607.46	8 657.63	13 211.5	0.04	−0.90	8 616.21	8 666.15	4299.8	0.14	−0.80	8 659.23	8 674.06	2166.5	0.64	−0.30
pr07	72	6	1 723.63	1 727.25	1 736.14	65.1	0.21	**1 723.63**	13.1	0.00	1 725.55	1 727.74	2 162.5	0.11	−0.10	1 725.55	1 728.32	746.6	0.11	−0.10	1 725.61	1 729.78	402.3	0.11	−0.09
pr08	144	6	4 004.11	4 023.21	4 046.68	150.1	0.48	**4 004.11**	118.5	0.00	**4 004.11**	4 015.78	4 411.3	0.00	−0.47	**4 004.11**	4 015.01	1487.3	0.00	−0.47	4 008.65	4 015.09	760.2	0.11	−0.36
pr09	216	6	5 889.02	5 937.19	6 043.29	268.6	0.82	**5 889.02**	512.3	0.00	5 899.64	5 932.25	8 134.3	0.18	−0.63	5 904.08	5 923.84	2741.1	0.26	−0.56	5 904.08	5 920.67	1352.2	0.26	−0.56
pr10	288	6	9 113.49	9 166.57	9 336.73	523.4	0.58	**9 113.49**	1555.5	0.00	9 135.86	9 177.17	11 912.7	0.25	−0.34	9 162.80	9 183.27	4035.7	0.54	−0.04	9 165.34	9 178.64	1932.6	0.57	−0.01
Global avg			5 521.56	5 543.59	5 608.71	**192.5**	0.28	**5 521.56**	421.7	**0.00**	5 527.06	5 545.04	5 261.5	0.07	−0.22	5 529.71	5 546.55	1766.7	0.10	−0.18	5 536.41	5 547.76	882.4	0.17	−0.11

Table 9
Detailed results for the MDCCVRP lr data set.

Instance	N_c	N_d	N_v	BKS	PLS				BCP			Formulations			M-ILS 30 runs					M-ILS 10 runs					M-ILS 5 runs				
					Best	Avg	Time	gap_B	Best	Time	gap_B	Best	Time	gap_B	Best	Avg	Time	gap_B	gap_{PLS}	Best	Avg	Time	gap_B	gap_{PLS}	Best	Avg	Time	gap_B	gap_{PLS}
lr1	10	4	5	545.69	**545.69**	554.97	2.7	0.00	**545.69**	0.0	0.00	**545.69**	0.0	0.00	**545.69**	546.13	13.4	0.00	0.00	**545.69**	**545.69**	3.4	0.00	0.00	**545.69**	**545.69**	2.0	0.00	0.00
lr2	10	4	5	832.69	**832.69**	847.80	2.6	0.00	**832.69**	0.0	0.00	**832.69**	0.0	0.00	**832.69**	**832.69**	38.2	0.00	0.00	**832.69**	**832.69**	12.8	0.00	0.00	**832.69**	**832.69**	6.4	0.00	0.00
lr3	10	4	5	832.78	**832.78**	841.47	2.6	0.00	**832.78**	0.0	0.00	**832.78**	0.0	0.00	**832.78**	**832.78**	21.8	0.00	0.00	**832.78**	**832.78**	7.5	0.00	0.00	**832.78**	**832.78**	3.3	0.00	0.00
lr4	25	4	10	2082.28	**2082.28**	2099.58	11.5	0.00	**2082.28**	0.8	0.00	**2082.28**	0.3	0.00	**2082.28**	2082.57	377.7	0.00	0.00	**2082.28**	2083.14	124.0	0.00	0.00	**2082.28**	2083.43	60.7	0.00	0.00
lr5	25	4	10	1827.41	**1827.41**	1870.90	11.9	0.00	**1827.41**	0.8	0.00	**1827.41**	0.3	0.00	**1827.41**	1837.30	335.0	0.00	0.00	**1827.41**	1836.14	111.3	0.00	0.00	**1827.41**	1834.39	57.9	0.00	0.00
lr6	25	4	10	1786.95	**1786.95**	1808.86	11.9	0.00	**1786.95**	0.7	0.00	**1786.95**	0.3	0.00	**1786.95**	**1786.95**	229.1	0.00	0.00	**1786.95**	**1786.95**	79.2	0.00	0.00	**1786.95**	**1786.95**	41.3	0.00	0.00
lr7	50	4	20	5424.57	**5424.57**	5440.37	49.8	0.00	**5424.57**	14.2	0.00	**5424.57**	3.0	0.00	**5424.57**	**5424.57**	1382.5	0.00	0.00	**5424.57**	**5424.57**	450.6	0.00	0.00	**5424.57**	**5424.57**	242.7	0.00	0.00
lr8	50	4	20	3737.38	**3737.38**	3759.67	45.6	0.00	**3737.38**	12.5	0.00	**3737.38**	4.1	0.00	**3737.38**	3743.96	713.0	0.00	0.00	**3737.38**	3741.14	196.9	0.00	0.00	**3737.38**	3742.68	96.6	0.00	0.00
lr9	50	4	20	3802.88	**3802.88**	3811.65	49.0	0.00	**3802.88**	11.3	0.00	**3802.88**	3.7	0.00	**3802.88**	3808.52	838.8	0.00	0.00	**3802.88**	3808.80	288.5	0.00	0.00	**3802.88**	3811.90	129.7	0.00	0.00
lr10-25V	50	6	25	2866.73	2868.39	2883.50	111.6	0.06	**2866.73**	8.9	0.00	**2866.73**	3.4	0.00	2867.28	2874.88	1009.5	0.02	−0.04	2869.43	2874.55	311.0	0.09	0.04	2870.00	2876.04	147.2	0.11	0.06
lr11-25V	50	6	25	2978.78	2987.75	3008.88	116.6	0.30	**2978.78**	10.1	0.00	**2978.78**	4.0	0.00	2979.46	2992.10	803.0	0.02	−0.28	2979.46	2993.20	281.9	0.02	−0.28	2985.06	2995.79	118.3	0.21	−0.09
lr12-25V	50	6	25	3090.38	3095.52	3112.60	112.5	0.17	**3090.38**	9.8	0.00	**3090.38**	3.8	0.00	**3090.38**	3095.64	896.8	0.00	−0.17	**3090.38**	3095.38	322.1	0.00	−0.17	3093.25	3096.55	161.4	0.09	−0.07
lr10-20V	50	6	20	2969.83	–	–	–	–	**2969.83**	9.6	0.00	–	–	–	**2969.83**	2992.04	1000.9	0.00	–	**2969.83**	2993.71	337.6	0.00	–	**2969.83**	2993.19	167.8	0.00	–
lr11-20V	50	6	20	3095.22	–	–	–	–	**3095.22**	14.9	0.00	–	–	–	**3095.22**	3120.25	545.4	0.00	–	3113.23	3126.52	170.8	0.58	–	3113.81	3129.93	88.4	0.60	–
lr12-20V	50	6	20	3171.75	–	–	–	–	**3171.75**	11.0	0.00	–	–	–	3174.98	3192.62	814.4	0.10	–	3184.43	3194.44	271.1	0.40	–	3188.09	3193.38	132.3	0.52	–
lr13	100	4	25	8288.43	8293.42	8331.27	111.8	0.06	**8288.43**	155.3	0.00	**8288.43**	78.8	0.00	**8288.43**	8324.16	1892.3	0.00	−0.06	**8288.43**	8331.18	629.4	0.00	−0.06	**8288.43**	8332.12	343.5	0.00	−0.06
lr14	100	4	25	7257.31	7273.03	7353.68	100.7	0.22	**7257.31**	158.3	0.00	**7257.31**	71.3	0.00	**7257.31**	7272.96	1925.0	0.00	−0.22	**7257.31**	7272.84	708.8	0.00	−0.22	**7257.31**	7275.67	346.4	0.00	−0.22
lr15	100	4	25	8626.13	8626.13	8644.64	133.0	0.00	**8626.13**	204.0	0.00	**8626.13**	49.5	0.00	**8626.13**	8643.77	2584.7	0.00	0.00	**8626.13**	8643.36	873.4	0.00	0.00	**8626.13**	8640.30	447.1	0.00	0.00
lr16	100	6	25	5265.30	5306.50	5483.36	128.8	0.78	**5265.30**	115.5	0.00	**5265.30**	66.1	0.00	**5265.30**	5281.53	1407.9	0.00	−0.78	**5265.30**	5278.95	475.4	0.00	−0.78	5268.68	5289.54	241.4	0.06	−0.71
lr17	100	6	25	6107.32	6141.07	6265.57	138.2	0.55	**6107.32**	132.0	0.00	**6107.32**	57.8	0.00	**6107.32**	6112.13	1638.5	0.00	−0.55	**6107.32**	**6107.32**	562.7	0.00	−0.55	**6107.32**	**6107.32**	274.7	0.00	−0.55
lr18	100	6	25	5788.73	5804.19	5905.71	143.3	0.27	**5788.73**	140.3	0.00	**5788.73**	66.3	0.00	**5788.73**	5811.51	1562.1	0.00	−0.27	5789.77	5811.17	499.3	0.02	−0.25	5789.77	5806.50	252.0	0.02	−0.25
Global avg				3827.55	–	–	–	–	**3827.55**	**48.1**	**0.00**	–	–	–	3827.76	3838.53	953.8	0.01	–	3829.22	3838.79	319.9	0.05	–	3830.01	3839.59	160.1	0.08	–
Global avg PLS/Formulations				3952.32	3959.37	4001.36	71.3	0.13	**3952.32**	54.1	**0.00**	**3952.32**	**22.9**	**0.00**	3952.39	3961.34	981.6	0.00	−0.13	3952.56	3961.10	329.9	0.01	−0.13	3953.25	3961.94	165.1	0.03	−0.11

since for all the instances the solution values provided by BCP$_{fix}$ are equal to BKS (so the value of gap_{BCP} is always equal to gap_B), and all the instances solved by the MILP formulations were solved to proven optimally (so the value of gap_F is equal to gap_B for these instances). Furthermore, by considering only the 15 instances for which the MILP formulations can provide the optimal solution, the global computing time required by the formulations is larger than that required by BCP$_{fix}$ (134.4 s > 70.5 s). Since the MILP formulations are dominated by BCP, the results associated with them are not reported in Table 8.

The global average value of gap_B obtained by M-ILS (with 30 runs) is equal to 0.07%. M-ILS finds the optimal solution value for 10 instances and near-optimal solution values (the largest gap_B value is equal to 0.34%) for all the remaining ones. Compared to PLS, M-ILS (with 30 runs) provides a better solution value for 20 instances, the same solution value for 3 instances, and a worse solution value for one instance. The global value of gap_{PLS} for M-ILS (with 30 runs) is equal to −0.22%, while the global average percentage gap between the Avg value provided by M-ILS and the best solution value obtained by PLS is equal to 0.04%. Regarding the global computing time, M-ILS presents large computing times when 30 runs are considered. However, when the number of runs is reduced to 10 or to 5, the quality of the solutions is not affected significantly, and the computing times decrease considerably. The global values of gap_B are equal to 0.10% and 0.17% when the number of runs is reduced to 10 and 5, respectively. M-ILS provides better solution values than those obtained by PLS for 19 and 18 instances when 10 and 5 runs are considered, respectively.

The average solution value provided by M-ILS is better than (for 21 instances) or equal to (for two instances) the average solution value provided by PLS, independently of the number of runs. Furthermore, for 8 and 7 instances, the average solution value provided by M-ILS considering 30 (or 5) and 10 runs, respectively, is better than the best value provided by PLS. For two instances, these values are equal, independently of the number of runs. The global average percentage gap between the value Avg provided by M-ILS and the best solution value obtained by PLS equals 0.05% and 0.07% when 10 and 5 runs are considered, respectively.

For two instances, the average solution value provided by M-ILS is equal to the optimal solution value independently of the number of runs. In addition, the global percentage gap between the average solution value provided by M-ILS and BKS is equal to 0.33% and 0.35%, when M-ILS is executed with 30 (or 10) and 5 runs, respectively. A single run may provide solution values close to BKS, making M-ILS competitive with respect to the global computing time.

3.3.3. *The lr data set*

The lr data set is composed of 21 instances and corresponds to the simplest data set due to the small number of customers and the relatively large size of the fleet. The corresponding computational results are reported in Table 9. This data set was proposed in Lalla-Ruiz and Voß (2020) as a small data set which allows the exact methods to find optimal solutions. Indeed, all the instances of this data set were solved to proven optimality in Damião et al. (2021). Besides, all the instances but three were also solved optimally in Nucamendi-Guillén et al. (2022). For the same reasons given in the previous Section (3.3.2), the columns gap_{BCP} and gap_F are not reported in Table 9.

By comparing the best results provided by the metaheuristic algorithms, it is possible to note that M-ILS (with 30 runs) provides solutions better than those found by PLS for 8 instances, and the same solution value for 10 instances. The value of gap_{PLS} is equal to −0.13%. Regarding the computing times, PLS is globally much faster than M-ILS. For all the instances, the Avg. solution value provided by M-ILS is better than the Avg. solution value provided by PLS. Furthermore, for 3 instances, the Avg. solution value provided by M-ILS is better than the best solution value reported for PLS. For 4 instances, these values are the same. The global average percentage gap between the average solution value provided by M-ILS and the best solution value provided by PLS is equal to 0.06%.

Also for this data set, the results indicate that M-ILS is more stable than PLS; hence, it needs fewer runs to provide good-quality solutions. Indeed, the number of instances for which M-ILS provides better solution values than those obtained by PLS is equal to 7 when the number of runs is reduced to 10 or to 5. Furthermore, for 10 instances, both algorithms found the same solution value independently of the number of runs executed by M-ILS.

Globally, the gap_{PLS} value is equal to −0.13% and −0.11% by considering 10 and 5 runs, respectively, which means that independently of the number of runs, M-ILS overcomes PLS in terms of solution quality. Furthermore, when the number of runs is equal to 10 (resp. 5), the number of instances for which the average solution value provided by M-ILS is better than the best solution value found by PLS is equal to 4 (resp. 2), and for 5 instances these values are equal when 10 or 5 runs are considered.

Comparing the results obtained by M-ILS versus the optimal solution values, it is possible to see that M-ILS can find the optimal solution value for 18, 16, and 14 instances, with a global value of gap_B equal to 0.01%, 0.05%, and 0.08%, when respectively, 30, 10, and 5 runs are executed. Furthermore, for 4 and 6 instances the average solution value obtained by M-ILS is equal to the optimal solution value by considering 30, and 10 (or 5) runs, respectively. In addition, the global percentage gap between the average solution value provided by M-ILS and the optimal solution value is equal to 0.27%, 0.28%, and 0.30% when the algorithm is executed with 30, 10, and 5 runs, respectively. A single run may provide near-optimal solution values, making M-ILS competitive in terms of computing time.

3.3.4. *Overall results on the MDCCVRP*

Analyzing the results, it is possible to state that the proposed algorithm M-ILS overcomes PLS in terms of solution quality for all the studied data sets by executing 30, 10, or 5 runs. The global average percentage gap_{PLS} value is equal to −0.51%, -0.41%, and −0.31% when M-ILS is executed with 30, 10, and 5 runs, respectively. As it is possible to note, the current MILP formulations (Nucamendi-Guillén et al., 2022), and the BCP algorithm (Damião et al., 2021) can manage small-size instances in reasonable computing times. Indeed, BCP$_{fix}$ can solve medium and large-size instances with large fleet sizes. Nevertheless, for the most challenging instances, the proposed M-ILS proved to be the most effective algorithm for what concerns the solution quality, providing the best results within competitive computing times with respect to PLS, and much shorter computing times with respect to BCP. As a consequence of the stability in the performance shown by M-ILS, the number of runs needed for obtaining good quality solutions is not large.

Regarding the proposed lower bounds, we found that the average value of gap_{LB} for the 23 instances not solved to proven optimally is equal to 26.48%. Despite this value is large, it does not mean that the BKS values for these instances correspond to bad quality solutions. Indeed, the average value of gap_{LB} obtained by considering only the instances solved to proven optimally is equal to 14.31%, which means that the proposed lower bounds are not tight. It is possible to note that the proposed lower bounds provide reasonable good approximations of the optimal solution value for instances with a relative large number of vehicles. The average value of gap_{LB} obtained by considering only the instances in the data sets p-pr with $N_v = 35$ and lr is equal to 6.40%.

In order to compare the efficiency of the algorithm M-ILS with that of the algorithm PLS, new experiments were carried out to compare the quality of the solutions obtained by both algorithms within the same global computing time, and to determine the computing time required by M-ILS to reach for each instance a given target value. Let us consider, for each instance, a target value (TV) given by the best solution value obtained by the algorithm PLS and a time limit (TL) given by the global computing time required by PLS to find the target value.

Table 10
Average results obtained by each metaheuristic considering time limits and target values for each MDCCVRP data set.

Data set	# Instances	PLS		M-ILS			
		TV	TL	Best$_{TL}$	gap_{TL}	#TV	time$_{TL}$
p-pr	33	9758.57	369.4	**9718.85**	**−0.32**	**21**	**205.5**
p-pr with $N_v = 35$	24	**5543.59**	192.5	5545.27	0.02	**15**	**155.5**
lr	18	**3959.37**	71.3	3960.42	0.00	**12**	**39.1**
All the instances	75	7017.97	241.3	**7001.28**	**−0.13**	**48**	**149.6**

The summary of the average results of this experiment is presented in Table 10 for each data set and for the overall set of instances. The reported columns correspond to: the averages of the target values (TV) and of the time limits (TL), the average of the best solution value found by M-ILS within the time limit (Best$_{TL}$), the average of the percentage gap between Best$_{TL}$ and TV (gap_{TV}), the number of instances for which the target value is found by M-ILS within the time limit (#TV), and the average of the computing times required by M-ILS to reach the target values (time$_{TV}$). It is to note that, for the instances for which M-ILS cannot find the target value, time$_{TV}$ is equal to the time limit.

The results show that M-ILS is able to find globally better results than PLS by considering the same global computing time for both algorithms (considering all the instances). It can be noted that M-ILS can find the target value (for 11 instances) or improve it (for 37 instances) for 48 out of the 75 considered instances within the time limit, obtaining a global average value of gap_{TV} equal to -0.13%. This means that M-ILS can find better quality solutions than those obtained by PLS within computing times slightly larger than half of the times reported for PLS. It is to note that M-ILS clearly dominates PLS for the p-pr data set, which contains the most challenging instances. For the other two data sets, both algorithms provide similar results, nevertheless, for each of the three considered data sets, M-ILS is able to find or improve the target value for more than 60% of the instances within the time limit.

As we proved in the previous sections, running the proposed metaheuristic for a longer time leads to better solutions than those obtained by the PLS algorithm. Nevertheless, this experiment also proved that in general the proposed M-ILS is superior to the current state-of-the-art metaheuristic when both algorithms compete with the same conditions.

3.4. The multi-depot k-traveling repairman problem

In this sections we compare the proposed M-ILS algorithm with the two mathematical formulations and the two configurations of the genetic algorithm (GA in the tables) presented in Bruni et al. (2022a), the only work in the literature dealing with the multi-depot k-traveling repairman problem. For the MDk-TRP, the M-ILS algorithm is executed for each instance with a number of runs equal to 10, 5 and 1. It is to note that both configurations of the algorithm GA are executed with a single run (the stopping criterion being defined by the maximum number of iterations, whose value is defined depending on the size of the instance.)

In order to present a fair comparison between M-ILS and the solution methods proposed in Bruni et al. (2022a), for each instance the global computing times presented in Tables 11–14 for the latter methods correspond to the scaled values (using a scaling factor equal to 0.61) of:

(i) the sum of the computing times reported in Bruni et al. (2022a) for each configuration of GA (since there is no dominance between the two configurations), and

(iia) the computing time reported in Bruni et al. (2022a) of the dominant formulation (Model 2) when the time limit is not reached, or (iib) the sum of the computing times of both formulations when the time limit is reached for Model 2. Furthermore, for each number of runs of M-ILS and each instance, also the following values are reported:

- gap_{GA}: Percentage gap between the Best solution value found by M-ILS ($Best$) and the Best solution value found by GA (Best$_{GA}$), computed as gap$_{GA} = 100\frac{(Best-Best_{GA})}{Best_{GA}}$.
- gap_F: Percentage gap between the Best solution value found by M-ILS and the Best solution value found by the Formulations (Best$_F$), computed as gap$_F = 100\frac{(Best-Best_F)}{Best_F}$.

3.4.1. The p-pr data set with reduced fleet

The computational results corresponding to the 24 instances of the p-pr data set with reduced fleet are reported in Table 11. This data set was proposed in Bruni et al. (2022a), and contains the most challenging MDk-TRP instances due to the large number of customers (N_c) and the small number of vehicles (N_v). According to the results presented in Table 11, M-ILS (executed with 10 and 5 runs) can find globally better solutions than those obtained by the formulations (with values of gap_F equal to −0.05 and −0.03, respectively) in shorter computing times (almost three and six times smaller, respectively). The maximum value of gap_B associated with M-ILS, executed with 10 or 5 runs, is equal to 2.15%, while the maximum value of gap_B associated with the formulations is equal to 8.52%. M-ILS, executed with 10 runs, can find the proven optimal solution value for 5 instances and provides new best-known solution values for 3 large instances. The corresponding values of LB and gap_{LB} for the 8 instances for which the formulations have not been solved to proven optimally are the following: p09: 11 742.3 (25.47%), p10: 10 478.5 (34.13%), p11: 9883.96 (41.61%), p15: 9048.53 (14.61%), p18: 13 535.5 (15.86%), pr05: 6436.69 (26.73%), pr06: 7434.05 (25.99%), and pr10: 7645.9 (29.09%).

Compared to GA, M-ILS is superior in terms of solution quality, improving the global best solution value of GA by over 7%, even when a single run is executed. M-ILS can find a better best solution value than that found by GA for all the instances but one, when 10 and 5 runs are executed, and for all the instances but 2 when a single run is performed. The average solution value obtained by M-ILS by executing 10 or 5 runs is better than the best solution value obtained by GA for all the instances but three. The global computing time required for executing M-ILS with 5 runs is slightly smaller than the global computing time required by GA. Nevertheless, the quality of the solutions provided by M-ILS is clearly better. In addition, when M-ILS is executed with a single run, the global computing time is 5 times smaller than the global computing time required by GA, and the quality of the solutions provided by M-ILS is still considerably better than that of GA.

3.4.2. The p-pr data set with $N_v = 35$

Table 12 presents the results for the p-pr data set with $N_v = 35$, obtained by the p-pr data set with the reduced fleet by setting $N_v = 35$ for all 24 instances. We found that for 10 instances of this data set, the values reported in Bruni et al. (2022a) as optimal/best solution values found by the formulations were smaller than the corresponding LB values.[1] The 10 instances with the respective values reported in Bruni et al. (2022a), and the corresponding LB values are the following: p01:

[1] We have jointly checked the results of the considered instances with the authors of Bruni et al. (2022a), who acknowledged an error in the procedure used to read the input files.

Table 11
Detailed results for the MD*k*-TRP p-pr data set with reduced fleet.

Instance	N_c	N_d	N_v	BKS	Formulations			GA			M-ILS 10 runs						M-ILS 5 runs						M-ILS 1 run				
					Best	Time	gap_B	Best	Time	gap_B	Best	Avg	Time	gap_B	gap_F	gap_{GA}	Best	Avg	Time	gap_B	gap_F	gap_{GA}	Best	Time	gap_B	gap_F	gap_{GA}
p01	50	4	5	1 958.95	**1 958.95**	77.9	0.00	2 852.25	1.0	45.60	**1958.95**	1970.08	196.3	0.00	0.00	−31.32	1959.16	1971.88	98.1	0.01	0.01	−31.31	1971.12	21.3	0.62	0.62	−30.89
p02	50	4	5	1 958.95	**1 958.95**	77.5	0.00	2 117.80	1.2	8.11	**1958.95**	1970.08	196.2	0.00	0.00	−7.50	1959.16	1971.88	98.1	0.01	0.01	−7.49	1971.12	21.2	0.62	0.62	−6.93
p03	75	5	8	2 271.22	**2 271.22**	1886.1	0.00	2 407.24	1864.3	5.99	2291.56	2301.25	454.5	0.90	0.90	−4.81	2291.56	2302.12	226.3	0.90	0.90	−4.81	2296.51	44.1	1.11	1.11	−4.60
p04	100	2	10	3 122.13	**3 122.13**	973.1	0.00	3 271.47	1090.9	4.78	3128.16	3138.33	578.8	0.19	0.19	−4.38	3128.16	3133.11	291.7	0.19	0.19	−4.38	3128.65	56.0	0.21	0.21	−4.37
p05	100	2	10	3 103.26	**3 103.26**	346.8	0.00	3 320.63	252.1	7.00	3110.70	3127.44	619.3	0.24	0.24	−6.32	3110.70	3126.47	316.1	0.24	0.24	−6.32	3113.37	55.9	0.33	0.33	−6.24
p06	100	3	10	2 870.05	**2 870.05**	501.0	0.00	3 126.83	602.8	8.95	2881.00	2895.27	639.4	0.38	0.38	−7.86	2881.00	2893.13	328.6	0.38	0.38	−7.86	2881.00	60.9	0.38	0.38	−7.86
p07	100	4	10	2 899.07	**2 899.07**	3833.4	0.00	3 087.86	2318.1	6.51	2927.51	2958.16	630.0	0.98	0.98	−5.19	2927.51	2956.66	313.7	0.98	0.98	−5.19	2972.98	69.6	2.55	2.55	−3.72
p08	249	2	25	16 620.90	**16 620.90**	3145.0	0.00	16 648.23	226.1	0.16	16 623.10	16 669.80	2263.9	0.01	0.01	−0.15	16 623.10	16 651.60	1126.8	0.01	0.01	−0.15	16 623.10	207.4	0.01	0.01	−0.15
p09	249	3	25	14 732.70	14 809.50	8784.0	0.52	14 816.49	502.5	0.57	**14 732.70**	14 786.37	2474.8	0.00	−0.52	−0.57	14 736.70	14 779.98	1277.9	0.03	−0.49	−0.54	14 793.40	225.1	0.41	−0.11	−0.16
p10	249	4	25	14 054.90	14 116.10	8784.0	0.44	14 255.40	538.2	1.43	**14 054.90**	14 088.27	2342.1	0.00	−0.43	−1.41	**14 054.90**	14 086.50	1217.8	0.00	−0.43	−1.41	14 177.50	275.2	0.87	0.43	−0.55
p11	249	5	25	13 996.20	15 189.20	8784.0	8.52	15 209.67	454.3	8.67	**13 996.20**	14 084.56	2428.0	0.00	−7.85	−7.98	14 009.00	14 101.48	1191.6	0.09	−7.77	−7.89	14 192.80	227.9	1.40	−6.56	−6.69
p12	80	2	8	5 479.51	**5 479.51**	3406.8	0.00	5 739.15	1100.1	4.74	**5479.51**	5480.73	400.8	0.00	0.00	−4.52	**5479.51**	5480.73	195.6	0.00	0.00	−4.52	**5479.51**	40.1	0.00	0.00	−4.52
p15	160	4	16	10 370.70	**10 370.70**	8784.0	0.00	15 834.43	1271.1	52.68	10 593.90	10 605.93	883.1	2.15	2.15	−33.10	10 593.90	10 605.72	455.1	2.15	2.15	−33.10	10 593.90	99.0	2.15	2.15	−33.10
p18	240	6	24	15 682.10	**15 682.10**	8784.0	0.00	16 884.47	715.6	7.67	15 702.30	15 770.61	1794.0	0.13	0.13	−7.00	15 702.30	15 750.56	912.3	0.13	0.13	−7.00	15 702.30	172.3	0.13	0.13	−7.00
pr01	48	4	5	3 036.43	**3 036.43**	164.7	0.00	3 245.12	3.4	6.87	**3036.43**	**3036.43**	185.4	0.00	0.00	−6.43	**3036.43**	**3036.43**	92.9	0.00	0.00	−6.43	**3036.43**	18.0	0.00	0.00	−6.43
pr02	96	4	10	4 092.51	**4 092.51**	1110.7	0.00	4 367.49	596.5	6.72	4110.50	4120.43	463.2	0.44	0.44	−5.88	4112.54	4117.15	249.0	0.49	0.49	−5.84	4112.54	53.1	0.49	0.49	−5.84
pr03	144	4	15	6 474.18	**6 474.18**	4334.5	0.00	7 024.86	1180.4	8.51	6482.60	6524.29	877.6	0.13	0.13	−7.72	6501.53	6531.16	419.7	0.42	0.42	−7.45	6562.09	80.3	1.36	1.36	−6.59
pr04	192	4	20	7 102.26	**7 102.26**	4252.0	0.00	7 687.69	745.9	8.24	7114.70	7143.20	1863.6	0.18	0.18	−7.45	7114.70	7140.08	936.0	0.18	0.18	−7.45	7114.70	185.5	0.18	0.18	−7.45
pr05	240	4	24	8 157.22	**8 157.22**	8784.0	0.00	9 061.75	397.1	11.09	8208.90	8253.24	3134.3	0.63	0.63	−9.41	8208.90	8261.97	1551.8	0.63	0.63	−9.41	8241.27	312.1	1.03	1.03	−9.05
pr06	288	4	29	9 366.42	**9 366.42**	8784.0	0.00	9 385.54	439.7	0.20	9384.92	9418.05	4001.0	0.20	0.20	−0.01	9384.92	9427.40	1983.8	0.20	0.20	−0.01	9417.88	385.9	0.55	0.55	0.34
pr07	72	6	8	3 496.68	**3 496.68**	246.5	0.00	3 781.00	140.3	8.13	**3496.68**	3519.80	292.3	0.00	0.00	−7.52	**3496.68**	3515.14	141.5	0.00	0.00	−7.52	**3496.68**	30.0	0.00	0.00	−7.52
pr08	144	6	15	5 906.88	**5 906.88**	3695.0	0.00	6 319.81	2550.7	6.99	5931.75	5995.08	1108.5	0.42	0.42	−6.14	5931.75	5995.11	560.6	0.42	0.42	−6.14	5931.75	114.7	0.42	0.42	−6.14
pr09	216	6	22	7 309.01	**7 309.01**	1724.2	0.00	8 128.12	351.8	11.21	7311.92	7391.88	2140.3	0.04	0.04	−10.04	7311.92	7400.36	1105.2	0.04	0.04	−10.04	7401.46	201.9	1.26	1.26	−8.94
pr10	288	6	29	9 869.74	**9 869.74**	8784.0	0.00	9 876.42	425.7	0.07	9923.98	9964.28	3756.2	0.55	0.55	0.48	9934.39	9971.67	1924.7	0.66	0.66	0.59	9969.99	373.4	1.02	1.02	0.95
Global avg				7 247.17	7 302.62	4168.6	0.40	7 852.07	740.4	9.62	**7268.4**	7300.6	1405.2	**0.32**	**−0.05**	**−7.59**	7270.4	**7300.3**	708.9	0.34	−0.03	−7.57	7299.3	**138.8**	0.71	0.34	−7.23

Table 12
Detailed results for the MDk-TRP p-pr data set with $N_v = 35$.

Instance	N_c	N_d	BKS	Formulations			GA			M-ILS 10 runs					M-ILS 5 runs					M-ILS 1 run			
				Best	Time	gap_B	Best	Time	gap_B	Best	Avg	Time	gap_B	gap_{GA}	Best	Avg	Time	gap_B	gap_{GA}	Best	Time	gap_B	gap_{GA}
p01	50	4	712.50	**712.50**	0.7	0.00	959.12	0.0	34.61	**712.50**	714.00	717.3	0.00	−25.71	712.52	713.76	317.4	0.00	−25.71	714.78	77.4	0.32	−25.48
p02	50	4	712.50	**712.50**	0.7	0.00	**712.50**	0.0	0.00	**712.50**	714.00	717.3	0.00	0.00	712.52	713.76	317.4	0.00	0.00	714.78	77.5	0.32	0.32
p03	75	5	950.25	**950.25**	6.0	0.00	958.69	6.9	0.89	**950.25**	954.02	722.9	0.00	−0.88	**950.25**	954.41	353.8	0.00	−0.88	957.51	51.9	0.76	−0.12
p04	100	2	1 955.31	**1 955.31**	17.5	0.00	1 970.90	31.1	0.80	**1 955.31**	1 955.94	1291.6	0.00	−0.79	**1 955.31**	1 955.82	649.0	0.00	−0.79	**1 955.31**	127.2	0.00	−0.79
p05	100	2	1 982.33	**1 982.33**	14.4	0.00	2 002.56	16.3	1.02	1 982.77	1 984.31	1230.6	0.02	−0.99	1 982.77	1 984.12	651.0	0.02	−0.99	1 984.80	136.6	0.12	−0.89
p06	100	3	1 551.64	**1 551.64**	17.8	0.00	1 587.70	28.5	2.32	1 552.15	1 554.31	1041.8	0.03	−2.24	1 552.15	1 554.40	524.8	0.03	−2.24	1 553.88	127.8	0.14	−2.13
p07	100	4	1 520.46	**1 520.46**	16.0	0.00	1 546.16	29.6	1.69	1 521.19	1 522.41	1281.5	0.05	−1.61	1 521.19	1 522.28	662.6	0.05	−1.61	1 521.60	135.9	0.07	−1.59
p08	249	2	15 368.20	**15 368.20**	1815.9	0.00	16 220.42	268.9	5.55	15 378.70	15 394.01	3210.7	0.07	−5.19	15 387.70	15 393.34	1651.1	0.13	−5.13	15 392.80	340.4	0.16	−5.10
p09	249	3	13 047.60	**13 047.60**	1446.3	0.00	14 129.02	459.9	8.29	13 073.60	13 082.89	2895.6	0.20	−7.47	13 077.50	13 085.88	1457.8	0.23	−7.44	13 081.60	256.6	0.26	−7.41
p10	249	4	12 037.50	**12 037.50**	1661.5	0.00	13 087.11	427.7	8.72	**12 037.50**	12 145.70	2789.5	0.00	−8.02	12 067.50	12 152.66	1406.6	0.25	−7.79	12 211.10	266.3	1.44	−6.69
p11	249	5	11 932.70	**11 932.70**	1315.8	0.00	13 282.14	372.2	11.31	11 967.10	12 014.93	2720.4	0.29	−9.90	11 967.10	12 014.54	1359.1	0.29	−9.90	12 053.50	271.1	1.01	−9.25
p12	80	2	2 897.06	**2 897.06**	3.8	0.00	**2 897.06**	5.3	0.00	**2 897.06**	**2 897.06**	574.4	0.00	0.00	**2 897.06**	**2 897.06**	288.4	0.00	0.00	**2 897.06**	59.5	0.00	0.00
p15	160	4	5 794.11	**5 794.11**	61.5	0.00	8 563.86	325.0	47.80	**5 794.11**	**5 794.11**	1237.5	0.00	−32.34	**5 794.11**	**5 794.11**	617.4	0.00	−32.34	**5 794.11**	122.9	0.00	−32.34
p18	240	6	11 433.90	**11 433.90**	8784.0	0.00	12 033.14	622.6	5.24	11 457.10	11 479.31	1921.9	0.20	−4.79	11 457.10	11 473.96	976.9	0.20	−4.79	11 478.90	182.3	0.39	−4.61
pr01	48	4	1 261.53	**1 261.53**	0.8	0.00	–	–	–	1 262.43	1 267.78	452.4	0.07	–	1 262.43	1 266.64	203.0	0.07	–	1 275.76	20.8	1.13	–
pr02	96	4	2 572.84	**2 572.84**	12.8	0.00	2 576.27	36.9	0.13	**2 572.84**	2 574.36	998.2	0.00	−0.13	**2 572.84**	2 574.35	529.0	0.00	−0.13	**2 572.84**	102.9	0.00	−0.13
pr03	144	4	4 462.50	**4 462.50**	76.2	0.00	4 636.24	80.1	3.89	4 473.50	4 478.26	1541.4	0.25	−3.51	4 473.50	4 478.61	749.3	0.25	−3.51	4 477.15	141.2	0.33	−3.43
pr04	192	4	5 804.15	**5 804.15**	258.1	0.00	6 026.25	310.6	3.83	5 810.69	5 824.67	2725.3	0.11	−3.58	5 810.69	5 822.00	1379.0	0.11	−3.58	5 821.34	256.8	0.30	−3.40
pr05	240	4	7 119.35	**7 119.35**	1160.8	0.00	7 527.32	296.1	5.73	7 121.18	7 147.74	4076.8	0.03	−5.40	7 135.82	7 159.51	2015.1	0.23	−5.20	7 163.66	390.6	0.62	−4.83
pr06	288	4	8 595.64	**8 595.64**	1852.9	0.00	9 492.53	405.8	10.43	8 638.02	8 661.39	4718.7	0.49	−9.00	8 646.44	8 672.22	2376.9	0.59	−8.91	8 722.22	522.8	1.47	−8.11
pr07	72	6	1 723.63	**1 723.63**	3.9	0.00	–	–	–	1 725.61	1 729.03	917.7	0.11	–	1 725.61	1 729.79	457.7	0.11	–	1 729.09	100.4	0.32	–
pr08	144	6	4 004.11	**4 004.11**	70.4	0.00	4 077.16	145.9	1.82	4 007.90	4 015.10	1684.2	0.09	−1.70	4 015.45	4 018.61	857.9	0.28	−1.51	4 016.36	201.8	0.31	−1.49
pr09	216	6	5 889.02	**5 889.02**	432.4	0.00	6 291.82	297.9	6.84	5 891.65	5 923.69	2991.7	0.04	−6.36	5 891.65	5 913.58	1484.1	0.04	−6.36	5 906.75	284.3	0.30	−6.12
pr10	288	6	9 108.08	**9 108.08**	8784.0	0.00	17 574.79	399.8	92.96	9 150.65	9 167.29	4656.6	0.47	−47.93	9 150.65	9 159.22	2336.6	0.47	−47.93	9 158.48	465.9	0.55	−47.89
Global avg			5 518.20	**5 518.20**	1158.9	**0.00**	–	–	–	5 526.93	5 541.51	1963.18	0.11	–	5 529.99	5 541.86	984.24	0.14	–	5 548.14	**196.70**	0.43	–
Global avg GA			5 884.17	**5 884.17**	1264.1	**0.00**	6 734.22	**207.60**	11.54	5 893.56	5 909.07	2079.38	0.11	**−8.07**	5 896.90	5 909.46	1043.69	0.14	−8.03	5 915.93	209.08	0.40	−7.80

Table 13
Detailed results for the MD*k*-TRP lr data set.

Instance	N_c	N_d	N_v	BKS	Formulations			GA			M-ILS 10 runs					M-ILS 5 runs					M-ILS 1 run			
					Best	Time	gap_B	Best	Time	gap_B	Best	Avg	Time	gap_B	gap_{GA}	Best	Avg	Time	gap_B	gap_{GA}	Best	Time	gap_B	gap_{GA}
lr1	10	4	5	545.69	**545.69**	0.6	0.00	**545.69**	0.0	0.00	**545.69**	**545.69**	10.6	0.00	0.00	**545.69**	**545.69**	5.3	0.00	0.00	**545.69**	0.7	0.00	0.00
lr2	10	4	5	832.69	**832.69**	0.1	0.00	**832.69**	0.0	0.00	**832.69**	**832.69**	29.8	0.00	0.00	**832.69**	**832.69**	14.6	0.00	0.00	**832.69**	3.0	0.00	0.00
lr3	10	4	5	832.78	**832.78**	0.3	0.00	**832.78**	0.0	0.00	**832.78**	**832.78**	16.5	0.00	0.00	**832.78**	**832.78**	8.2	0.00	0.00	**832.78**	1.6	0.00	0.00
lr4	25	4	10	2082.28	**2082.28**	0.1	0.00	**2082.28**	0.0	0.00	**2082.28**	2089.09	206.8	0.00	0.00	**2082.28**	2091.49	103.1	0.00	0.00	2109.15	23.2	1.29	1.29
lr5	25	4	10	1827.41	**1827.41**	0.1	0.00	**1827.41**	0.0	0.00	**1827.41**	1841.37	236.3	0.00	0.00	**1827.41**	1837.88	115.9	0.00	0.00	1844.86	22.3	0.95	0.95
lr6	25	4	10	1786.95	**1786.95**	0.1	0.00	**1786.95**	0.0	0.00	**1786.95**	**1786.95**	138.3	0.00	0.00	**1786.95**	**1786.95**	70.6	0.00	0.00	**1786.95**	14.5	0.00	0.00
lr7	50	4	20	5424.57	**5424.57**	1.9	0.00	5444.57	0.0	0.37	**5424.57**	**5424.57**	536.5	0.00	−0.37	**5424.57**	**5424.57**	259.1	0.00	−0.37	**5424.57**	40.7	0.00	−0.37
lr8	50	4	20	3737.38	**3737.38**	2.5	0.00	3760.03	0.0	0.61	**3737.38**	3745.99	299.8	0.00	−0.60	**3737.38**	3743.70	152.7	0.00	−0.60	**3737.38**	27.1	0.00	−0.60
lr9	50	4	20	3802.88	**3802.88**	2.5	0.00	3812.65	0.0	0.26	**3802.88**	3807.09	393.6	0.00	−0.26	3803.00	3810.07	214.8	0.00	−0.25	3803.00	44.1	0.00	−0.25
lr10-25V	50	6	25	2866.73	**2866.73**	1.5	0.00	–	–	–	2870.57	2875.00	415.9	0.13	–	2870.57	2876.24	176.2	0.13	–	2872.77	33.8	0.21	–
lr11-25V	50	6	25	2978.78	**2978.78**	1.3	0.00	–	–	–	2980.63	2989.29	428.8	0.06	–	2981.31	2990.45	199.8	0.08	–	2989.62	30.9	0.36	–
lr12-25V	50	6	25	3090.38	**3090.38**	1.5	0.00	–	–	–	**3090.38**	3094.37	420.1	0.00	–	**3090.38**	3093.64	206.8	0.00	–	3092.76	41.7	0.08	–
lr10-20V	50	6	20	2969.83	**2969.83**	1.3	0.00	2986.69	0.0	0.57	2972.04	2991.55	551.5	0.07	−0.49	2989.60	2996.12	275.4	0.67	0.10	2999.53	46.5	1.00	0.43
lr11-20V	50	6	20	3095.22	**3095.22**	2.1	0.00	3107.61	0.0	0.40	3103.22	3116.61	286.4	0.26	−0.14	3103.22	3108.71	164.9	0.26	−0.14	3113.81	33.1	0.60	0.20
lr12-20V	50	6	20	3171.75	**3171.75**	2.0	0.00	3186.67	0.0	0.47	3189.57	3194.18	419.4	0.56	0.09	3189.57	3191.56	210.8	0.56	0.09	3192.75	38.1	0.66	0.19
lr13	100	4	25	8288.43	**8288.43**	37.8	0.00	8563.04	21.4	3.31	8297.34	8310.99	774.3	0.11	−3.10	8297.34	8309.47	351.5	0.11	−3.10	8324.77	69.3	0.44	−2.78
lr14	100	4	25	7257.31	**7257.31**	30.5	0.00	7519.98	23.7	3.62	**7257.31**	7270.42	759.2	0.00	−3.49	**7257.31**	7265.17	366.6	0.00	−3.49	**7257.31**	82.6	0.00	−3.49
lr15	100	4	25	8625.13	**8625.13**	26.8	0.00	8803.73	23.2	2.07	**8625.13**	8643.69	956.7	0.00	−2.03	**8625.13**	8645.55	467.7	0.00	−2.03	8670.41	82.8	0.52	−1.51
lr16	100	6	25	5265.30	**5265.30**	28.9	0.00	5399.68	26.4	2.55	**5265.30**	5278.97	620.9	0.00	−2.49	**5265.30**	5288.72	326.7	0.00	−2.49	5268.68	77.9	0.06	−2.43
lr17	100	6	25	6107.32	**6107.32**	28.7	0.00	6353.43	24.3	4.03	**6107.32**	**6107.32**	676.8	0.00	−3.87	**6107.32**	**6107.32**	323.8	0.00	−3.87	**6107.32**	82.3	0.00	−3.87
lr18	100	6	25	5788.73	**5788.73**	30.5	0.00	6017.02	21.0	3.94	**5788.73**	5811.32	756.8	0.00	−3.79	**5788.73**	5805.32	391.3	0.00	−3.79	**5788.73**	70.8	0.00	−3.79
Global avg				3827.50	**3827.50**	**9.6**	**0.00**	–	–	–	3829.53	3837.61	425.48	0.06	–	3830.41	3837.34	209.80	0.09	–	3837.88	41.27	0.29	–
Global avg GA				3968.98	**3968.98**	10.9	**0.00**	4047.94	**7.77**	1.23	3971.03	3979.51	426.12	0.06	**−1.14**	3972.01	3979.10	212.39	0.09	−1.11	3980.02	42.24	0.31	−0.89

Table 14
Detailed results for the MDk-TRP lr data set with reduced fleet.

Instance	N_c	N_d	N_v	BKS	Formulations			GA			M-ILS 10 runs					M-ILS 5 runs					M-ILS 1 run			
					Best	Time	gap_B	Best	Time	gap_B	Best	Avg	Time	gap_B	gap_{GA}	Best	Avg	Time	gap_B	gap_{GA}	Best	Time	gap_B	gap_{GA}
lr1	10	4	4	592.27	**592.27**	0.0	0.00	**592.27**	0.0	0.00	**592.27**	**592.27**	19.9	0.00	0.00	**592.27**	**592.27**	10.0	0.00	0.00	**592.27**	1.5	0.00	0.00
lr2	10	4	4	885.89	**885.89**	0.0	0.00	**885.89**	0.0	0.00	**885.89**	**885.89**	23.8	0.00	0.00	**885.89**	**885.89**	12.0	0.00	0.00	**885.89**	2.4	0.00	0.00
lr3	10	4	4	846.91	**846.91**	0.1	0.00	**846.91**	0.0	0.00	**846.91**	**846.91**	24.2	0.00	0.00	**846.91**	**846.91**	12.2	0.00	0.00	**846.91**	2.4	0.00	0.00
lr4	25	4	4	3 267.66	**3 267.66**	2.9	0.00	–	–	–	3 300.74	3 300.74	70.9	1.01	–	3 300.74	3 300.74	36.0	1.01	–	3 300.74	7.3	1.01	–
lr5	25	4	4	3 206.83	**3 206.83**	1.5	0.00	3 267.66	0.0	1.90	**3 206.83**	**3 206.83**	79.1	0.00	−1.86	**3 206.83**	**3 206.83**	39.4	0.00	−1.86	**3 206.83**	8.6	0.00	−1.86
lr6	25	4	4	2 965.13	**2 965.13**	3.3	0.00	2 965.14	0.0	0.00	**2 965.13**	2 975.84	52.9	0.00	0.00	**2 965.13**	**2 965.13**	27.3	0.00	0.00	**2 965.13**	5.0	0.00	0.00
lr7	50	4	5	8 325.97	**8 325.97**	332.4	0.00	8 371.32	3.9	0.54	**8 325.97**	**8 325.97**	195.2	0.00	−0.54	**8 325.97**	**8 325.97**	95.5	0.00	−0.54	**8 325.97**	17.9	0.00	−0.54
lr8	50	4	5	7 089.64	**7 089.64**	4099.1	0.00	7 143.59	21.5	0.76	**7 089.64**	**7 089.64**	151.6	0.00	−0.76	**7 089.64**	**7 089.64**	76.9	0.00	−0.76	**7 089.64**	16.4	0.00	−0.76
lr9	50	4	5	7 390.97	**7 390.97**	732.4	0.00	7 417.23	2.3	0.36	**7 390.97**	7 434.33	167.8	0.00	−0.35	**7 390.97**	7 446.88	85.1	0.00	−0.35	7 516.49	18.4	1.70	1.34
lr10	50	6	6	6 143.47	**6 143.47**	427.1	0.00	6 598.34	0.3	7.40	**6 143.47**	6 154.12	190.4	0.00	−6.89	6 151.12	6 157.07	97.9	0.12	−6.78	6 151.12	19.3	0.12	−6.78
lr11	50	6	6	5 756.81	**5 756.81**	214.9	0.00	6 168.67	4.7	7.15	5 785.93	5 817.54	171.8	0.51	−6.20	5 785.93	5 809.85	87.6	0.51	−6.20	5 849.86	18.5	1.62	−5.17
lr12	50	6	6	5 699.16	**5 699.16**	85.3	0.00	5 730.72	1.9	0.55	**5 699.16**	5 706.05	147.2	0.00	−0.55	**5 699.16**	5 705.44	74.1	0.00	−0.55	**5 699.16**	14.0	0.00	−0.55
lr13	100	4	10	11 744.60	**11 744.60**	3279.2	0.00	12 161.63	1563.5	3.55	11 777.50	11 890.21	341.5	0.28	−3.16	11 812.50	11 896.26	162.8	0.58	−2.87	11 963.00	29.5	1.86	−1.63
lr14	100	4	10	10 742.60	**10 742.60**	654.6	0.00	11 160.95	355.4	3.89	**10 742.60**	10 827.05	522.8	0.00	−3.75	10 760.00	10 821.74	257.8	0.16	−3.59	10 799.50	52.3	0.53	−3.24
lr15	100	4	10	11 035.60	**11 035.60**	159.7	0.00	12 033.57	2261.5	9.04	**11 035.60**	11 049.89	478.1	0.00	−8.29	**11 035.60**	11 064.18	232.6	0.00	−8.29	**11 035.60**	46.9	0.00	−8.29
lr16	100	6	10	9 442.62	**9 442.62**	3537.5	0.00	9 685.37	1657.4	2.57	9 574.04	9 657.29	350.0	1.39	−1.15	9 574.04	9 636.79	186.5	1.39	−1.15	9 680.04	31.8	2.51	−0.06
lr17	100	6	10	9 917.56	**9 917.56**	1303.2	0.00	10 343.79	859.9	4.30	9 942.78	10 048.18	554.4	0.25	−3.88	9 945.93	10 052.49	271.9	0.29	−3.85	10 069.60	64.4	1.53	−2.65
lr18	100	6	10	9 220.77	**9 220.77**	3074.7	0.00	9 604.71	1251.3	4.16	9 240.67	9 361.50	435.8	0.22	−3.79	9 240.67	9 304.71	215.7	0.22	−3.79	9 277.06	48.8	0.61	−3.41
Global avg				6 348.58	**6 348.58**	994.9	**0.00**	–	–	–	6 363.67	6 398.35	220.96	0.20	–	6 367.18	6 394.93	110.07	0.24	–	6 403.05	**22.53**	0.64	–
Global avg GA				6 529.81	**6 529.81**	1053.2	**0.00**	6763.40	469.6	2.72	6 543.85	6 580.56	229.78	0.16	−2.42	6 547.56	6 576.94	114.43	0.19	−2.39	6 585.53	**23.43**	0.62	−1.98

660.34 < 707.68, p02: 660.34 < 707.68, p04: 1881.48 < 1926.16, p05: 1871.62 < 1956.75, p06: 1460.6 < 1500.48, p12: 2769.07 < 2897.06, p15: 5618.84 < 5794.11, pr01: 1167.74 < 1260.41, pr02: 2422.94 < 2527.15, and pr07: 1594.15 < 1691.14.

In order to determine the right solution values found by the two MILP formulations proposed in Bruni et al. (2022a) for the instances of this data set, we implemented both MILP formulations and solved all the instances on our computer using the MILP solver Gurobi 9.1.2 (Gurobi Optimization, 2021) with a time limit of 4392 s, which is the scaled value of the time limit of 7200 s imposed in Bruni et al. (2022a). First, we solved Model 2; only in case the time limit is reached we solved Model 1, and the best solution value found is reported in column Best. All the values presented in Table 12 for the MILP formulations correspond to those found by our implementation and should be used in future research. It is to note that all the instances but p18 and pr10 were solved to proven optimality within the time limit. For the instance p18 the best lower bound found was equal to 11 364.06, implying an optimality gap equal to 0.61%, while for the instance pr10, the best lower bound found was equal to 9082.66, implying an optimality gap equal to 0.28%. These lower bounds are tighter than LB (11 364.06 > 9934.63, and 9082.66 > 7645.90, for the instances p18 and pr10, respectively), and should be used in future research as best lower bounds for these instances. In the Appendix the optimal solutions for two instances are presented. In the results reported in Bruni et al. (2022a), the solution values obtained by algorithm GA for the instances pr01 and pr07 are smaller than the corresponding optimal solution values, hence these values are not considered in Table 12. It is to note that Table 12 does not include the values of gap_F since the formulations always provide the best-known solution values, which implies that gap_F is equal to gap_B for all the instances of this data set.

According to the results shown in Table 12, M-ILS can find the optimal solution values for 8, 5, and 4 instances by performing 10 run, 5 runs, and 1 run, respectively. For all the other instances, it can provide near-optimal solutions independently of the number of runs. The global average value of gap_B is equal to 0.11% by executing M-ILS with 10 runs, 0.14% when it is executed with 5 runs and 0.43% considering a single run. Although, by executing 10 runs for each instance, M-ILS is more time-consuming than the formulations, by considering 5 runs, the global computing time is slightly smaller than that required by the formulations, and by considering a single run, M-ILS is six times faster than the formulations.

Compared to GA, M-ILS is superior in the solution quality since the global value of gap_{GA} is around -8%, independently of the number of runs. For all the instances but two, the best solution value provided by M-ILS is better than that provided by GA (independently of the number of runs). For the two remaining instances, when 10 runs are considered, M-ILS and GA find the same best (optimal) solution value for both instances, while when 5 runs are considered, each algorithm finds a solution better than that of the competitor for one instance. Furthermore, the Avg. solution value provided by M-ILS, by considering 10 and 5 runs, is smaller than (for 20 instances) or equal to (for one instance) the best solution value obtained by GA for all the instances but one. Regarding the computing times of the two metaheuristics, M-ILS is more time-consuming than GA when it is executed with 10 and 5 runs, while it is almost the same when it is executed with a single run.

3.4.3. The lr data set

The computational results corresponding to the 21 instances of the lr data set are reported in Table 13. Since the instances lr10, lr11, and lr12 with 25 vehicles were not considered in Bruni et al. (2022a), we solved them optimally under the same conditions mentioned in Section 3.4.2. The values reported in Bruni et al. (2022a) for both GA and the formulations are presented for all the other instances. This is the easiest of the considered data sets since all its instances have been solved to proven optimality by the formulations within very short computing times (at most 37.8 s).

The results presented in Table 13 show that M-ILS can find the optimal solution value for 15, 14, and 9 instances by performing 10 runs, 5 runs, and 1 run, respectively. For all the other instances, M-ILS can provide near-optimal solution values independently of the number of runs. The average value of gap_B is equal to 0.06% by performing 10 runs, 0.09% for 5 runs and 0.29% considering a single run. M-ILS is more time-consuming than both the formulations and GA, independently of the number of runs. Nevertheless, by executing only one run, M-ILS provides good quality solutions in reasonable computing times (on average around 40 s). For what concerns the values of the solutions provided by M-ILS compared to those obtained by GA, it is possible to conclude that M-ILS outperforms GA independently of the number of runs. By executing M-ILS with 10 runs, the best solution value found by M-ILS is better than (for 12 instances) or equal to (for 5 instances) the best solution value provided by GA for all the instances but one. Similarly, by executing M-ILS with 5 runs, the best solution value found by M-ILS is better than (for 11 instances) or equal to (for 5 instances) the best solution value provided by GA for all the instances but two. By executing a single run, the best solution value provided by M-ILS is better than (for 10 instances) or equal to (for 3 instances) the best solution value provided by GA for all the instances but 5. The good performance of M-ILS is more evident for the large size instances of this data set ($N_C = 100$) for which the average value of gap_{GA} is equal to –2.98% (considering a single run for M-ILS).

3.4.4. The lr data set with reduced fleet

The computational results corresponding to the 18 instances of the lr data set with reduced fleet are reported in Table 14. Since for the instances lr5 and lr15 the best solution values of the feasible solutions found by M-ILS are smaller than the optimal solution values reported in Bruni et al. (2022a), we solved these two instances optimally under the same conditions mentioned in Section 3.4.2. The optimal solution values presented in Table 14 are the correct ones, and should be used in future research. For what concerns the results reported in Bruni et al. (2022a) regarding GA, the best solution value presented for the instance lr4 is smaller than the optimal solution value, so this value is not considered in Table 14. The original values reported in Bruni et al. (2022a) are presented for all the other instances. It is to note that all the instances of this data set have been solved to proven optimality with the formulations.

The results presented in Table 14 show that M-ILS can find the optimal solution value for 12, 10, and 9 instances by performing 10 runs, 5 runs, and 1 run, respectively. The proposed algorithm can find near-optimal solution values for all the remaining instances, obtaining an average value of gap_B equal to 0.20%, 0.24%, and 0.64% when the number of runs executed for each instance is 10, 5, and 1, respectively. Regarding the global computing times, M-ILS is more than four times faster than the formulations and two times faster than GA when 10 runs are performed. M-ILS outperforms GA for what concerns both the computing time and the solution quality. When the number of runs executed is equal to 10 or 5 for each instance, M-ILS finds better solution values than those obtained by GA for 14 instances and the same solution value for the remaining 3 instances. When a single run is considered for M-ILS, it finds solution values better than (for 13 instances) or equal to (for 3 instances) those obtained by GA for all the instances but one. In addition, the average solution value obtained by M-ILS by performing 10 runs is better than (for 12 instances) or equal to (for 3 instances) the best solution value obtained by GA for all the instances but two. Similarly, the average solution value obtained by M-ILS by performing 5 runs is better than (for 13 instances) or equal to (for 3 instances) the best solution value obtained by GA for all the instances but one. M-ILS obtained an average value of gap_{GA} equal to –2.42%, –2.39%, and –1.98% when the number of runs executed for each instance is 10, 5, and 1, respectively. It is to note also that for 6 (7) instances, the average solution value found by M-ILS corresponds to the optimal solution value when 10 (5) runs are performed.

3.4.5. Overall results on the MDk-TRP

After analyzing the results, it is possible to conclude that the proposed algorithm M-ILS overcomes GA in terms of solution quality for all the considered data sets by executing a number of runs equal to 10, 5, or 1 for each instance. The global average value of gap_{GA} is equal to –5.20%, –5.17%, and –4.87% when M-ILS is executed with 10, 5, and 1 runs, respectively. By considering a single run, M-ILS is globally faster than GA for 2 of the 4 data sets, while the average global computing time is similar and competitive (less than 60 s) for the remaining two data sets. By considering the global average computing time (computed over all the instances), M-ILS (executed with 1 run) is three times faster than GA (112.21 s vs. 376.05 s, respectively). For the two data sets with reduced fleets (which contain the most challenging instances), M-ILS is faster than GA when 5 runs are executed. Compared to the formulations, M-ILS finds optimal or near-optimal solution values in globally shorter computing times for most of the considered instances. The global average value of gap_B (computed over all the instances) is equal to 0.11% for the formulations, and 0.17% for M-ILS (when 10 runs are executed), while the average global computing time is 1677.82 s for the formulations and 1077.61 s for M-ILS (when 10 runs are executed).

Regarding the proposed lower bounds, we found that the average value of gap_{LB} for the 10 instances not solved to proven optimality is equal to 24.77%. Also for this problem, despite this value is large, it does not mean that the BKS values for these instances correspond to bad quality solutions. Indeed, the average value of gap_{LB} obtained by considering only the instances solved to proven optimality is equal to 20.52%, which means that the proposed lower bounds are not tight. It is possible to note that the proposed lower bounds provide reasonable good approximations of the optimal solution value for instances with a relative large number of vehicles. The average value of gap_{LB} obtained by considering only the instances in the data sets p-pr with $N_v = 35$ and lr is equal to 6.38%, while when only the instances in the data sets p-pr and lr with reduced fleet are considered, the value obtained is equal to 36.68%.

In order to compare the efficiency of the algorithm M-ILS with that of the algorithm GA, new experiments were carried out to compare the quality of the solutions obtained by both algorithms within the same global computing time, and to determine the computing time required by M-ILS to reach for each instance a given target value. Let us consider, for each instance, a target value (TV) given by the best solution value obtained by the algorithm GA and a time limit (TL) given by the global computing time required by GA to find the target value.

The summary of the average results of this experiment is presented in Table 15 for each data set and for the overall set of instances. The reported columns correspond to: the averages of the target values (TV) and of the time limits (TL), the average of the best solution value found by M-ILS within the time limit (Best$_{TL}$), the average of the percentage gap between Best$_{TL}$ and TV (gap_{TV}), the number of instances for which the target value is found by M-ILS within the time limit (#TV), and the average of the computing times required by M-ILS to find the target values (time$_{TV}$). It is to note that, for the instances for which M-ILS cannot find the target value, time$_{TV}$ is equal to the time limit. For the instances for which TL is equal to 0.0 we allow M-ILS to perform only the Constructive phase, which generally requires small computing times, and for small instances requires less than 1 s.

The results show that the M-ILS algorithm can find (for 7 instances) or improve (for 58 instances) the target value for 65 out of 81 instances within the time limit, obtaining a global average value of gap_{TV} equal to −4.50%. Considering all the instances, M-ILS requires considerably less computing time than that required by GA for finding solutions with similar quality, and when M-ILS is executed for the same global time reported for GA it is able to largely outperform GA in terms of solution quality. The only data set for which M-ILS does not totally dominate GA is the lr data set. For these instances M-ILS is generally able to perform only the Constructive procedure. Although the results of both heuristics are competitive in terms of solution quality and computing time, M-ILS performs slightly better than GA, being able to find the target value for half of the instances within the time limit. On the other hand, the average computing times required by GA for finding the target values are 17, 12, and 44 times larger than those required by M-ILS for the p-pr with reduced fleet, p-pr with $N_v = 35$, and lr with reduced fleet data sets, respectively.

The computational experiments presented in this Section show that M-ILS clearly outperforms GA both in terms of solution quality and computing time.

3.5. The latency location routing problem

The algorithms proposed in the literature for the solution of the LLRP are the following: the memetic algorithm (MA) and the recursive granular algorithm (RGA) proposed in Moshref-Javadi and Lee (2016), the exact methods and the GRASP-based iterated local search algorithm (GBILS) presented in Nucamendi-Guillén et al. (2022), and the three simulated annealing-variable neighborhood descent based metaheuristics SA-VND0, SA-VND1, and SA-VND2 presented in Osorio-Mora et al. (2023). For the LLRP, the M-ILS algorithm is executed for each instance with a number of runs equal to 30 and 5.

The results reported in Moshref-Javadi and Lee (2016) indicate that the algorithm MA dominates the RGA approach. Besides, the exact methods proposed in Nucamendi-Guillén et al. (2022) are able to solve to optimality only instances with up to 50 customers. In particular, they solve the instances with 20 customers, from 21 to 36 customers, and with 50 customers, with average scaled computing times equal to 28, 465, and 5312 s, respectively. For all the instances with more than 50 customers, the exact methods are able to find, within a time limit of 21 960 s, feasible solutions with an average value of gap_B (considering the values of BKS reported in Tables 17 and 18) equal to 5.05%. For large-size instances, the metaheuristic algorithms GBILS, SA-VND0, SA-VND1, and SA-VND2 perform globally better than the other methods for what concerns both the solution quality and the computing times. According to the results presented in Osorio-Mora et al. (2023), the algorithms SA-VND0, SA-VND1, and SA-VND2 are also more effective than RGA for all the considered instances. Therefore, we have excluded the algorithm RGA (Moshref-Javadi and Lee, 2016) and the exact methods (Nucamendi-Guillén et al., 2022) for comparison purposes. On the other hand, despite the algorithms SA-VND0, SA-VND1, and SA-VND2 have been proved to outperform the GBILS approach, the latter is included in the comparison since it presents relatively good solution quality and short computing times. Note that no average solution values have been reported in Nucamendi-Guillén et al. (2022) for GBILS.

In order to present a fair comparison, the global computing times reported in Tables 16–18 for each instance correspond to:

(i) for MA to the times reported in Moshref-Javadi and Lee (2016), which correspond to the execution of 30 runs. The experiments in Moshref-Javadi and Lee (2016) were performed on a 3.1 GHz computer with 4 GB RAM. The above is the only information available about this computer, and it does not allow us to determine a scaling factor. Nevertheless, considering the ratio between the corresponding values of GHz, it is possible to estimate that our computer is about 1.2 times faster than that used in Moshref-Javadi and Lee (2016).

(ii) for GBILS to the scaled computing time (considering a scaling factor equal to 0.61) of the times reported in Nucamendi-Guillén et al. (2022), which correspond to the execution of 5 runs;

(iii) for SA-VND0, SA-VND1, and SA-VND2 to the computing time reported in Osorio-Mora et al. (2023), corresponding to the global computing time associated with 30 runs for each algorithm. It is to note that the computing times of the mentioned algorithms do not need to be scaled since these algorithms were executed on the same computer on which the M-ILS has been executed;

The computing times reported for M-ILS correspond to the average computing times multiplied by the respective number of runs.

Furthermore, for each number of runs of M-ILS, the following values are reported:

Table 15
Average results obtained by each metaheuristic considering time limits and target values for each MDk-TRP data set.

Data set	# Instances	GA		M-ILS			
		TV	TL	Best$_{TL}$	gap_{TL}	#TV	time$_{TL}$
p-pr with reduced fleet	24	7852.07	740.4	**7293.30**	**−7.08**	**22**	**43.7**
p-pr with $N_v = 35$	22	6734.22	207.6	**5910.76**	**−7.75**	**19**	**18.0**
lr	18	4047.94	7.8	**4001.30**	**−0.15**	9	**6.1**
lr with reduced fleet	17	6763.40	469.6	**6573.52**	**−2.02**	**15**	**10.6**
All the instances	81	6474.61	376.1	**6035.17**	**−4.66**	**65**	**21.4**

- gap_{MA}: Percentage gap between the Best solution value found by M-ILS ($Best$) and the Best solution value found by MA (Best$_{MA}$), computed as $gap_{MA} = 100\frac{(Best-Best_{MA})}{Best_{MA}}$.
- gap_{SA}: Percentage gap between the Best solution value found by M-ILS ($Best$) and the Best solution value found by the best among SA-VND0, SA-VND1, and SA-VND2 (Best$_{SA}$), computed as $gap_{SA} = 100\frac{(Best-Best_{SA})}{Best_{SA}}$.
- gap_{GBILS}: Percentage gap between the Best solution value found by M-ILS and the Best solution value found by GBILS (Best$_{GBILS}$), computed as $gap_{GBILS} = 100\frac{(Best-Best_{GBILS})}{Best_{GBILS}}$.

3.5.1. The Tuzun–Burke data set

This data set contains the most challenging benchmark instances for the LLRP. Table 16 gives the corresponding results. Since no results for GBILS are reported on this data set in Nucamendi-Guillén et al. (2022), M-ILS is compared only with the MA algorithm presented in Moshref-Javadi and Lee (2016), and the three metaheuristics presented in Osorio-Mora et al. (2023). As the table indicates, M-ILS outperforms MA, SA-VND0, SA-VND1, and SA-VND2 regarding both the solution quality and the computing time.

The proposed M-ILS improves the best-known solution value for 32 out of the 36 instances of this data set. The average values of gap_B are 5.56%, 0.77%, 0.84%, and 0.86% for MA, SA-VND0, SA-VND1, and SA-VND2, respectively, while this value is equal to 0.04% for M-ILS. Regarding the computing times, M-ILS is more than two times faster than SA-VND1 and SA-VND2 and 1.5 times faster than SA-VND0 (considering 30 runs for each algorithm). On the other hand, when 30 runs are considered for M-ILS, its computing times are larger than those of MA; nevertheless, when the number of runs is reduced to 5, M-ILS is three times faster than MA, being able to provide a better solution value than MA for all the instances. Furthermore, when 5 runs are considered, M-ILS provides a better solution value than the best found by SA-VND0, SA-VND1, and SA-VND2 for 22 instances, in global computing times more than 9 times shorter than those required by SA-VND0, which is the fastest among the three mentioned algorithms.

The average solution value obtained by M-ILS is better than that obtained by MA, SA-VND0, SA-VND1, and SA-VND2 for 36, 33, 34, and 32 instances, respectively. Furthermore, for seven instances, the average solution value obtained by M-ILS is better than the best solution value obtained by the best among the four competitors. Note that the average solution value provided by M-ILS is better than the best solution value reported for MA for all the instances but one (considering 30 runs). In the same direction, the average gap_{SA} value equals −0.51%, and −0.22%, and the average gap_{MA} value equals −5.17%, −4.90%, when 30 and 5 runs are considered for M-ILS, respectively.

3.5.2. The Prodhon data set

The results of this data set are presented in Table 17. Over the 30 instances of this data set, M-ILS, executed with 30 runs, can find the proved optimal solution value for 11 instances, provides new best-known solution values for 16 instances, and finds the current best-known solution value for one instance. The average values of gap_B obtained by M-ILS are equal to 0.05% and 0.19% considering 30 and 5 runs, respectively. These values of gap_B are better than those associated with all the competitors. It is to note that instance 50-5-3 was not solved to proven optimality in Nucamendi-Guillén et al. (2022). Nevertheless, we implemented the MILP formulation "Model 2" presented in Nucamendi-Guillén et al. (2022) and solved this instance without considering a time limit, proving that the best solution value reported in Table 17 corresponds to the optimal one.

Comparing M-ILS with the current state-of-the-art metaheuristics proposed in Osorio-Mora et al. (2023), it is possible to conclude that M-ILS outperforms the three algorithms SA-VND0, SA-VND1, and SA-VND2, obtaining an average value of gap_{SA} equal to −0.24%, and −0.09% when 30, and 5 runs are considered, respectively. Regarding the computing time, M-ILS is 1.7, 2.7, and 2.5 times faster than SA-VND0, SA-VND1, and SA-VND2, respectively (considering 30 runs for each algorithm). When 5 runs are considered, M-ILS can find solution values better than (for 13 instances) or equal to (for 9 instances) the best found by SA-VND0, SA-VND1, and SA-VND2 for 22 instances, in global computing times more than ten times smaller than those required by SA-VND0, which is the fastest among the three mentioned algorithms. Although GBILS is faster than M-ILS (independently of the number of runs), the values of the solutions obtained by M-ILS are considerably better than those found by GBILS. The average value of gap_{GBILS} equals −2.40% and −2.26% considering 30 and 5 runs, respectively. Furthermore, the average solution value obtained by M-ILS (considering 30 runs) is better than or equal to the best solution value obtained by GBILS for all the instances but 3. Also, considering 5 runs, M-ILS provides a solution value better than (for 25 instances) or equal to (for 4 instances) the best solution value reported for GBILS for all the instances but one. Finally, M-ILS is able to provide a solution value better than (for 28 instances) or equal to (for one instance) that reported for MA for all the instances but one, independently of the number of runs. When 30 runs are considered, M-ILS is more time-consuming than MA, obtaining an average gap_{MA} equal to −3.77%; nevertheless, considering 5 runs, M-ILS is 3.3 times faster than MA, providing an average value of gap_{MA} equal to −3.63%.

3.5.3. The Barreto data set

This data set considers the less complex instances for the LLRP. Indeed, 6 out of 10 instances of this data set (those for which $N_c \leq 50$) have been solved to proven optimality with the MILP models presented in Nucamendi-Guillén et al. (2022). The results of this data set are presented in Table 18. For the instances Min-134-8 and Or-117-14, no results are reported for GBILS in Nucamendi-Guillén et al. (2022), while for the instances Christ-50-5 and Christ-75-10 the results provided in Nucamendi-Guillén et al. (2022) for GBILS were neglected (as done in Osorio-Mora et al. (2023)), since they correspond to different instances. The last line of Table 18 gives the average values (Global avg GBILS) computed by considering only the 6 instances whose values are correctly reported in Nucamendi-Guillén et al. (2022). Since GBILS obtains the optimal solution value for all the instances reported but for Christ-100-10, the column gap_{GBILS} is not included in the table. According to the results, M-ILS is able to find the proven optimal solution value for 6 instances and, for two instances, new best-known solution values. There are no significant differences concerning the solution quality and the computing times between M-ILS and the three heuristic

Table 16
Detailed results for the LLRP Tuzun-Burke data set.

Instance	N_c	N_d	N_v	BKS	MA				SA-VND0				SA-VND1				SA-VND2				M-ILS 30 runs						M-ILS 5 runs					
					Best	Avg	Time	gap_B	Best	Avg	Time	gap_B	Best	Avg	Time	gap_B	Best	Avg	Time	gap_B	Best	Avg	Time	gap_B	gap_{MA}	gap_{SA}	Best	Avg	Time	gap_B	gap_{MA}	gap_{SA}
111112	100	10	11	3834.91	4017.90	4270.10	1200.0	4.77	3862.86	3971.50	1 912.8	0.73	3892.97	3972.93	2 390.6	1.51	3882.76	3964.12	2 801.2	1.25	**3834.91**	3890.13	2149.2	0.00	−4.55	−0.72	3849.10	3895.17	342.0	0.37	−4.20	−0.36
111122	100	20	11	3612.36	3719.10	4087.50	1200.0	2.95	**3612.36**	3694.70	1 934.6	0.00	3633.60	3712.64	2 419.2	0.59	3623.69	3687.85	2 860.6	0.31	3659.46	3695.69	2130.6	1.30	−1.60	1.30	3683.58	3692.65	342.1	1.97	−0.96	1.97
111212	100	10	10	3919.74	4264.40	4563.50	1245.0	8.79	3960.24	4038.24	1 805.4	1.03	3988.11	4067.91	2 335.5	1.74	3938.88	4066.13	2 692.0	0.49	**3919.74**	4000.94	2455.1	0.00	−8.08	−0.49	**3919.74**	3993.54	394.9	0.00	−8.08	−0.49
111222	100	20	11	4065.04	4278.30	4557.30	1242.0	5.25	4086.74	4140.33	1 909.6	0.53	4077.87	4147.90	2 376.2	0.32	4068.34	4138.78	2 785.2	0.08	**4065.04**	4139.64	2312.7	0.00	−4.98	−0.08	**4065.04**	4125.54	379.0	0.00	−4.98	−0.08
112112	100	10	11	2726.41	2795.60	3049.70	1242.0	2.54	2739.16	2755.53	2 281.4	0.47	2740.21	2759.43	2 908.3	0.51	2741.82	2756.07	3 271.1	0.57	**2726.41**	2749.56	1649.9	0.00	−2.47	−0.47	2747.04	2751.00	265.7	0.76	−1.74	0.29
112122	100	20	11	2057.30	2097.90	2702.60	1329.0	1.97	2060.29	2072.57	2 211.6	0.15	2057.45	2078.03	2 797.4	0.01	2060.80	2071.85	3 205.3	0.17	**2057.30**	2063.89	1193.4	0.00	−1.94	−0.01	2059.08	2059.28	169.6	0.09	−1.85	0.08
112212	100	10	12	1394.65	1442.20	1604.90	1455.0	3.41	1402.97	1416.20	2 349.9	0.60	1403.57	1416.72	3 050.0	0.64	1397.39	1414.98	3 335.3	0.20	**1394.65**	1411.85	1490.7	0.00	−3.30	−0.20	**1394.65**	1411.25	240.1	0.00	−3.30	−0.20
112222	100	20	11	1621.40	1659.00	2084.10	1314.0	2.32	1623.69	1633.00	2 522.2	0.14	**1621.40**	1633.94	3 137.8	0.00	1626.86	1633.89	3 562.2	0.34	1623.39	1630.36	1652.2	0.12	−2.15	0.12	1623.39	1636.93	337.6	0.12	−2.15	0.12
113112	100	10	11	2828.24	2897.70	3162.60	1170.0	2.46	2837.51	2852.63	2 043.2	0.33	2835.76	2853.57	2 735.6	0.27	2839.50	2857.50	3 014.0	0.40	**2828.24**	2841.93	2091.6	0.00	−2.40	−0.27	2837.82	2843.77	377.9	0.34	−2.07	0.07
113122	100	20	11	2772.98	2912.00	3330.00	1248.0	5.01	2776.38	2782.52	2 063.4	0.12	2774.36	2784.42	2 762.2	0.05	2776.38	2782.07	2 998.4	0.12	**2772.98**	2797.08	2753.4	0.00	−4.77	−0.05	**2772.98**	2799.46	445.6	0.00	−4.77	−0.05
113212	100	10	12	1815.62	1832.10	1901.00	1458.0	0.91	1817.00	1823.15	2 326.4	0.08	**1815.62**	1822.81	2 922.2	0.00	**1815.62**	1822.76	3 326.7	0.00	1817.00	1835.80	2023.4	0.08	−0.82	0.08	1817.33	1832.76	382.7	0.09	−0.81	0.09
113222	100	20	11	1876.14	2037.50	2360.60	1239.0	8.60	**1876.14**	1888.46	2 117.2	0.00	1879.63	1890.96	2 816.5	0.19	1876.93	1888.90	3 063.5	0.04	1876.58	1885.31	1940.0	0.02	−7.90	0.02	1882.30	1888.46	323.2	0.33	−7.62	0.33
131112	150	10	16	5410.96	5863.80	6160.70	1797.0	8.37	5473.17	5582.94	4 060.2	1.15	5464.21	5570.12	6 108.7	0.98	5448.86	5576.01	6 409.4	0.70	**5410.96**	5478.43	3854.6	0.00	−7.72	−0.70	5434.61	5490.75	653.7	0.44	−7.32	−0.26
131122	150	20	16	4926.87	5310.30	5649.30	1746.0	7.78	4993.36	5142.06	4 462.0	1.35	5009.26	5143.19	6 549.2	1.67	4974.28	5105.72	6 876.6	0.96	**4926.87**	5053.13	3970.3	0.00	−7.22	−0.95	5003.55	5060.78	663.0	1.56	−5.78	0.59
131212	150	10	16	5528.85	5967.60	6234.20	1746.0	7.94	5679.70	5787.34	4 588.6	2.73	5606.31	5785.18	6 765.0	1.40	5653.20	5771.63	7 051.5	2.25	**5528.85**	5639.90	4007.7	0.00	−7.35	−1.38	**5528.85**	5615.26	693.1	0.00	−7.35	−1.38
131222	150	20	16	5060.71	5261.00	5610.40	1767.0	3.96	5141.89	5284.45	4 580.2	1.60	5126.95	5277.65	6 627.3	1.31	5134.39	5296.62	7 093.9	1.46	**5060.71**	5107.53	4093.6	0.00	−3.81	−1.29	**5060.71**	5090.48	685.6	0.00	−3.81	−1.29
132112	150	10	16	3850.90	4026.70	4246.20	1746.0	4.57	3868.88	3895.91	5 963.0	0.47	3883.40	3899.41	8 542.7	0.84	3874.44	3894.29	8 771.5	0.61	**3850.90**	3881.70	4034.9	0.00	−4.37	−0.46	3856.34	3874.13	573.8	0.14	−4.23	−0.32
132122	150	20	16	3734.83	3874.00	4185.50	1785.0	3.73	3740.10	3795.93	5 155.0	0.14	3752.76	3787.72	7 796.2	0.48	3755.51	3797.33	7 768.9	0.55	**3734.83**	3762.41	3762.9	0.00	−3.59	−0.14	3744.44	3760.24	590.9	0.26	−3.34	0.12
132212	150	10	17	2835.66	2906.00	3129.90	1779.0	2.48	2842.10	2857.43	5 839.7	0.23	2837.84	2860.70	8 792.6	0.08	2843.18	2857.02	8 714.5	0.27	**2835.66**	2850.38	3345.7	0.00	−2.42	−0.08	2844.11	2853.48	580.1	0.30	−2.13	0.22
132222	150	20	17	1651.91	1784.80	2152.30	1773.0	8.04	1660.89	1691.97	6 340.4	0.54	1672.86	1697.45	9 145.6	1.27	1677.95	1695.15	9 288.8	1.58	**1651.91**	1676.40	2824.2	0.00	−7.45	−0.54	1664.08	1678.64	485.9	0.74	−6.76	0.19
133112	150	10	16	4578.87	5034.00	5465.90	1737.0	9.94	4588.37	4619.91	4 670.1	0.21	4598.23	4630.69	7 313.4	0.42	4596.35	4625.72	7 130.4	0.38	**4578.87**	4612.46	3339.0	0.00	−9.04	−0.21	4590.97	4614.54	532.7	0.26	−8.80	0.06
133122	150	20	16	3211.98	3474.00	3849.10	1767.0	8.16	3223.44	3259.45	5 317.0	0.36	3225.56	3271.04	8 052.6	0.42	3223.40	3248.42	8 012.0	0.36	**3211.98**	3236.27	3813.4	0.00	−7.54	−0.35	**3211.98**	3222.46	677.2	0.00	−7.54	−0.35
133212	150	10	17	2903.36	3008.00	3284.50	1770.0	3.60	2911.58	2938.05	5 887.3	0.28	2911.35	2938.01	8 572.0	0.28	2906.96	2935.00	8 754.9	0.12	**2903.36**	2918.06	3116.3	0.00	−3.48	−0.12	2908.51	2917.75	516.9	0.18	−3.31	0.05
133222	150	20	17	2485.07	2617.40	3016.90	1755.0	5.33	2502.97	2550.01	5 692.6	0.72	2502.68	2558.34	8 274.9	0.71	2501.03	2531.56	8 664.8	0.64	**2485.07**	2496.12	3020.2	0.00	−5.06	−0.64	**2485.07**	2497.99	555.3	0.00	−5.06	−0.64
121112	200	10	21	6572.43	7008.70	7371.10	2502.0	6.64	6608.45	6821.20	8 392.7	0.55	6621.55	6881.38	13 919.9	0.75	6683.24	6848.60	14 006.4	1.69	**6572.43**	6632.64	6096.4	0.00	−6.22	−0.55	**6572.43**	6630.65	1038.2	0.00	−6.22	−0.55
121122	200	20	22	5612.03	6039.60	6501.60	2490.0	7.62	5730.53	5954.23	9 991.8	2.11	5788.72	5966.38	15 672.5	3.15	5784.71	5952.63	16 313.2	3.08	**5612.03**	5668.75	5948.1	0.00	−7.08	−2.07	5631.62	5648.04	997.3	0.35	−6.76	−1.73
121212	200	10	21	6409.74	6744.30	7318.60	2559.0	5.22	6503.36	6613.84	8 391.2	1.46	6429.62	6608.92	13 778.5	0.31	6502.73	6610.46	13 910.1	1.45	**6409.74**	6449.92	5906.0	0.00	−4.96	−0.31	6421.16	6446.48	911.5	0.18	−4.79	−0.13
121222	200	20	21	6383.30	6828.50	7567.20	2553.0	6.97	6551.73	6759.22	8 391.4	2.64	6562.11	6796.71	14 023.1	2.80	6648.20	6776.44	13 851.2	4.15	**6383.30**	6526.13	6610.6	0.00	−6.52	−2.57	6480.12	6537.58	1158.5	1.52	−5.10	−1.09
122112	200	10	21	6111.52	6643.80	7106.90	2571.0	8.71	6154.64	6255.10	9 463.2	0.71	6184.70	6280.31	18 340.7	1.20	6168.32	6268.36	14 679.6	0.93	**6111.52**	6208.87	9151.9	0.00	−8.01	−0.70	6167.63	6206.17	1486.9	0.92	−7.17	0.21
122122	200	20	21	3725.07	4012.90	4915.70	2547.0	7.73	3757.37	3782.47	10 555.0	0.87	3757.27	3792.68	17 314.9	0.86	3744.74	3785.46	16 701.0	0.53	**3725.07**	3756.16	4976.8	0.00	−7.17	−0.53	3728.34	3768.28	748.8	0.09	−7.09	−0.44
122212	200	10	21	4025.13	4227.50	4448.10	2535.0	5.03	4046.81	4075.78	9 845.3	0.54	4046.42	4078.60	17 116.3	0.53	4043.53	4077.42	15 490.2	0.46	**4025.13**	4042.78	3871.8	0.00	−4.79	−0.46	4040.88	4045.36	584.4	0.39	−4.41	−0.07
122222	200	20	22	2049.68	2127.90	2345.50	2511.0	3.82	2054.31	2083.56	11 266.8	0.23	2052.22	2084.10	18 533.2	0.12	2052.16	2079.68	17 652.2	0.12	**2049.68**	2056.08	3826.1	0.00	−3.68	−0.12	**2049.68**	2052.86	720.0	0.00	−3.68	−0.12
123112	200	10	22	4868.90	5099.00	5527.40	2559.0	4.73	4916.97	5024.07	10 869.1	0.99	4967.11	5047.87	17 275.1	2.02	4940.81	5029.64	17 259.7	1.48	**4868.90**	4921.23	6253.2	0.00	−4.51	−0.98	4880.27	4917.27	1102.4	0.23	−4.29	−0.75
123122	200	20	22	4675.19	5188.70	5862.90	2544.0	10.98	4725.91	4771.90	10 590.0	1.08	4707.61	4785.89	16 689.1	0.69	4719.90	4777.07	17 021.6	0.96	**4675.19**	4703.17	6099.4	0.00	−9.90	−0.69	4682.46	4690.47	988.0	0.16	−9.76	−0.53
123212	200	10	22	5135.21	5363.00	5678.50	2544.0	4.44	5170.77	5218.43	10 206.0	0.69	5178.03	5225.04	17 737.1	0.83	5195.48	5247.18	16 205.9	1.17	**5135.21**	5174.74	4033.7	0.00	−4.25	−0.69	5143.39	5166.96	619.7	0.16	−4.09	−0.53
123222	200	20	22	2522.89	2657.50	3917.60	2577.0	5.34	2567.20	2629.46	10 676.9	1.76	2555.18	2633.86	18 065.4	1.28	2553.21	2606.54	17 096.2	1.20	**2522.89**	2551.90	3961.3	0.00	−5.07	−1.19	2526.68	2544.00	640.0	0.15	−4.92	−1.04
Global avg.				3799.88	4028.41	4422.78	1861.2	5.56	3835.27	3901.77	5 740.9	0.77	3837.85	3909.51	8 990.5	0.84	3840.99	3902.19	8 934.4	0.86	**3801.30**	3842.98	3715.6	**0.04**	**−5.17**	**−0.51**	3814.16	**3840.57**	**616.8**	0.34	−4.90	−0.22

Table 17
Detailed results for the LLRP Prodhon data set.

Instance	N_v	BKS	MA				GBILS			SA-VND0				SA-VND1				SA-VND2				M-ILS 30 runs							M-ILS 5 runs						
			Best	Avg	Time	gap_B	Best	Time	gap_B	Best	Avg	Time	gap_B	Best	Avg	Time	gap_B	Best	Avg	Time	gap_B	Best	Avg	Time	gap_B	gap_{MA}	gap_{GBILS}	gap_{SA}	Best	Avg	Time	gap_B	gap_{MA}	gap_{GBILS}	gap_{SA}
20-5-1	5	330.00	337.30	378.00	387.0	2.21	**330.00**	1.0	0.00	**330.00**	**330.00**	125.2	0.00	**330.00**	**330.00**	117.8	0.00	**330.00**	**330.00**	166.4	0.00	**330.00**	**330.00**	167.8	0.00	−2.16	0.00	0.00	**330.00**	**330.00**	30.2	0.00	−2.16	0.00	0.00
20-5-1b	3	608.06	**608.06**	636.80	375.0	0.00	608.06	0.5	0.00	**608.06**	**608.06**	145.7	0.00	**608.06**	**608.06**	122.3	0.00	**608.06**	**608.06**	244.9	0.00	615.66	615.66	113.6	1.25	1.25	1.25	1.25	615.66	615.66	19.7	1.25	1.25	1.25	1.25
20-5-2	5	301.97	304.80	354.40	381.0	0.94	**301.97**	0.7	0.00	**301.97**	**301.97**	106.3	0.00	**301.97**	**301.97**	98.5	0.00	**301.97**	**301.97**	150.5	0.00	**301.97**	**301.97**	161.8	0.00	−0.93	0.00	0.00	**301.97**	**301.97**	27.2	0.00	−0.93	0.00	0.00
20-5-2b	3	486.55	**486.55**	511.20	381.0	0.00	**486.55**	0.8	0.00	**486.55**	**486.55**	158.7	0.00	**486.55**	**486.55**	132.8	0.00	**486.55**	**486.55**	266.6	0.00	**486.55**	**486.55**	114.3	0.00	0.00	0.00	0.00	**486.55**	**486.55**	18.7	0.00	0.00	0.00	0.00
50-5-1	12	843.94	859.90	917.30	546.0	1.89	846.88	153.5	0.35	846.17	849.77	709.2	0.27	846.52	850.10	815.3	0.31	**843.94**	850.40	859.5	0.00	**843.94**	845.80	589.0	0.00	−1.86	−0.35	0.00	**843.94**	845.19	92.3	0.00	−1.86	−0.35	0.00
50-5-1b	6	1293.46	1330.20	1379.80	522.0	2.84	1293.93	65.5	0.04	**1293.46**	1293.71	619.7	0.00	**1293.46**	**1293.54**	602.2	0.00	**1293.46**	1293.55	933.8	0.00	**1293.46**	1293.95	682.8	0.00	−2.76	−0.04	0.00	**1293.46**	**1293.46**	107.5	0.00	−2.76	−0.04	0.00
50-5-2	12	684.13	723.40	786.20	552.0	5.74	691.67	117.4	1.10	**684.13**	694.42	624.0	0.00	**684.13**	692.43	756.3	0.00	**684.13**	694.69	752.8	0.00	**684.13**	690.41	930.0	0.00	−5.43	−1.09	0.00	**684.13**	689.20	164.4	0.00	−5.43	−1.09	0.00
50-5-2b	6	953.25	965.70	1009.40	573.0	1.31	954.88	68.4	0.17	**953.25**	953.50	534.8	0.00	**953.25**	953.35	534.2	0.00	**953.25**	953.33	799.3	0.00	**953.25**	953.67	563.9	0.00	−1.29	−0.17	0.00	**953.25**	953.27	94.9	0.00	−1.29	−0.17	0.00
50-5-2BIS	12	945.45	955.20	981.50	537.0	1.03	952.55	120.7	0.75	949.13	950.77	883.4	0.39	949.57	951.13	1 081.1	0.44	950.12	950.93	1 024.7	0.49	**945.45**	945.77	1493.0	0.00	−1.02	−0.75	−0.39	**945.45**	946.31	230.3	0.00	−1.02	−0.75	−0.39
50-5-2bBIS	6	803.90	811.80	884.90	534.0	0.98	**803.90**	96.6	0.00	**803.90**	**803.90**	626.9	0.00	**803.90**	**803.90**	649.9	0.00	**803.90**	**803.90**	883.7	0.00	**803.90**	**803.90**	708.2	0.00	−0.97	0.00	0.00	**803.90**	**803.90**	123.5	0.00	−0.97	0.00	0.00
50-5-3	12	831.57	848.10	928.90	612.0	1.99	832.15	119.2	0.07	831.97	835.10	712.3	0.05	833.01	834.91	810.3	0.17	833.59	835.17	863.8	0.24	**831.57**	834.22	1084.5	0.00	−1.95	−0.07	−0.05	**831.57**	833.95	179.3	0.00	−1.95	−0.07	−0.05
50-5-3b	6	1101.57	1163.90	1198.80	531.0	5.66	1106.57	69.4	0.45	**1101.57**	1103.15	538.4	0.00	**1101.57**	1103.95	541.3	0.00	**1101.57**	1103.53	794.5	0.00	**1101.57**	1102.76	467.3	0.00	−5.36	−0.45	0.00	**1101.57**	**1101.57**	80.0	0.00	−5.36	−0.45	0.00
100-5-1	24	2000.80	2030.90	2044.30	891.0	1.50	2035.60	64.8	1.74	2004.33	2023.35	3 039.4	0.18	2010.49	2023.78	4 791.3	0.48	2008.95	2026.45	3 777.3	0.41	**2000.80**	2012.93	2474.4	0.00	−1.48	−1.71	−0.18	2005.30	2013.23	496.7	0.22	−1.26	−1.49	0.05
100-5-1b	12	2311.21	2374.90	2507.80	744.0	2.76	2357.87	102.8	2.02	2311.84	2336.64	2 221.6	0.03	2312.53	2337.27	2 871.7	0.06	2313.70	2342.80	3 126.5	0.11	**2311.21**	2346.56	2164.6	0.00	−2.68	−1.98	−0.03	2316.20	2343.44	383.1	0.22	−2.47	−1.77	0.19
100-5-2	24	1128.43	1226.10	1500.90	852.0	8.66	1144.70	81.1	1.44	1132.36	1135.99	2 760.8	0.35	1129.83	1135.49	3 976.1	0.12	1131.28	1135.70	3 582.7	0.25	**1128.43**	1133.53	2718.1	0.00	−7.97	−1.42	−0.12	1131.17	1133.01	441.3	0.24	−7.74	−1.18	0.12
100-5-2b	11	1507.88	1622.90	1701.00	855.0	7.63	1567.44	96.8	3.95	**1507.88**	1517.11	2 530.4	0.00	1510.57	1519.04	3 141.3	0.18	1510.57	1519.56	3 448.8	0.18	**1507.88**	1512.55	1540.1	0.00	−7.09	−3.80	0.00	1511.06	1514.17	215.0	0.21	−6.89	−3.60	0.21
100-5-3	24	1572.61	1710.40	1726.20	903.0	8.76	1596.77	49.2	1.54	1581.93	1587.20	2 784.3	0.59	1581.93	1586.49	4 072.3	0.59	1581.93	1587.20	3 595.7	0.59	**1572.61**	1581.49	2943.9	0.00	−8.06	−1.51	−0.59	**1572.61**	1581.17	483.5	0.00	−8.06	−1.51	−0.59
100-5-3b	11	1933.00	2054.80	2190.50	870.0	6.30	2032.13	114.9	5.13	1933.70	1950.89	2 315.8	0.04	1935.70	1953.85	2 932.6	0.14	**1933.00**	1954.97	3 247.9	0.00	1934.93	1954.27	2140.7	0.10	−5.83	−4.78	0.10	1934.93	1946.68	374.3	0.10	−5.83	−4.78	0.10
100-10-1	26	1458.80	1524.10	1589.90	1215.0	4.48	1481.56	80.3	1.56	1472.85	1511.00	2 994.7	0.96	1470.71	1503.92	4 143.4	0.82	1478.01	1510.98	3 908.8	1.32	**1458.80**	1465.01	2593.2	0.00	−4.28	−1.54	−0.81	1460.55	1465.33	455.5	0.12	−4.17	−1.42	−0.69
100-10-1b	12	1894.92	1960.70	2138.60	1185.0	3.47	1984.91	100.8	4.75	1901.27	1953.96	2 120.3	0.34	1915.77	1972.26	2 723.1	1.10	1908.63	1963.53	2 999.3	0.72	**1894.92**	1916.90	2211.6	0.00	−3.35	−4.53	−0.33	1911.89	1928.24	372.8	0.90	−2.49	−3.68	0.56
100-10-2	24	1137.59	1175.00	1236.30	1326.0	3.29	1287.50	75.8	13.18	1143.30	1152.81	2 899.7	0.50	1142.31	1155.47	4 132.2	0.41	1145.93	1155.29	3 739.8	0.73	**1137.59**	1145.91	3167.4	0.00	−3.18	−11.64	−0.41	1142.28	1145.11	483.9	0.41	−2.78	−11.28	0.00
100-10-2b	11	1561.40	1625.80	1724.20	1230.0	4.12	1645.07	139.1	5.36	1566.48	1585.67	2 208.4	0.33	1566.48	1588.24	2 889.5	0.33	1563.99	1578.90	3 115.3	0.17	**1561.40**	1570.81	2425.8	0.00	−3.96	−5.09	−0.17	**1561.40**	1565.77	410.8	0.00	−3.96	−5.09	−0.17
100-10-3	25	1204.64	1246.80	1288.30	1332.0	3.50	1216.20	57.2	0.96	1209.20	1221.52	3 145.2	0.38	1209.86	1225.98	4 290.2	0.43	1211.49	1224.72	4 060.6	0.57	**1204.64**	1209.10	1919.8	0.00	−3.38	−0.95	−0.38	1204.95	1208.05	360.6	0.03	−3.36	−0.93	−0.35
100-10-3b	11	1653.83	1799.00	1890.20	1287.0	8.78	1745.05	70.1	5.52	1662.43	1705.63	2 146.0	0.52	1665.69	1706.68	2 831.1	0.72	1670.17	1707.11	3 058.5	0.99	**1653.83**	1673.19	2334.1	0.00	−8.07	−5.23	−0.52	**1653.83**	1669.64	426.0	0.00	−8.07	−5.23	−0.52
200-10-1	49	2780.03	2920.70	3092.40	3414.0	5.06	2861.85	278.2	2.94	2798.58	2854.10	16 846.7	0.67	2797.86	2863.27	26 673.1	0.64	2803.57	2860.64	23 819.4	0.85	**2780.03**	2788.66	6664.8	0.00	−4.82	−2.86	−0.64	2785.75	2790.28	1125.8	0.21	−4.62	−2.66	−0.43
200-10-1b	22	3290.73	3532.20	3809.30	3240.0	7.34	3557.96	304.1	8.12	3368.71	3477.07	11 341.6	2.37	3355.70	3478.91	17 983.6	1.97	3327.08	3452.84	17 334.9	1.10	**3290.73**	3334.98	6272.8	0.00	−6.84	−7.51	−1.09	**3290.73**	3312.41	1090.0	0.00	−6.84	−7.51	−1.09
200-10-2	49	1972.33	2064.20	2153.30	3411.0	4.66	1997.01	342.5	1.25	1984.96	2001.97	17 482.5	0.64	1986.55	2004.51	29 590.3	0.72	1988.31	2002.48	24 959.6	0.81	**1972.33**	1981.41	5560.2	0.00	−4.45	−1.24	−0.64	1979.99	1982.13	954.4	0.39	−4.08	−0.85	−0.25
200-10-2b	23	2325.43	2516.40	2684.50	3141.0	8.21	2473.24	273.7	6.36	2336.11	2379.01	12 347.0	0.46	2355.15	2378.21	19 720.6	1.28	2368.88	2380.09	18 699.6	1.87	**2325.43**	2358.02	5160.5	0.00	−7.59	−5.98	−0.46	2341.77	2353.46	827.8	0.70	−6.94	−5.32	0.24
200-10-3	48	2727.15	2805.90	3066.1	3084.0	2.89	2783.20	323.8	2.06	2741.16	2758.09	14 786.4	0.51	2744.67	2757.42	27 900.5	0.64	2751.23	2763.79	21 623.8	0.88	**2727.15**	2736.49	5479.3	0.00	−2.81	−2.01	−0.51	**2727.15**	2736.07	789.2	0.00	−2.81	−2.01	−0.51
200-10-3b	22	3190.34	3347.00	3454.60	3267.0	4.91	3413.34	281.4	6.99	3242.18	3274.58	9 511.1	1.62	3233.89	3267.81	16 534.2	1.37	3225.75	3273.68	14 933.5	1.11	**3190.34**	3218.10	3924.1	0.00	−4.68	−6.53	−1.10	3208.88	3219.06	604.2	0.58	−4.13	−5.99	−0.52
Global avg.		1494.50	1564.42	1610.33	1272.6	4.03	1546.35	**121.7**	2.59	1502.98	1521.25	3 975.5	0.37	1503.92	1522.28	6 248.6	0.43	1503.77	1521.76	5 692.4	0.45	**1494.82**	1504.82	2292.4	**0.05**	**−3.77**	**−2.40**	**−0.24**	1497.73	**1503.61**	382.1	0.19	−3.63	−2.26	−0.09

Table 18
Detailed results for the LLRP Barreto data set.

Instance	N_v	BKS	MA				GBILS			SA-VND0				SA-VND1				SA-VND2				M-ILS 30 runs						M-ILS 5 runs					
			Best	Avg	Time	gap_B	Best	Time	gap_B	Best	Avg	Time	gap_B	Best	Avg	Time	gap_B	Best	Avg	Time	gap_B	Best	Avg	Time	gap_B	gap_{MA}	gap_{SA}	Best	Avg	Time	gap_B	gap_{MA}	gap_{SA}
Christ-50-5	6	1 661.64	1 690.80	1 782.40	591.0	1.75	–	–	–	**1 661.64**	1 662.07	541.5	0.00	**1 661.64**	1 662.13	528.7	0.00	**1 661.64**	1 662.35	813.7	0.00	**1 661.64**	1 669.05	614.2	0.00	−1.72	0.00	**1 661.64**	1 663.39	107.1	0.00	−1.72	0.00
Christ-75-10	9	2 370.73	2 590.30	2 689.80	873.0	9.26	–	–	–	2 403.79	2 459.03	1159.1	1.39	2 383.04	2 457.45	1384.4	0.52	2 408.92	2 454.95	1588.4	1.61	**2 370.73**	2 409.75	1612.4	0.00	−8.48	−0.52	2 391.95	2 406.42	284.5	0.90	−7.66	0.37
Christ-100-10	8	3 791.98	4 058.20	4 194.90	1023.0	7.02	3984.05	451.9	5.07	**3 791.98**	3 831.18	1958.7	0.00	3 806.39	3 838.89	2293.6	0.38	3 795.15	3 825.37	2876.3	0.08	3 803.50	3 845.85	2110.8	0.30	−6.28	0.30	3 825.29	3 842.92	360.7	0.88	−5.74	0.88
Gaskell-21-5	4	653.48	658.40	741.10	441.0	0.75	**653.48**	0.9	0.00	**653.48**	**653.48**	116.4	0.00	**653.48**	**653.48**	102.2	0.00	**653.48**	**653.48**	169.5	0.00	**653.48**	**653.48**	143.0	0.00	−0.75	0.00	**653.48**	**653.48**	23.1	0.00	−0.75	0.00
Gaskell-29-5	4	1 199.33	1 224.50	1 296.30	468.0	2.10	**1199.33**	5.2	0.00	**1 199.33**	**1 199.33**	311.0	0.00	**1 199.33**	**1 199.33**	255.4	0.00	**1 199.33**	**1 199.33**	485.9	0.00	**1 199.33**	**1 199.33**	283.2	0.00	−2.06	0.00	**1 199.33**	**1 199.33**	46.7	0.00	−2.06	0.00
Gaskell-32-5b	3	1 552.84	1 571.00	1 668.40	483.0	1.17	**1552.84**	4.9	0.00	**1 552.84**	1 553.29	417.8	0.00	**1 552.84**	1 553.29	337.6	0.00	**1 552.84**	1 553.29	650.7	0.00	**1 552.84**	1 556.58	284.4	0.00	−1.16	0.00	**1 552.84**	1 555.52	43.7	0.00	−1.16	0.00
Gaskell-36-5	4	1 627.17	1 642.40	1 647.00	522.0	0.94	**1627.17**	3.2	0.00	**1 627.17**	**1 627.17**	308.3	0.00	**1 627.17**	**1 627.17**	274.0	0.00	**1 627.17**	**1 627.17**	465.8	0.00	**1 627.17**	1 628.12	369.1	0.00	−0.93	0.00	**1 627.17**	**1 627.17**	61.0	0.00	−0.93	0.00
Min-27-5	4	5 387.55	**5 387.55**	5 697.00	834.0	0.00	**5387.55**	84.0	0.00	**5 387.55**	**5 387.55**	176.3	0.00	**5 387.55**	**5 387.55**	147.4	0.00	**5 387.55**	**5 387.55**	267.0	0.00	**5 387.55**	**5 387.55**	134.0	0.00	0.00	0.00	**5 387.55**	**5 387.55**	23.8	0.00	0.00	0.00
Min-134-8	11	21 752.00	23 387.00	26 012.50	2220.0	7.52	–	–	–	21 852.40	22 307.28	2225.7	0.46	21 910.50	22 309.20	2624.0	0.73	21 881.80	22 278.05	3250.7	0.60	**21 752.00**	22 442.41	2127.0	0.00	−6.99	−0.46	21 865.00	22 297.08	355.4	0.52	−6.51	0.06
Or-117-14	7	53 798.50	56 209.00	61 396.20	1545.0	4.48	–	–	–	**53 798.50**	54 866.72	1263.1	0.00	53 859.10	54 805.88	1265.5	0.11	**53 798.50**	54 905.77	1958.4	0.00	54 413.90	56 794.12	1598.2	1.14	−3.19	1.14	56 228.40	57 517.50	273.0	4.52	0.03	4.52
Global avg.		9 379.52	9 841.92	10 712.56	900.0	3.50	–	–	–	**9 392.87**	9 554.71	847.8	0.19	9 404.10	**9 549.44**	921.3	0.17	9 396.64	9 554.73	1252.6	0.23	9 442.21	9 758.62	927.6	**0.14**	−4.06	**0.05**	9 639.26	9 815.04	**157.9**	0.68	−2.06	0.58
Global avg GBILS		2 368.72	2 423.68	2 540.78	628.5	2.00	2400.74	**91.7**	0.84	**2 368.72**	2 375.33	548.1	**0.00**	2 371.13	2 376.62	568.4	0.06	2 369.25	**2 374.36**	819.2	0.01	2 370.64	2 378.48	554.1	0.05	**−3.24**	**0.05**	2 374.28	2 377.66	93.1	0.15	−2.59	0.15

algorithms presented in Osorio-Mora et al. (2023). Nevertheless, M-ILS presents the smaller global value for gap_B (0.14%) among all the algorithms.

Although the global computing time required by M-ILS, considering 30 runs, is larger than the one required by GBILS, the average computing times of both algorithms (considering a single run) are similar. Thus, by comparing M-ILS and GBILS, both considering 5 runs, the computing times are equivalent, and both algorithms find the proved optimal solution for 5 instances. Nevertheless, for the instance Christ-100-10 M-ILS provides a better solution value than GBILS with a gap_{GBILS} equal to −3.98%. Finally, regarding MA, M-ILS outperforms it regarding the solution quality in similar computing times. M-ILS provides a better solution value than MA for all the instances but one in which both algorithms provide the same optimal solution value. Furthermore, the average solution value provided by M-ILS is better than the one provided by MA for all the instances. The average value of gap_{MA} equals −3.24% and −2.59% when 30 and 5 runs are considered for M-ILS, respectively. Indeed, M-ILS outperforms MA, even considering 5 runs, in global computing times almost 6 times smaller.

3.5.4. *Overall results on the LLRP*

After analyzing the results, we can state that the proposed algorithm M-ILS outperforms the state-of-the-art algorithms regarding the solution quality for all the considered data sets. The global average value of gap_B (computed over all the instances) is equal to 0.06% and 0.32% when M-ILS is executed with 30 and 5 runs, respectively. This value is better than the best among all the competitors (0.53% for SA-VND0). Comparing M-ILS to the algorithms proposed in Osorio-Mora et al. (2023), the global average value of gap_{SA} equals −0.33% and −0.07% when M-ILS is executed with 30 and 5 runs, respectively. Considering 30 runs, M-ILS is globally faster than the algorithms SA-VND0, SA-VND1, and SA-VND2 for the two more complex data sets. In contrast, all the algorithms require similar computing times in the third data set (Barreto data set). The average global computing time is 2786.95 s for M-ILS and 4400.23 s for SA-VND0 (the fastest among the state-of-the-art algorithms) when 30 runs are executed for both algorithms. On the other hand, comparing M-ILS to GBILS, the global average value of gap_{GBILS} (computed over all the instances reported for GBILS) is equal to −2.13% and −2.00% when M-ILS is executed with 30, and 5 runs, respectively. Finally, comparing M-ILS to MA, the global average value of gap_{MA} (computed over all the instances reported for MA) equals −4.35% and −4.07% when M-ILS is executed with 30 and 5 runs, respectively.

It is to note that for this problem we do not present the computational experiments regarding the comparison of the computing times required by the considering algorithms to reach the target values. The reason is that most of the values BKS_0 were provided for one of the three metaheuristics presented in Osorio-Mora et al. (2023), and since M-ILS outperforms these algorithms, finding better solution values within shorter computing times, considering a time limit larger than the computing time reported for M-ILS would lead to redundant conclusions. The same situation applies if a target value worse than the best solution value obtained by M-ILS is considered. Furthermore, for the instances in the Barreto data set, M-ILS (considering 5 runs) was able to find the same solution values as those obtained by GBILS (the fastest among the heuristics) in equivalent computing times for all the instances but one, in which M-ILS finds a better solution.

4. Conclusions and future research

An effective metaheuristic (M-ILS) is proposed for solving the MD-CCVRP, the MDk-TRP, and the LLRP. The algorithm was tested on several benchmark data sets, with a total of 78 instances for the MD-CCVRP, 87 instances for the MDk-TRP, and 76 instances for the LLRP. Extensive computational experiments show that M-ILS outperforms in terms of solution quality the state-of-the-art metaheuristic algorithms PLS (proposed in Wang et al. (2020) for the MDCCVRP), GA (proposed in Bruni et al. (2022a) for the MDk-TRP), and SA-VND0, SA-VND1, and SA-VND2 (proposed in Osorio-Mora et al. (2023) for the LLRP), with competitive computing times.

The experiments also show that the stability of the proposed metaheuristic allows for a reduction of the number of runs necessary to provide good-quality solutions, implying global computing times shorter than those required by the currently published heuristic methods. Indeed, when M-ILS is executed with a time limit equal to that required by PLS and GA for the MDCCVRP and the MDk-TRP, respectively, M-ILS is able to find solution values better than those obtained by the mentioned competitors. For the LLRP, M-ILS requires shorter computing times and finds better quality solutions than those corresponding to the algorithms SA-VND0, SA-VND1, and SA-VND2, considering the same number of runs (30). Indeed, M-ILS outperforms the mentioned algorithms even when the number of runs considered is much smaller (5).

According to the results reported in Section 3, the proposed metaheuristic is globally the most effective algorithm for the considered problems when challenging instances, in which the number of customers is large, and the number of vehicles is relatively small, must be solved. In addition, the proposed algorithm provides optimal or near-optimal solution values for the easiest instances.

Based on the obtained results, it is possible to suggest the application of the proposed methodology to other related problems as future research directions. Some examples could be extensions of single-depot latency vehicle routing problems studied in the literature, for example, including time windows (generalizing the problem studied in Kyriakakis et al. (2022)), considering priorities for the customers (generalizing the problem studied in Bruni et al. (2020)), or combining truck and drones in last-mile delivery operations (generalizing the problem studied in Bruni et al. (2022b)). Also, since M-ILS can provide good quality solutions for large-size instances within short computing times compared to those required by the exact methods, M-ILS could be useful to solve real-life problems related to humanitarian logistics (Bruni et al., 2020; Ajam et al., 2022) or other applications. Another interesting variant of the problem (especially suitable for post-disaster management) could consider pickup and delivery decisions, where some vital products, such as water and food, must be delivered, while injured people must be picked-up.

CRediT authorship contribution statement

Alan Osorio-Mora: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **John Willmer Escobar:** Methodology, Formal analysis, Writing – original draft, Writing – review & editing, Visualization. **Paolo Toth:** Conceptualization, Validation, Resources, Writing – original draft, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Data availability

Data will be made available on request.

Acknowledgments

This work is partially supported by the National Agency for Research and Development (ANID)/Scholarship Program/Doctorado Becas Chile/2020-72210275. We also acknowledge Air Force Office of Scientific Research Grant FA8655-20-1-7019.

Appendix. Optimal solution for the MDk-TRP instances with $N_v = 35$

Optimal Solution Instance p01 (see Table A.19).
Optimal Solution Instance p12 (see Table A.20).

Table A.19

Route 1	**Route 6**	**Route 11**	**Route 16**	**Route 21**	**Route 26**	**Route 31**
D1 - 13	D3 - 38	D3 - 16	D1 - 15	D3 - 30	D1 - 41	D1 - 17 - 37
Route 2	**Route 7**	**Route 12**	**Route 17**	**Route 22**	**Route 27**	**Route 32**
D4 - 21	D2 - 1	D2 - 23 - 43	D1 - 4 - 18	D4 - 22	D1 - 42	D3 - 49 - 33
Route 3	**Route 8**	**Route 13**	**Route 18**	**Route 23**	**Route 28**	**Route 33**
D4 - 35 - 36	D1 - 25	D3 - 10	D3 - 39	D2 - 48 - 7	D2 - 6 - 24	D2 - 46 - 11
Route 4	**Route 9**	**Route 14**	**Route 19**	**Route 24**	**Route 29**	**Route 34**
D2 - 14	D2 - 26	D2 - 12	D2 - 32	D3 - 5	D1 - 44 - 45	D3 - 34
Route 5	**Route 10**	**Route 15**	**Route 20**	**Route 25**	**Route 30**	**Route 35**
D4 - 29 - 2	D4 - 20 - 3	D2 - 27 - 8	D2 - 47	D3 - 9 - 50	D4 - 28 - 31	D1 - 19 - 40

Table A.20

Route 1	**Route 8**	**Route 15**	**Route 22**	**Route 29**
D1 - 26	D1 - 5 - 13	D1 - 28 - 36	D2 - 42 - 50 - 58 - 66 - 74	D1 - 14
Route 2	**Route 9**	**Route 16**	**Route 23**	**Route 30**
D1 - 2 - 18	D2 - 44 - 52 - 60 - 68 - 76	D1 - 6 - 22 - 30	D2 - 45 - 53 - 61 - 69 - 77	D1 - 38
Route 3	**Route 10**	**Route 17**	**Route 24**	**Route 31**
D1 - 10	D1 - 9	D1 - 34	D1 - 39	D1 - 27
Route 4	**Route 11**	**Route 18**	**Route 25**	**Route 32**
D1 - 4 - 12 - 20	D2 - 46 - 54 - 62 - 70 - 78	D1 - 3	D2 - 51 - 67	D2 - 47
Route 5	**Route 12**	**Route 19**	**Route 26**	**Route 33**
D2 - 65	D1 - 7 - 15 - 23 - 31	D2 - 41 - 49 - 57 - 73	D1 - 11 - 19 - 35	D2 - 71 - 79
Route 6	**Route 13**	**Route 20**	**Route 27**	**Route 34**
D2 - 48 - 56 - 64 - 72 - 80	D1 - 40	D1 - 8 - 16 - 24 - 32	D2 - 59 - 75	D2 - 55 - 63
Route 7	**Route 14**	**Route 21**	**Route 28**	**Route 35**
D1 - 1	D1 - 17 - 33	D1 - 25	D1 - 21 - 29 - 37	D2 - 43

References

Ajam, M., Akbari, V., Salman, F.S., 2022. Routing multiple work teams to minimize latency in post-disaster road network restoration. European J. Oper. Res. 300 (1), 237–254.

Ángel-Bello, F., Alvarez, A., García, I., 2013. Two improved formulations for the minimum latency problem. Appl. Math. Model. 37 (4), 2257–2266.

Ángel-Bello, F., Cardona-Valdés, Y., Álvarez, A., 2019. Mixed integer formulations for the multiple minimum latency problem. Oper. Res. 19 (2), 369–398.

Blum, A., Chalasani, P., Coppersmith, D., Pulleyblank, B., Raghavan, P., Sudan, M., 1994. The minimum latency problem. In: Proceedings of the Annual ACM Symposium on Theory of Computing, Vol. Part F129502. Association for Computing Machinery, pp. 163–171. http://dx.doi.org/10.1145/195058.195125.

Bruni, M.E., Khodaparasti, S., 2022. A variable neighborhood descent matheuristic for the drone routing problem with beehives sharing. Sustainability 14 (16), 9978.

Bruni, M., Khodaparasti, S., Beraldi, P., 2020. The selective minimum latency problem under travel time variability: An application to post-disaster assessment operations. Omega 92, 102154.

Bruni, M.E., Khodaparasti, S., Martínez-Salazar, I., Nucamendi-Guillén, S., 2022a. The multi-depot k-traveling repairman problem. Optim. Lett. 16, 2681–2709.

Bruni, M., Khodaparasti, S., Moshref-Javadi, M., 2022b. A logic-based Benders decomposition method for the multi-trip traveling repairman problem with drones. Comput. Oper. Res. 145, 105845.

Corona-Gutiérrez, K., Nucamendi-Guillén, S., Lalla-Ruiz, E., 2022. Vehicle routing with cumulative objectives: a state of the art and analysis. Comput. Ind. Eng. 108054.

Damião, C.M., Silva, J.M.P., Uchoa, E., 2021. A branch-cut-and-price algorithm for the cumulative capacitated vehicle routing problem. 4OR 1–25.

Escobar, J.W., Linfati, R., Baldoquin, M.G., Toth, P., 2014a. A granular variable tabu neighborhood search for the capacitated location-routing problem. Transp. Res. B 67, 344–356.

Escobar, J.W., Linfati, R., Toth, P., 2013. A two-phase hybrid heuristic algorithm for the capacitated location-routing problem. Comput. Oper. Res. 40 (1), 70–79.

Escobar, J.W., Linfati, R., Toth, P., Baldoquin, M.G., 2014b. A hybrid granular tabu search algorithm for the multi-depot vehicle routing problem. J. Heuristics 20 (5), 483–509.

Fakcharoenphol, J., Harrelson, C., Rao, S., 2007. The k-traveling repairmen problem. ACM Trans. Algorithms (TALG) 3 (4), 40–es.

Fischetti, M., Laporte, G., Martello, S., 1993. Delivery man problem and cumulative matroids. Oper. Res. 41 (6), 1055–1064. http://dx.doi.org/10.1287/opre.41.6.1055, URL https://pubsonline.informs.org/doi/abs/10.1287/opre.41.6.1055.

Gurobi Optimization, L., 2021. Gurobi optimizer reference manual.

Helsgaun, K., 2017. An Extension of the Lin-Kernighan-Helsgaun TSP Solver for Constrained Traveling Salesman and Vehicle Routing Problems. Roskilde University.

IBM, 2021. IBM ILOG CPLEX 20.1 user's manual.

Kara, İ., Kara, B.Y., Yetiş, M.K., 2008. Cumulative vehicle routing problems. In: Vehicle Routing Problem. In-Teh, pp. 85–98.

Kyriakakis, N.A., Sevastopoulos, I., Marinaki, M., Marinakis, Y., 2022. A hybrid Tabu search–Variable neighborhood descent algorithm for the cumulative capacitated vehicle routing problem with time windows in humanitarian applications. Comput. Ind. Eng. 164, 107868.

Lalla-Ruiz, E., Voß, S., 2020. A POPMUSIC approach for the multi-depot cumulative capacitated vehicle routing problem. Optim. Lett. 14 (3), 671–691. http://dx.doi.org/10.1007/s11590-018-1376-1.

Lin, S., Kernighan, B.W., 1973. An effective heuristic algorithm for the traveling-salesman problem. Oper. Res. 21 (2), 498–516.

López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Birattari, M., Stützle, T., 2016. The irace package: Iterated racing for automatic algorithm configuration. Oper. Res. Perspect. 3, 43–58.

Lourenço, H.R., Martin, O.C., Stützle, T., 2019. Iterated local search: Framework and applications. In: Handbook of Metaheuristics. Springer, pp. 129–168.

Martello, S., Toth, P., 1990. Bin-packing problem. In: Knapsack Problems: Algorithms and Computer Implementations. John Wiley & Sons, Inc., pp. 221–245.

Mattos Ribeiro, G., Laporte, G., 2012. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. Comput. Oper. Res. 39 (3), 728–735. http://dx.doi.org/10.1016/j.cor.2011.05.005.

Montoya-Torres, J.R., Franco, J.L., Isaza, S.N., Jiménez, H.F., Herazo-Padilla, N., 2015. A literature review on the vehicle routing problem with multiple depots. Comput. Ind. Eng. 79, 115–129.

Moshref-Javadi, M., Lee, S., 2016. The latency location-routing problem. European J. Oper. Res. 255 (2), 604–619.

Ngueveu, S.U., Prins, C., Wolfler Calvo, R., 2010. An effective memetic algorithm for the cumulative capacitated vehicle routing problem. Comput. Oper. Res. 37 (11), 1877–1885. http://dx.doi.org/10.1016/j.cor.2009.06.014.

Nucamendi-Guillén, S., Angel-Bello, F., Martínez-Salazar, I., Cordero-Franco, A.E., 2018. The cumulative capacitated vehicle routing problem: New formulations and iterated greedy algorithms. Expert Syst. Appl. 113, 315–327. http://dx.doi.org/10.1016/j.eswa.2018.07.025.

Nucamendi-Guillén, S., Martínez-Salazar, I., Angel-Bello, F., Moreno-Vega, J.M., 2016. A mixed integer formulation and an efficient metaheuristic procedure for the k-travelling repairmen problem. J. Oper. Res. Soc. 67 (8), 1121–1134.

Nucamendi-Guillén, S., Martínez-Salazar, I., Khodaparasti, S., Bruni, M.E., 2022. New formulations and solution approaches for the latency location routing problem. Comput. Oper. Res. 143, 105767.

Osorio-Mora, A., Rey, C., Toth, P., Vigo, D., 2023. Effective metaheuristics for the latency location routing problem. Int. Trans. Oper. Res..

Sze, J.F., Salhi, S., Wassan, N., 2017. 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. B 101, 162–184. http://dx.doi.org/10.1016/j.trb.2017.04.003.

Tsitsiklis, J.N., 1992. Special cases of traveling salesman and repairman problems with time windows. Networks 22 (3), 263–282. http://dx.doi.org/10.1002/net.3230220305, URL https://onlinelibrary.wiley.com/doi/full/10.1002/net.3230220305.

Wang, X., Choi, T.-M., Li, Z., Shao, S., 2020. An effective local search algorithm for the multidepot cumulative capacitated vehicle routing problem. IEEE Trans. Syst. Man Cybern.: Syst. 1–11. http://dx.doi.org/10.1109/tsmc.2019.2938298.

Wang, X., Choi, T.-M., Liu, H., Yue, X., 2016. A novel hybrid ant colony optimization algorithm for emergency transportation problems during post-disaster scenarios. IEEE Trans. Syst. Man Cybern.: Syst. 48 (4), 545–556.

Zhong, S., Cheng, R., Jiang, Y., Wang, Z., Larsen, A., Nielsen, O.A., 2020. Risk-averse optimization of disaster relief facility location and vehicle routing under stochastic demand. Transp. Res. E 141, 102015.