Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Exact Solution of Several Families of Location-Arc Routing Problems

Elena Fernández, Gilbert Laporte, Jessica Rodríguez-Pereira

To cite this article:
Elena Fernández, Gilbert Laporte, Jessica Rodríguez-Pereira (2019) Exact Solution of Several Families of Location-Arc Routing Problems. Transportation Science

Published online in Articles in Advance 02 Jul 2019

. https://doi.org/10.1287/trsc.2018.0881

informs
http://pubsonline.informs.org/journal/trsc/

TRANSPORTATION SCIENCE
Articles in Advance, pp. 1–21
ISSN 0041-1655 (print), ISSN 1526-5447 (online)

Exact Solution of Several Families of Location-Arc Routing Problems

Elena Fernández,[a,b] Gilbert Laporte,[c] Jessica Rodríguez-Pereira[a]

[a] Department of Statistics and Operations Research, Polytechnic University of Catalonia, 08034 Barcelona, Spain; [b] Barcelona Graduate School of Mathematics, 08193 Bellaterra, Spain; [c] Department of Decision Sciences, HEC Montréal, Montreal, Quebec H3T 2A7, Canada

Contact: e.fernandez@upc.edu, http://orcid.org/0000-0003-4714-0257 (EF); gilbert.laporte@cirrelt.ca (GL); jessica.rodriguez@upc.edu, http://orcid.org/0000-0002-9726-7531 (JR-P)

Received: September 25, 2017
Revised: March 21, 2018; September 4, 2018
Accepted: September 24, 2018
Published Online in Articles in Advance: July 2, 2019

https://doi.org/10.1287/trsc.2018.0881

Abstract. We model and solve several families of location-arc routing problems on an undirected graph. These problems extend the multidepot rural postman problem to the case where the depots are not fixed. The aim is to select the facility locations and to construct a set of routes traversing each required edge of the graph, where each route starts and ends at the same facility. The models differ from each other in their objective functions and on whether they include a capacity constraint. Alternative formulations are presented that use only binary variables, and are valid even when the input graph is not complete. This applies, in particular, to a compact two-index formulation for problems minimizing the overall routing costs, with or without facility setup costs. This formulation incorporates a new set of constraints that force the routes to be consistent and return to their original depots. A polyhedral study is presented for some of the formulations, which indicates that the main families of constraints are facet defining. All formulations are solved by branch and cut, and instances with up to 200 vertices are solved to optimality. Despite the difficulty of the problems, the numerical results demonstrate the good performance of the algorithm.

Funding: This research was partially supported by the Spanish Ministry of Economy and Competitiveness [Grants EEBB-I-16-10670, BES-2013-063633, MTM2015-63779-R, and MDM-2014-044 (MINECO/FEDER)] and the Canadian Natural Sciences and Engineering Research Council [Grant 2015-06189].
Supplemental Material: The online appendix is available at https://doi.org/10.1287/trsc.2018.0881.

Keywords: arc routing • location • polyhedral analysis • facets • branch and cut

1. Introduction

Location-arc routing problems (LARPs) combine location and routing decisions in contexts where the arcs of a network must be serviced, as opposed to the nodes. These problems arise in most of the classical arc routing problems (ARPs) such as newspaper delivery, garbage collection, road gritting, snow removal, meter reading, etc. (see, e.g., Corberán and Laporte 2014, chapters 13–16). In such problems, the objective function is typically the total routing cost or the makespan, that is, the length of the longest route. LARPs were formally introduced by Ghiani (1998), but an earlier publication by Levy and Bodin (1989) described an application in the U.S. Postal Service in which a postman parks his van in several locations from which he proceeds to deliver mail on foot.

LARPs are the arc routing counterpart of location routing problems occurring in node routing contexts (for surveys, see Min, Jayaraman, and Srivastava 1998; Nagy and Salhi 2007; Drexl, Schneider, and Scheneker 2013; Prodhon and Prins 2014; Albareda-Sambola 2015) but have been less extensively studied. According to Albareda-Sambola (2015), this may be because ARPs can often be transformed into node routing problems, as in Baldacci and Maniezzo (2006); Longo, Poggi de Aragão, and Uchoa (2006); and Pearn, Assad, and Golden (1987). To the best of our knowledge, Ghiani and Laporte (1999) and Arbib et al. (2014) present the only exact algorithms for uncapacitated LARPs. Ghiani and Laporte (1999) reduce the original problem to an undirected rural postman problem (RPP) and solve it by an exact branch-and-cut algorithm. Arbib et al. (2014) present a mathematical programming formulation and a branch-and-cut algorithm for a directed profitable LARP in which the facilities are located at both end points of the selected arcs according to the facility opening costs, to the profit collected on these arcs, and to the cost of traversing them.

Some authors have focused on capacitated LARPs. Thus, Hashemi Doulabi and Seifi (2013) presented two formulations on mixed graphs: one for the general case and one for the case of a single facility. They proposed a simulated annealing heuristic incorporating several arc routing heuristics. Lopes et al. (2014) presented a four-index flow formulation as well as constructive heuristics, classical improvement heuristics, and metaheuristics. Several authors have studied extensions of the capacitated arc routing problem (CARP) with a location component. In Ghiani, Improta, and Laporte (2001), these are intermediate facilities at which vehicles such as garbage trucks can unload to not exceed their capacity. Pia and Filippi (2006) considered a CARP with mobile depots, and Amaya, Langevin, and Trépanier (2007) solved a CARP in which extra vehicles replenish the main fleet at meeting points to be located. The authors formulated the problem and solved it by branch and cut. Salazar-Aguilar, Langevin, and Laporte (2013) studied a related problem in the context of road marking.

The purpose of this paper is to study, model, and solve exactly several families of LARPs defined on undirected graphs. We develop models that differ from each other in their objective functions, on whether the number of facilities to be located is upper bounded, or on whether the facilities are capacitated. In particular, we consider two types of objective functions: min-cost objectives aiming at minimizing the overall routing costs, and min–max objectives aiming at minimizing the makespan. Whereas some of the models assume that there are no capacity limitations, we also study problems that include a cardinality constraint on the number of users that can be served from an open facility. Finally, some of the models ignore facility setup costs but include a limitation on the maximum number of facilities to be located, whereas in other models, the number of open facilities is not limited, but the facility setup costs are included in the objective function. Dealing with both types of models allows us to analyze the trade-off between models with a simple objective function, focusing only on routing costs but requiring a cardinality constraint on the number of facilities, and models without such constraint but with a richer objective function with setup costs for the open facilities.

To a large extent, this work extends our previous works on the multidepot RPP (MDRPP; Fernández and Rodríguez-Pereira 2017; Fernández, Laporte, and Rodríguez-Pereira 2018), where we proposed exact solution algorithms based on three- and two-index formulations for the arc routing problem in which the set of depots for the routes is given. As we will see, when location decisions are incorporated into arc routing problems, several nontrivial extensions of the MDRPP arise.

This paper makes the following scientific contributions:

- We study six LARP models (see Table 1), discuss their modeling assumptions, and derive optimality conditions.
- We present two types of formulations. The first class uses *disaggregated* decision variables (*three-index variables*) that link routes with open facilities. All models can be handled with this type of formulation. The second class of formulations *aggregates* the information of all the routes. This leads to *two-index variables*, associated with the edges traversed by the routes, but that do not explicitly link them to the facilities from which the routes operate. This approach indeed reduces the number of required variables at the expense of presenting some additional difficulties.
- The formulations that we study exploit optimality conditions, which allow the use of binary variables only. Preliminary testing showed that for the problems that we study, such formulations clearly outperform those that do not exploit optimality conditions, producing tighter lower bounds and smaller enumeration trees.
- We perform a polyhedral study for the disaggregated three-index formulations (3IFs), and we prove that the main families of constraints are facet defining. To the best of our knowledge, no polyhedral study has ever been carried out for three-index variable formulations of arc routing problems with multiple depots, with or without location decisions.
- For the Min-Cost- (MC)-p-LARP and MC-LARP models, we exploit of the optimal condition on location variables to reinforce the three-index formulation.
- For the compact two-index formulation (2IF), we prove that there exists an optimal solution in which no edge is traversed more than twice. As a consequence of this optimality condition, the two-index formulation is valid even when the input graph is not complete. This is an important difference with the MDRPP, for which this optimality condition does not hold, and where two-index formulations are valid only when the input graph is complete.
- For the two-index formulation, we present a new set of constraints guaranteeing that the routes are consistent and return to their original depots. These

Table 1. Summary of Models

	Objective function	Capacity	Limit on the number of open facilities
MC-p-LARP	Min routing cost	No	Yes
MM-p-LARP	Min makespan	No	Yes
MC-LARP	Min facility setup cost plus routing cost	No	No
MC-p-LARP-UD	Min routing cost	Yes	Yes
MM-p-LARP-UD	Min makespan	Yes	Yes
MC-LARP-UD	Min facility setup cost plus routing cost	Yes	No

inequalities incorporate location decisions and cannot be immediately derived from the case of the MDRPP.

The remainder of this paper is organized as follows. Section 2 contains formal definitions of the problems. The mathematical models are presented in Section 3, followed by the branch-and-cut algorithm in Section 4. Extensive computational results are presented in Section 5. This paper closes with some conclusions in Section 6.

2. Location-Arc Routing Problems

We consider LARPs defined on an undirected connected graph $G = (V, E)$, where V is the vertex set, $|V| = n$, and E is the edge set, with $|E| = m$. The set $D \subset V$ denotes a set of potential locations where facilities may be established. A given set $R \subset E$ of edges must be traversed (*served*), which are referred to as *required edges*. The connected components induced by the required edges are referred to as *required components* and are denoted by $C_k = (V_k, R_k)$, $k \in K$. Hence, $R = \bigcup_{k \in K} R_k$. Also let $V_R = \bigcup_{k \in K} V_k$. There is a traversal cost $c_e \geq 0$ associated with each edge $e \in E$, and a value $f_d \geq 0$ associated with each potential location $d \in D$, which indicates the setup cost of *opening* a facility at d. Let p be an upper bound on the number of facilities to be located. When there is a limitation on the service capacity of open facilities, we use b_d to denote the maximum number of required edges that can be served from a facility located at $d \in D$. We use the term *route* to denote a closed walk that starts and ends at a selected location $d \in D$. We say that a required edge $e \in R$ is served if a route traverses it at least once. The cost of a route is the sum of the costs of edges, where the cost of each edge is counted as many times as it is traversed.

Feasible LARP solutions consist of a subset of open facilities $D^* \subseteq D$, together with a set of nonempty routes, at least one for each selected facility, that serve all the required edges. Alternative objective functions or additional constraints characterize the different problems under study.

Definition 2.1.

a. The MC-p-LARP is to determine a feasible solution with at most p open facilites, that is, $|D^*| \leq p$, that minimizes the sum of the routing costs.

b. The Min-Makespan- (MM)MM-p-LARP is to determine feasible solution with at most p open facilites, that is, $|D^*| \leq p$, that minimizes the makespan.

c. The MC-LARP is to determine a feasible solution that minimizes the sum of the setup costs of the selected facilities plus the routing costs.

We also consider capacitated versions of each of the above defined problems, where we assume that each required edge has a unit demand (UD), and for each potential facility, there is a constraint on the maximum demand that it can serve if it is opened. Because we consider unit demands, these capacitated versions reduce to cardinality constraints on the maximum number of required edges served by each facility. We denote by MC-p-LARP-UD, MM-p-LARP-UD, and MC-LARP-UD the capacitated versions of MC-p-LARP, MM-p-LARP, and MC-LARP, respectively.

The MC-MDRPP, where the location of the facilities is known in advance, is a particular case of both the MC-p-LARP and the MC-LARP. Moreover, the MC-MDRPP is also a particular case of the MC-p-LARP-UD and the MC-LARP-UD, where the location of the facilities are known and there are no facilities capacity constraints. Similarly, the MM-MDRP is a particular case of both the MM-p-LARP and the MM-p-LARP-UD. Because the MC-MDRPP and the MM-MDRP are known to be NP-hard (Fernández and Rodríguez-Pereira 2017), we can state the following proposition.

Proposition 2.1.

- *The MC-p-LARP and the MC-p-LARP-UD are NP-hard.*
- *The MM-p-LARP and the MM-p-LARP-UD are NP-hard.*
- *The MC-LARP and the MC-LARP-UD are NP-hard.*

In the remainder of this paper, we assume that G has been simplified so that V is the set of vertices incident to the edges of R plus the set of potential locations D, that is, $V = V_R \cup D$. The set E contains the edges of R plus additional unrequired edges connecting every pair of vertices and representing shortest paths in the original graph. To this end, following the procedure described in Christofides et al. (1981), we first add to $G_R = (V_R \cup D, R)$ an edge between every pair of vertices of $V_R \cup D$ having a cost equal to the shortest path length on G. We then remove all unrequired edges (i, j) for which $c_{ij} = c_{ik} + c_{kj}$ for some $k \in V$, and one of two parallel edges whenever they both have the same cost. Hence, the costs of the simplified graph satisfy the triangle inequality.

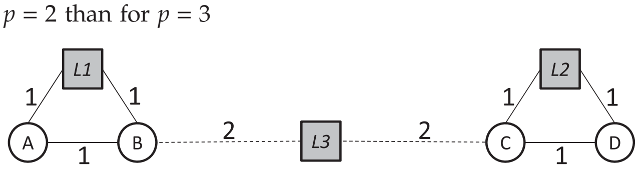
Without loss of generality, we also assume that $|D| \geq 3$. Indeed, if $|D| = 1$, no location decision must be made, so we just have an arc routing problem. If $|D| = 2$, we can define an additional potential location placed at a fictitious node and connect it with only one vertex of V_R with an edge of cost greater than twice the sum of the costs of all other edges. This hypothesis will be used in the proofs of our polyhedral analysis, where we sometimes use three different depots to obtain the number of affinely independent points of the studied polyhedron that are needed.

We denote by T_C a minimum spanning tree with respect to cost function c of the multigraph $G_C = (V_C, E_C)$ induced by the connected components plus the potential locations that do not belong to any component $D \setminus V_R$. In addition, we will use the following usual notation. For any nonempty vertex subset $S \subset V$, $\delta(S) = \{(u,v) \in E | u \in S, v \in V \setminus S\} = \delta(V \setminus S)$ is the set edges in the cut between S and $V \setminus S$, and $\gamma(S) = \{(u,v) \in E | u, v \in S\}$ is the set of edges with both vertices in S. In particular, for $k \in K$, we use the notation $E_k = \gamma(V_k) \supseteq R_k$. For a singleton $S = \{v\}$, with $v \in V$, we simply write $\delta(v)$ instead of $\delta(\{v\})$. For $H \subset E$, we use $\delta_H(S) = \delta(S) \cap H$ and $\gamma_H(S) = \gamma(S) \cap H$. Furthermore, a vertex $v \in V$ is H-odd if $|\delta_H(v)|$ is odd; otherwise, v is H-even. Finally, we use the standard compact notation $f(A) \equiv \sum_{e \in A} f_e$, where f is a vector defined over a set Ω and $A \subseteq \Omega$. Thus, if x is a vector defined on the edge set E and $H \subseteq E$, then $x(H) = \sum_{e \in H} x_e$. Similarly, if z is a vector defined on the set of potential locations D and $D' \subseteq D$, then $z(D') = \sum_{d \in D'} z_d$.

Remark 2.1. We assume that all opened facilities will be used, in the sense that there will be at least one nonempty route at each open facility. Note that, except for the LARPs with facility setup costs (MC-LARP and MC-LARP-UD), it is necessary to explicitly impose this condition because, otherwise, alternative optimal solutions could exist, where some facility is open but never used. As we will see below, this basic requirement also justifies the hypothesis that *at most* p facilities be used, instead of the usual condition that *exactly* p facilities be opened. Intuitively, one could think that, when only routing costs are considered, opening more facilities would necessarily lead to solutions with smaller routing costs, because required edges could be served from *closer* facilities. However, imposing to open (and use) exactly p facilities may lead to suboptimal routing decisions or may even force the activation of a route that does not serve any required edge and deteriorate the value of the objective function. In Fernández, Laporte, and Rodríguez-Pereira (2018), it was proven that the optimal value of an MDRPP where all depots must be used can asymptotically be twice the optimal value of the RPP on the same input graph. Indeed, this result can be extended to the MC-p-LARP, and one can find instances where, asymptotically, the optimal value of an instance with p open facilities is twice the optimal value of the same instance with just one open facility.

Also for the case of the MM-p-LARP, forcing exactly p facilities to be opened may produce undesirable solutions. A simple example is given in Figure 1, which depicts two components and three potential locations for the facilities, where the solid lines represent required edges and the dotted lines the remaining edges. As can be seen, the optimal solution for the MM-p-LARP in that instance, when exactly two facilities must be opened, will activate facilities $L1$ and $L2$ and serve from each of them the required edges in their respective components. The makespan of that solution is three. This solution has a better objective than a solution in which three facilities are opened. Indeed, when $p = 3$, facility $L3$ must also be opened and a route must be associated with it, for instance, $(L3, B, L3)$, which does not serve any required edge and gives an objective function value of four units.

Figure 1. Example of an Instance with a Better Solution for $p = 2$ than for $p = 3$



Hence, we avoid potential awkward situations, like the one in the above example, by assuming that p represents the maximum number of facilities that can be opened, so the models that we study also dictate the optimal decision in terms of the number of facilities to open.

2.1. Optimality Conditions

All the formulations that we propose use only binary variables. This follows from various optimality conditions that were established for uncapacitated arc routing problems on undirected graphs when nonnegative costs satisfy the triangle inequality (Christofides et al. 1981, Corberán and Sanchis 1994, Ghiani and Laporte 2000) and that were later extended to multidepot problems (Fernández and Rodríguez-Pereira 2017; Fernández, Laporte, and Rodríguez-Pereira 2018). These conditions apply to the maximum number of times that edges are traversed in each *individual route* in an optimal solution, and they obviously apply to LARPs:

O1 (Valid for MC-p-LARP, MC-LARP, and MM-p-LARP). There exists an optimal solution in which each required edge is served by exactly one route.

O2 (Valid for MC-p-LARP, MC-LARP, and MM-p-LARP). There exists an optimal solution in which no edge is traversed more than twice in each route.

O3 (Valid for MC-p-LARP, MC-LARP, and MM-p-LARP). There exists an optimal solution where no nonrequired edge with the two end-nodes in the same component ($e \in \gamma(V_k) \setminus R$) is traversed more than once in each route. Furthermore, because of the triangle inequality, the only edges of $\gamma(V_k) \setminus R$ that are used are those connecting two R-odd vertices.

O4 (Valid for MC-p-LARP and MC-LARP). There exists an optimal solution in which the only nonrequired edges that are traversed twice in the same route are edges of the T_C. As shown in Fernández and Rodríguez-Pereira (2017), this condition does not hold when the objective is to minimize the makespan, even when the set of depots for the routes is given. Thus, the adaptation of this condition to models with min–max objectives must take into account the fact that any least cost edge connecting any pair of components can be traversed twice in an optimal solution.

Optimality conditions O1–O4 refer to the edges that may appear in optimal solutions and to their numbers of traversals. The optimality condition O5, which we introduce below, is based on the number of facilities that can be opened in optimal solutions to the MC-p-LARP and the MC-LARP.

O5 (Valid for MC-p-LARP, MC-LARP). There exists an optimal solution in which every connected component of the graph induced by the edges that are used contains exactly one open facility.

Property O5 is obviously true for the MC-LARP. If some component of the graph induced by the edges used in an optimal solution contained more than one open facility, closing one of them would produce a solution with a better objective function value. In the case the MC-p-LARP, a similar process will produce an alternative optimal solution.

When dealing with arc routing problems with multiple depots, the counterpart of condition O2 applies to the number of edge traversals in *individual routes*, but not to the *total* number of edge traversals in optimal solutions. In particular, unless the underlying graph is a complete graph, it is possible to construct examples where an optimal MD-RPP solution traverses an edge up to $2|D|$ times, where $|D|$ is the number of depots (which is fixed; Fernández, Laporte, and Rodríguez-Pereira 2018). Unfortunately, completing the input graph becomes impractical, except for small-sized graphs, because of the increase in the number of edges (and thus of variables) that it may require.

In contrast, when dealing with min-cost LARPs (with or without setup costs), the fact that the number of operational depots is not known in advance allows us to prove that there exist optimal solutions in which no edge is traversed more than twice, provided that nonnegative costs satisfy the triangle inequality, independently of whether the graph is complete. This is a very useful property that we will exploit in some of the formulations that we propose.

Proposition 2.2.

a. *There exists an optimal MC-p-LARP solution in which no edge is traversed more than twice.*

b. *There exists an optimal MC-LARP solution in which no edge is traversed more than twice.*

Proof. First, we note that, because capacity constraints are not present, we can assume that only one route is carried out from each depot.

a. Consider an optimal solution to a given MC-p-LARP in which an edge $e \in E$ is traversed by two routes T_1 and T_2, operating from two different open facilities, d_1, d_2. The solution obtained by merging T_1 and T_2 into a single route, T, and arbitrarily closing one of the depots (for instance, d_2) is feasible for the MC-p-LARP because the parity of the vertices does not change, and the connectivity of the merged route with the remaining depot is guaranteed. Moreover, the merged solution is also optimal, because its routing cost has not changed. Edge e is traversed exactly twice in the merged route T, because otherwise two traversals of e could be removed, contradicting the optimality of the solution. This process can be repeated until all the routes traversing the same edge have been merged.

b. For the MC-LARP, we proceed as above, but now closing at each step the facility with the largest setup cost. Moreover, the merged solution will have the same routing cost and smaller setup costs. □

3. Mathematical Programming Formulations

We now present linear integer formulations for the LARPs we have defined. The main difficulty in the formulation of LARPs is to ensure that the routes are well defined (connected and closed, and preserving the parity of the vertices) and that each route starts and ends at the same facility without traversing any other facility.

A natural modeling option is to link routes with open facilities. This leads to formulations with *three-index variables*, associated with the edges traversed in the routes of the open facilities. Despite the large number of variables that such formulations entail, they can be very useful, because they allow us to easily recreate the routes from each facility once the values of the decision variables are known. Moreover, such a representation is necessary in some cases, like, for instance, when the objective function depends on the cost of some specific route (makespan) or when capacity constraints are present. The three-index variable formulation presented in Section 3.1 can thus be adapted to all six LARPs defined in Section 2.

An alternative modeling option is to work with formulations that *aggregate* the information of all the routes. This leads to the use of *two-index variables* associated with the edges traversed by the routes, but does not explicitly define the routes themselves. This approach reduces the number of required variables at the expense of requiring a final postprocessing phase to specify the route associated with each open facility. Furthermore, to guarantee the consistency of the routes produced with these variables, specific constraints are needed to impose that no route traverses more than one facility. Finally, such models are only valid for problems in which the objective is an aggregate measure of all routes (MC-p-LARP and MC-LARP) and the feasibility of the solutions can be derived from the aggregated information. Therefore, they are not valid if the objective is to minimize the makespan, which reflects the cost of one specific route, or for problems with capacity constraints, where the arcs traversed by each of the routes need to be known.

A formulation with two-index variables valid for the MC-p-LARP and MC-LARP will be presented in Section 3.2.

All the formulations that we propose exploit the optimality conditions presented in Section 2.1 and use binary variables only. In particular, we apply conditions O3 and O4 to identify the set of edges E^y that can be traversed twice in an optimal solution. Recall that for MC-p-LARP and MC-LARP, E^y contains all the required edges plus the edges of T_C, whereas for the remaining models, E^y contains all the required edges plus all edges connecting two distinct components.

3.1. Three-Index Variable Formulations

For each $e \in E$, let x_e^d be a binary variable indicating whether edge e is traversed by route from depot d. For each $e \in E^y$, let y_e^d be a binary variable taking the value one if and only if edge e is traversed twice in the solution by route from facility d. For each $d \in D$, let z_d be a binary variable designating whether facility d is opened.

3.1.1. MC-p-LARP. The mixed integer linear program for the MC-p-LARP is as follows:

$$\text{minimize} \sum_{d\in D}\sum_{e\in E} c_e x_e^d + \sum_{d\in D}\sum_{e\in E^y} c_e y_e^d \tag{1}$$

subject to

$$(x^d + y^d)(\delta(d)) \geq 2z_d \qquad d \in D \setminus V_R, \tag{2}$$

$$(x^d + y^d)(\delta(S)) \geq 2x_e^d \qquad d \in D, S \subseteq V \setminus \{d\}, \; e \in E(S), \tag{3}$$

$$(x^d - y^d)(\delta(S) \setminus H) + y^d(H) \geq x^d(H) - |H| + 1 \qquad S \subset V, H \subseteq \delta(S), \; |H| \text{ odd}, d \in D, \tag{4}$$

$$\sum_{d\in D} x_e^d \geq 1 \qquad e \in R, \tag{5}$$

$$y_e^d \leq x_e^d \qquad e \in E^y, d \in D, \tag{6}$$

$$x_e^d \leq z_d \qquad e \in E, d \in D, \tag{7}$$

$$y_e^d \leq z_d \qquad e \in E^y, d \in D, \tag{8}$$

$$z(D) \leq p, \tag{9}$$

$$x_e^d \in \{0,1\} \qquad e \in E, d \in D, \tag{10}$$

$$y_e^d \in \{0,1\} \qquad e \in E^y, d \in D, \tag{11}$$

$$z_d \in \{0,1\} \qquad d \in D. \tag{12}$$

Observe that the compact notation introduced in Section 2 is used in constraints (2), (3), (4), and (9). Inequalities (2) ensure that if a potential location is opened, then there are at least two edges incident to it. Inequalities (3) are an adaptation of the well-known connectivity constraints and ensure the connectivity of each route to its depot. This is guaranteed by imposing that if edge e is traversed by the route associated with facility $d \in D$, then the cut set of any vertex set containing the two end-nodes of e but not containing d must be crossed by at least two edges of that route. Inequalities (4) were proposed in Fernández and Rodríguez-Pereira (2017) for the MDRPP and ensure the parity (even degree) of every subset of vertices. They state that if the route associated with a given facility $d \in D$ uses a set H consisting of an odd number of edges incident to a set of vertices S, then it must use at least one additional traversal of some edge in the cut set $\delta(S)$. We further exploit the precedence relationship of the x variables with respect to the y variables. Therefore, the additional edge will either be a second traversal of some edge of H or a first traversal of some edge of $\delta(S)\setminus H$. Constraints (5) impose that all required edges be served, and (6) that no edge is traversed for the a second time unless it also has been traversed for a first time. By (7) and (8), no edge is traversed by the route of a facility that has not been opened. Inequality (9) means that at most p facilities are opened. The domains of the variables x, y, and z are defined in constraints (10)–(12).

The are $|E||D|$ x variables, $|E^y||D|$ y variables, and $|D|$ z variables. Moreover, there are $|D \setminus V_R|$ inequalities of type (2), $|R|$ inequalities of type (5), $|E^y||D|$ inequalities of type (6), and $(|E| + |E^y|)|D|$ inequalities of types (7) and (8). The sizes of the families of inequalities (3) and (4) are exponential in $|V|$.

3.1.2. MC-LARP. The formulation MC-LARP can be obtained from (2)–(12), by removing constraint (9), which limits the number of facilities to open, and adding the facility setup costs to the objective function, resulting in

$$\min \sum_{d\in D} f_d z_d + \sum_{d\in D}\sum_{e\in E} c_e x_e^d + \sum_{d\in D}\sum_{e\in E^y} c_e y_e^d. \tag{13}$$

3.1.3. MM-p-LARP. To minimize the makespan, it is necessary to define a new variable w that represents the length of the longest route. Hence, the objective function becomes the minimization of w, subject to (2)–(12). Furthermore, a new family of constraints is needed, which relates the new variable w to the route lengths. These inequalities also ensure that w represents the longest route:

$$w \geq \sum_{e\in E} c_e x_e^d + \sum_{e\in E^y} c_e y_e^d \qquad d \in D. \tag{14}$$

3.1.4. MC-p-LARP-UD, MC-LARP-UD, and MM-p-LARP-UD. Dealing with the unit customer demands and the maximum number of customers to serve from each potential location b_d only requires adding to the corresponding uncapacitated formulation the following family of capacity constraints, one for each facility:

$$\sum_{e \in R} x_e^d \leq b_d z_d \qquad d \in D. \tag{15}$$

3.1.5. Valid Inequalities. We next introduce some families of valid inequalities that can be used to reinforce the formulations presented above:

a. Because the vertices incident to required edges must be visited, for singletons $S = \{i\}$ with $i \in V_R$, the connectivity constraints (3) can be replaced with the tighter constraints

$$\sum_{d \in D} (x^d + y^d)(\delta(i)) \geq 2. \tag{16}$$

b. The connectivity constraints (3) associated with components containing no potential facility can also be replaced with a tighter set of constraints. In particular, for all $k \in K$ such that $V_k \cap D = \emptyset$, we have

$$\sum_{d \in D} (x^d + y^d)(\delta(V_k)) \geq 2. \tag{17}$$

c. In principle, only constraints (4) associated with singletons $S = \{v\}$ with $v \in V$ are needed to guarantee the parity of vertices in solutions. However, they are also valid for general vertex sets $S \subseteq V$, and imposing them for the general case leads to a formulation with a tighter linear programming (LP) relaxation. In fact, these inequalities can be further reinforced, as we show in the following proposition.

Proposition 3.1. *The inequality* (4) *associated with a given* $d \in D$, $S \subset V$, *and* $H \subseteq \delta(S)$, *with* $|H| \geq 3$ *and odd, is dominated by the valid inequality*

$$(x^d - y^d)(\delta(S) \setminus H) + y^d(H) \geq x^d(H) - |H| + 2 - z_d. \tag{18}$$

Proof. Let $d \in D$, $S \subset V$, and $H \subseteq \delta(S)$, with $|H| \geq 3$ and odd. To see that (18) is valid, recall that $z_d \in \{0, 1\}$ in any feasible solution. If $z_d = 0$, then $x_e^d = y_e^d = 0$, for all $e \in E$, so (18) reduces to $0 \geq -|H| + 2$, which holds by hypothesis. When $z_d = 1$, then (18) becomes (4). Indeed, (18) is tighter than (4) because $2 - z_d \geq 1$. □

Because the only inequalities (4) that are not dominated by the set (18) are those associated with odd edge sets $H \subset \delta(S)$ with $|H| < 3$, in the following we substitute the complete set of inequalities (4) by only its small family corresponding to singletons $S = \{v\}$ with $v \in V$, and subsets $H \subset \delta(S)$ consisting of just one edge, that is, $|H| = 1$, plus the complete set of *reinforced parity constraints* (18).

3.1.6. Reinforcing MC-*p*-LARP and MC-LARP with Condition O5. The optimality condition on the location variables, O5, can be used to reinforce the three-index formulations for MC-p-LARP and MC-LARP. Modeling O5 requires adding the following set of constraints:

$$z(V_k \cap D) \leq 1 \qquad k \in K, \tag{19}$$

$$x_e^d = z_d \qquad e \in R_k, d \in D \cap V_k, k \in K, \tag{20}$$

$$\sum_{d \in D \setminus V_k} x_e^d + z(V_k \cap D) \leq 1 \qquad e \in E_k \cup \delta(V_k), k \in K, \tag{21}$$

$$x_e^d \leq x_{e'}^d \qquad e \in (E_k \setminus R_k) \cup \delta(V_k), e' \in R_k, k \in K, d \in D \setminus V_k. \tag{22}$$

By (19), at most one facility per component will be opened. Moreover, (20) ensures that if a facility is opened in a component, then all the required edges in that component will be served from that facility. In its turn, (21) prevents any edge in the cut set of a component where a facility is opened to be traversed from any facility located at any other component. The correct *propagation* of the route associated with an open facility is guaranteed by (22) together with the original set of constraints (7). Note that the constraints (22) corresponding to required edges are not needed, because they are already implied by (20). In addition, the following sets of inequalities can be used to reinforce the resulting formulation:

$$\sum_{d \in D \setminus S} \left(x^d + y^d\right)(\delta(S)) \geq 2(1 - z(D \cap S)) \qquad S = \cup_{k \in K'} V_k, K' \subset K. \tag{23}$$

The reinforced connectivity constraints (23) impose that if no open facility belongs to the group of components defining S, then the cut set of S must contain at least two edges of some route associated with a depot that does not belong to S.

3.1.7. Polyhedral Analysis. In this section, we study some properties of the polyhedron associated with the three-index formulation. In the following, the convex hull of vectors (x, y, z) with components in $[0, 1]$ that satisfy (2)–(9) is denoted by $P_{(MC-LARP)}$. The proofs of the propositions are presented in the online appendix.

Proposition 3.2. *$dim(P_{(MC-LARP)}) = |E||D| + |E^y||D| + |D| - |R|$ if and only if every cut-edge set $\delta(S)$, $S \subset V \setminus D$, contains at least three edges, and every cut-edge set $\delta(S)$ such that $S = \bigcup_{i \in K'} V_i \setminus D$, $\emptyset \neq K' \subset K$, contains at least four edges.*

Proposition 3.3. *The inequality $x_e^d \geq 0$, $e \in E$, $d \in D$, defines a facet of $P_{(MC-LARP)}$ if and only if every cut set $\delta(S)$, $S \subset V \setminus D$, containing e contains at least four edges, and every $\delta(S)$ such that $S = \bigcup_{i \in K'} V_i \setminus D$ $(\emptyset \neq K' \subset K)$ contains at least five edges.*

Proposition 3.4. *The inequality $x_e^d \leq 1$, $e \in E$, $d \in D$, induces a facet of $P_{(MC-LARP)}$ if and only if every cut set $\delta(S)$ containing e contains at least four edges.*

Proposition 3.5. *The connectivity inequality* (3) *associated with $S = \bigcup_{i \in K'} V_i$ ($\emptyset \neq K' \subset K$), $S \bigcap D = \emptyset$, $e \in E(S)$, induces a facet of $P_{(MC-LARP)}$ if and only if the graphs induced by the connected components $G(S)$ and $G(V \backslash S)$ satisfy the following:* (i) *$G(S)$ is connected, and each connected component of $G(V \backslash S)$ contains at least one open facility;* (ii) *for every subset of components in $S' \subset S$ (or S' in $V \backslash S$) with $S' \bigcap D = \emptyset$, the inequality $|\delta(S') \backslash \delta(S)| \geq 2$ holds.*

Proposition 3.6. *The reinforced parity constraints* (18) *induce facets of $P_{(MC-LARP)}$ for S and H such that $|\delta(S)| \geq |H| + 1$ and $H \cap \delta(D) = \emptyset$.*

3.2. Two-Index Variable Formulations

Here we propose a new formulation for MC-p-LARP and MC-LARP in which the routes are represented by two-index variables, associated solely with edge traversals and not with the facilities they are linked to. The formulation exploits Proposition 2.2. Regardless of whether G is a complete graph, there exists an optimal solution to both MC-p-LARP and MC-LARP in which no edge is traversed more than twice. Therefore, in both cases, the total number of traversals of each edge can be represented by only two binary variables, one for the first one and one for the second one. Unlike the three-index formulations above, these variables now represent aggregated information over all the routes. In such a formulation, connectivity and parity conditions are no longer sufficient to guarantee that the routes start and end at the same facility. Hence, additional constraints are required to guarantee the consistency of routes. To this end, we introduce an extension of a set of constraints proposed in Fernández, Laporte, and Rodríguez-Pereira (2018) for the MDRPP, which now integrate locational decision variables as well.

We use the same location variables as above so the binary variable z_d, $d \in D$, indicates whether a facility is established at d. As for the routing, let x_e denote the binary variable for the first traversal of edge $e \in E$, and let y_e the binary variable indicating whether edge $e \in E^y$ is traversed a second time.

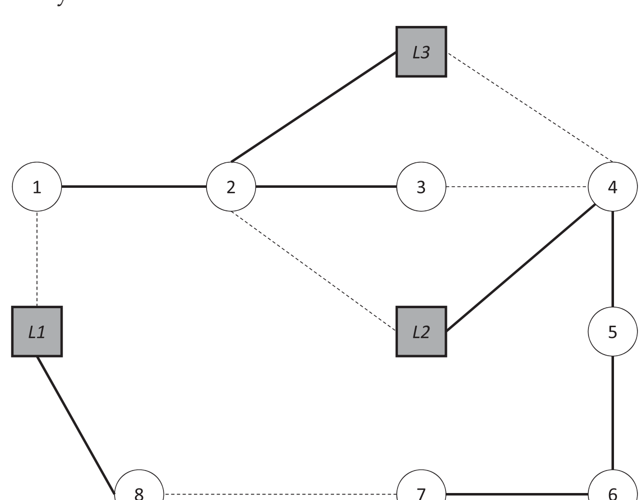
3.2.1. Return-to-Facility Constraints.

Before presenting the formulation, we discuss the return-to-facility constraints (RtFCs), which guarantee that all routes start and end at the same facility. These are a variation of the parity inequalities that stated that the set of open facilities involved in the edges of the cut sets is needed to guarantee consistent routes in LARPs. In fact, the RtFCs extend the family of inequalities introduced in Fernández, Laporte, and Rodríguez-Pereira (2018) for the MDRPP, which ensure that routes return to the appropriate depot. These inequalities are no longer valid for LARPs because they assume that the set of depots from which routes originate is known. However, because the set of potential locations that will actually become depots for the routes is not known in advance for LARPs, location variables are required in the proposed inequalities. As we will see, the resulting inequalities are quite involved.

In Figure 2, the gray squares represent potential facilities, and the solid lines correspond to required edges. This figure illustrates not only that connectivity and parity constraints are not sufficient to guarantee well-defined routes, but also that the conditions needed to guarantee consistent routes in LARPs necessarily depend on the set of open facilities. Observe that if only one or two of the three potential locations were opened, the displayed solution would be feasible and, depending on the case, it would consist of one or two well-defined routes. Instead, if all three potential facilities opened, the displayed solution would be infeasible because it would not be possible to decompose it into three routes, each starting and ending at the same facility. Moreover, if all three potential facilities opened, any feasible solution should have at least three more edges (or additional traversals of the existing edges) in the cut set of $S = \{1, 2\}$. This idea is formalized below.

Consider a vertex set $S \subset V \setminus D$ and a subset of potential facilities $D' = \{d_1, \ldots, d_r\} \subset D$. Consider also a subset of edges $H \subset \delta(S) \cap \delta(D')$. Denote by $H_i \neq \emptyset$ the set of edges of H incident with facility $i \in D'$ and assume that each H_i contains an odd number of edges. Finally, partition $\delta(S) \setminus H$ in the following two sets: $F_{S,H}^{D'} = (\delta(S) \setminus H) \setminus \delta(D \setminus D')$, the set of edges of $\delta(S) \setminus H$ that are not incident to any potential

Figure 2. Infeasible Solution Satisfying Connectivity and Parity Constraints



facility different from those of D', and $Q_{S,H}^{D'} = (\delta(S) \setminus H) \cap \delta(D \setminus D')$, the set of edges of $\delta(S) \setminus H$ incident to some potential facility not in D'. The inequality that we propose contains bilinear terms that will be discussed and linearized later on.

Proposition 3.7. *The RtFC*

$$(x-y)\left(F_{S,H}^{D'}\right) + \sum_{d \in D \setminus D'} (1-z_d)(x-y)\left(Q_{S,H}^{D'} \cap \delta(d)\right) + y(H) \geq x(H) - |H| + z(D') \tag{24}$$

associated with S, D', and H as defined above is valid for the MC-p-LARP and MC-LARP.

Proof. Let (z, x, y) be a feasible LARP solution, and note that the RtFC (24) is active only if $x(H) - |H| + z(D') > 0$. Because $x(H) - |H| \leq 0$, a necessary condition is that $z(D') \geq 1$. Consider the following cases:

(a) $x(H) = |H|$ and $z(D') \geq 1$. The right-hand side of the RtFC reduces to $z(D') \geq 1$. Because $x(H) = |H|$, $x_e = 1$, for all $e \in H_i$, $i \in \{1, \ldots, r\}$. Given that all the edges in each H_i are incident with the same potential facility and $|H_i|$ is odd, there must be at least one additional traversal of some edge in the cut set associated with each open facility of the set D'; that is, in total, $z(D')$ additional traversals are needed, which must correspond either to second traversals of edges in H (term $y(H)$) or to first traversals of edges in $\delta(S) \setminus H$. In the latter case, the first traversal may correspond to edges not incident with potential locations of $D \setminus D'$, represented by the first term of the left-hand side $(x-y)(F_{S,H}^{D'})$, or to potential locations of $D \setminus D'$, provided that the involved potential locations are not open, represented by the second term of the left-hand side $\sum_{d \in D \setminus D'} (1-z_d)(x-y)(Q_{S,H}^{D'} \cap \delta(d))$. The bilinear terms are necessary because the edges incident with potential locations in $D \setminus D'$ may contribute to the overall count only when the potential facility involved remains closed.

(b) $x(H) = |H| - 1$ and $z(D') \geq 2$. The right-hand side of the RtFC reduces to $z(D') - 1 \geq 1$. In this case, exactly one of the edges of H is not traversed in solution (z, x, y). In full, let us assume that $x(H_1) = |H_1| - 1$ (which is even), and $x(H_i) = |H_i|$, $i \in \{2, \ldots, r\}$. Consider now $\overline{D'} = D' \setminus \{d_1\}$ and $\overline{H} = H \setminus H_1$:

1. The RtFC associated with S, $\overline{D'}$, and $\overline{H}$ corresponds to Case a, because $x(H \setminus H_1) = (|H \setminus H_1|)$ and $z(\overline{D'}) = z(D' \setminus \{d_1\}) \geq 1$. Therefore, it is valid.

2. The RtFC associated with S, D', and H is dominated by the RtFC associated with S, $\overline{D'}$, and $\overline{H}$. Both inequalities have the same right-hand side, and the left-hand side of the former is weaker than the right-hand side of the latter because $y(H) \geq y(\overline{H})$ and $(x-y)\left(F_{S,H}^{D'}\right) \geq (x-y)\left(F_{S,\overline{H}}^{\overline{D'}}\right) + (1-z_{d_1})(x-y) \cdot ((\delta(S) \setminus H) \cap \delta(d_1))$.

Hence, the RtFC associated with S, D', and H is valid.

(c) $x(H) = |H| - 2$ and $z(D') \geq 3$. The right-hand side of the RtFC is $z(D') - 2 \geq 1$. There are exactly two edges of H, say, e_1, e_2, that are not traversed in the solution (z, x, y). Consider the two possible subcases:

1. $e_1, e_2 \in H_1$. Then, quite similarly to Case b, the RtFC associated with S, D', and H is dominated by that associated with S, D', and $\overline{H} = \{\overline{H_1}, H_2, \ldots, H_r\}$, with $\overline{H_1} = H_1 \setminus \{e_1, e_2\}$, which corresponds to Case a.

2. e_1 and e_2 are incident with two different depots; that is, $e_1 \in H_1$ and $e_2 \in H_2$. Then, the RtFC associated with S, D', and H is dominated by that associated with S, $\overline{H} = H \setminus \{H_1, H_2\}$, and $\overline{D}' = D' \setminus \{d_1, d_2\}$, which also corresponds to Case a.

Hence, the RtFC associated with S, D', and H is valid.

(d) All other cases can be handled similarly. □

To illustrate, consider again the solution depicted in Figure 2 with two alternative values for the location variables: one where all three potential facilities are open, that is, $z_{L1}^1 = z_{L2}^1 = z_{L3}^1 = 1$, which, as explained, is infeasible, and another one where only $L1$ and $L2$ are open, that is, $z_{L1}^2 = z_{L2}^2 = 1, z_{L3}^2 = 0$, which is feasible. Consider the vertex set $S = \{1, 2\}$, $H_1 = \{(1, L1)\}$, and $H_2 = \{(2, L2)\}$. In both cases, let $D' = \{L1, L2\}$, so $F_{S,H}^{D'} = \{(2, 3)\}$ and $Q_{S,H}^{D'} = \{(2, L3)\}$.

For the infeasible solution z^1, we have $z^1(D') = z_{L1}^1 + z_{L2}^1 = 2$. Because $z_{L3}^1 = 1$, we also have $h_{(2,L3)}^{d_3} = 0$, so $\sum_{d \in D \setminus D'} \overline{h}^d(Q_{S,H}^{D'} \cap \delta(d)) = 0$. Therefore, the associated RtDC (32) is violated because $x(H) - |H| + z(D') = 2$, but $(x-y)(F_{S,H}^{D'}) + \sum_{d \in D \setminus D'} h^d(Q_{S,H}^{D'} \cap \delta(d)) + y(H) = 1$.

If we instead consider the feasible solution z^2, we also have $z^2(D') = z_{L1}^2 + z_{L2}^2 = 2$, but now $h_{(2,L3)}^{d_3} = 1$, because $z_{L3}^2 = 0$. Hence, $\sum_{d \in D \setminus D'} \overline{h}^d(Q_{S,H}^{D'} \cap \delta(d)) = 1$, and the left-hand side of the RtDC becomes $1+1$, which coincides with the value of the right-hand side that does not change. Hence, as expected, the RtFC is not violated for this feasible solution.

To integrate the set of inequalities (24) within a mixed integer linear program (MILP), it is necessary to linearize the bilinear terms that they include. For this, we define additional decision variables representing the products $h_{ed} = (1 - z_d)(x_e - y_e)$ for the edges $e \in \delta(d)$, with $d \in D$. These variables will take the value one if and only if edge e, which is incident with potential facility d, is traversed exactly once and the facility located at d is not open. Observe that the number $\sum_{d \in D} |\delta(d)|$ of new variables is very moderate because we are assuming that $|V_k \cap D| \leq 1$, for all $k \in K$. This number is clearly smaller than the number of two-index variables. The new set of variables h and variables x, y, and z can be related with the usual *linearizing* constraints:

$$h_{ed} \le (1 - z_d) \qquad d \in D, e \in \delta(d), \quad (25)$$
$$h_{ed} \le (x_e - y_e) \qquad d \in D, e \in \delta(d), \quad (26)$$
$$(1 - z_d) + (x_e - y_e) \le 1 + h_{ed} \qquad d \in D, e \in \delta(d). \quad (27)$$

3.2.2. MILP Formulation for MC-p-LARP and MC-LARP. The mixed integer linear program for the MC-p-LARP is as follows:

$$\text{minimize} \sum_{e \in E} c_e x_e + \sum_{e \in E^y} c_e y_e \quad (28)$$

subject to

$$(x + y)(\delta(d)) \ge 2z_d \qquad d \in D, \quad (29)$$
$$(x + y)(\delta(S)) \ge 2(1 - z(S)) \qquad S \subseteq V, S \cap V_R \ne \emptyset, \quad (30)$$
$$(x - y)(\delta(S) \setminus H) + y(H) \ge x(H) - |H| + 1 \qquad S \subset V,\ H \subseteq \delta(S),\ |H| \text{ odd}, \quad (31)$$
$$(x - y)\left(F^{D'}_{S,H}\right) + \sum_{d \in D \setminus D'} h^d((\delta(S) \setminus H) \cap \delta(d)) + y(H) \ge x(H) - |H| + z(D') \qquad S \subset V \setminus D, D' = \{d_1, \ldots, d_r\} \subset D,\ H = H_1 \cup \ldots H_r,\ H_i \subseteq \delta(S) \cap \delta(d_i),\ |H_i| \text{ odd},\ i = 1, \ldots, r, r > 1, \quad (32)$$
$$x_e = 1 \qquad e \in R, \quad (33)$$
$$y_e \le x_e \qquad e \in E^y, \quad (34)$$
$$z(D) \le p, \quad (35)$$
$$h_{ed} + z_d \le 1 \qquad d \in D, e \in \delta(d), \quad (36)$$
$$h_{ed} + y_e \le x_e \qquad d \in D, e \in \delta(d), \quad (37)$$
$$x_e \le z_d + y_e + h_{ed} \qquad d \in D, e \in \delta(d), \quad (38)$$
$$x_e \in \{0, 1\} \qquad e \in E, \quad (39)$$
$$y_e \in \{0, 1\} \qquad e \in E^y, \quad (40)$$
$$z_d \in \{0, 1\} \qquad d \in D, \quad (41)$$
$$h_{ed} \in \{0, 1\} \qquad d \in D, e \in \delta(d). \quad (42)$$

Inequalities (29) ensure that open facilities are used, and the family (30) is an adaptation of the well-known connectivity inequalities: there must be at least two edge traversals in the cut set of a given set of vertices S containing no open facility whenever S contains some vertex that must be visited. Inequalities (31) have a similar explanation to that of (4) and ensure the parity (even degree) of every subset of vertices. They were used in the two-index formulation for the MDRPP proposed in Fernández, Laporte, and Rodríguez-Pereira (2018) (observe that they do not involve any location variable). The RtFCs (32) were discussed above. Equalities (33) ensure that all required edges are served, whereas constraints (34) mean that an edge cannot be traversed for a second time unless it also has been traversed for the first time. The limit on the maximum number of facilities that can be opened is imposed by (35). The linearization of the set of new variables h and its relation to the other decision variables is given in (36)–(38). Finally, the domains of the different sets of decision variables are stated in (39)–(42).

The above formulation contains $|E|$ x and $|E^y|$ y variables and $|D|$ z variables. As mentioned, the number of h variables is $\sum_{d \in D} |\delta(d)|$. There are $|D|$ inequalities of type (29), $|R|$ inequalities of type (33), and $|E^y|$ inequalities of type (34). The number of constraints in each family (36)–(38) is $\sum_{d \in D} |\delta(d)|$. The number of inequalities of types (30), (31), and (32) is exponential in $|V|$.

MC-LARP. Because the domains of MC-p-LARP and MC-LARP are the same, except for constraint (35) on the maximum number of open facilities, to adapt the above formulation to the MC-LARP, we need only to discard this constraint and to update the objective function to

$$\min \sum_{d \in D} f_d z_d + \sum_{e \in E} c_e x_e + \sum_{e \in E^y} c_e y_e. \quad (43)$$

Proposition 3.8. *Formulation* (29)–(42) *is valid for the MC-p-LARP and for the MC-LARP.*

Proof. By Proposition 3.7, inequalities (32) are valid. Therefore, if a solution $(\bar{x}, \bar{y}, \bar{z})$ is feasible for the MC-p-LARP or the MC-LARP, no violated inequality of this family exists. We now show that if a solution $(\bar{x}, \bar{y}, \bar{z})$ satisfying (29)–(31), (33)–(35), (39), and (40) is not feasible for the MC-p-LARP or the MC-LARP, then there exists a constraint (32) violated by the solution. Because of the connectivity and parity constraints (30) and (31), if $(\bar{x}, \bar{y}, \bar{z})$ is not feasible, then in any decomposition of the solution in edge-disjoint simple tours, there is one simple tour T traversing at least two open facilities. Let $d_1, d_2 \in D$ be two open facilities that are consecutive in the tour T, and let $P_{d_1 d_2}$ be the subpath of T connecting d_1 and d_2, and $S^T = V(P_{d_1 d_2}) \setminus D$.

a. If the decomposition contains no simple tour T' incident with some vertex of S^T, that is, $S^T \cap V(T') \ne \emptyset$, then the RtFC (32) associated with $S = S^T$, $D' = \{d_1, d_2\}$, $H_1 = S \cap \delta(d_1)$, $H_2 = S \cap \delta(d_2)$, $F^{D'}_{S,H} = (\delta(S) \setminus H) \setminus \delta(D \setminus D')$, and $Q^{D'}_{S,H} = (\delta(S) \setminus H) \cap \delta(D \setminus D')$ is violated by $(\bar{x}, \bar{y}, \bar{z})$, because all the terms on the left-hand side of (32) take the value zero, but the right-hand side takes the value two, because $\bar{z}_{d_1} = \bar{z}_{d_2} = 1$.

b. Suppose now that the decomposition contains a simple tour T' incident with some vertex of S. Let $\{v\} \in S \cap V(T')$ (arbitrarily selected, if there is more than one such vertex). Consider the following subcases:

– T' does not intersect with $V(T) \setminus P_{d_1d_2}$. Consider $S^{T'}$ consisting of all vertices of $V(T')$ that are not open facilities in $\bar{z}$ (possibly all $V(T')$). Then the RtFC (32) associated with $S = S^T \cup S^{T'}$, $D' = \{d_1, d_2\}$, $H_1 = S \cap \delta(d_1)$, $H_2 = S \cap \delta(d_2)$, $F^{D'}_{S,H} = (\delta(S) \setminus H) \setminus \delta(D \setminus D')$, and $Q^{D'}_{S,H} = (\delta(S) \setminus H) \cap \delta(D \setminus D')$ is violated by $(\bar{x}, \bar{y}, \bar{z})$. Again, all the terms on the left-hand side of (32) take the value zero, but the right-hand side takes the value two.

– T' intersects with $V(T) \setminus P_{d_1d_2}$. Let $\{v'\} \in V(T') \cap (V(T) \setminus P_{d_1d_2})$. If several such vertices exist, v' is the *first* vertex *after* d_2 following the same orientation as that of $P_{d_1d_2}$. Observe that now T' must traverse some open facility, say, $d' \in D \cup V(T')$, different from those of $\{d_1, d_2\}$. Otherwise, a different decomposition of the solution of simple tours would exist, where d_1 and d_2 are no longer consecutive open facilities in the same simple tour. Consider now the subpaths of T', $P_{v,v'}$, and $P_{v,d'}$, and define $S^{T'} = V(P_{v,v'}) \cup (V(P_{v,d'}) \setminus D)$. Then, the RtFC (32) associated with $S = S^T \cup S^{T'}$, $D' = \{d_1, d_2\}$, $H_1 = S \cap \delta(d_1)$, $H_2 = S \cap \delta(d_2)$, $F^{D'}_{S,H} = (\delta(S) \setminus H) \setminus \delta(D \setminus D')$, and $Q^{D'}_{S,H} = (\delta(S) \setminus H) \cap \delta(D \setminus D')$ is violated by $(\bar{x}, \bar{y}, \bar{z})$. Now the left-hand side of (32) takes the value one (corresponding to the last edge of the path $P_{v,v'}$), but the right-hand side is two. □

Remark 3.1. An additional consequence of the above proof is that RtFC inequalities (32) associated with subsets D' with two depots suffice to guarantee that the proposed formulation is valid.

Some of the valid inequalities presented in Section 3.1.1 can be adapted to reinforce the formulation above. In particular, the reinforced connectivity inequalities (16) associated with singletons that must be visited $S = \{i\}$ with $i \in V_R$ can be expressed in terms of the aggregated x and y variables as

$$(x + y)(\delta(i)) \geq 2. \tag{44}$$

Analogously, (17) can be expressed in terms of the aggregated x and y variables to reinforce constraints (30) associated with components containing no potential facility as

$$(x + y)(\delta(V_k)) \geq 2. \tag{45}$$

Finally, the logical relation between the z and x variables associated with edges connecting two facilities can be written as

$$x_e + z_d + z_f \leq 2 \qquad e = (d, f) \in \gamma(D). \tag{46}$$

Modeling optimality condition O5 for the two-index formulations is not easy. In fact, we do not know how to impose this condition without incorporating additional decision variables, and preliminary experiments clearly indicate that such an alternative would not be competitive with the original formulations.

4. Branch-and-Cut Algorithm

We have developed an exact branch-and-cut algorithm to solve each of the models presented, based on the formulations proposed in Section 3. The overall solution algorithm is similar for three- and two-index formulations. As usual, we initially relax the families of constraints of exponential size. After each LP iteration, these are then separated to detect whether there are constraints of any of these families violated by the current LP solution. If so, the detected violated constraints are incorporated in the current formulation, and the reinforced formulation is solved.

The algorithm starts with all integrality conditions relaxed and only a subset of constraints. In the initial formulations, we include all nonexponential sets of constraints, plus a small subset of connectivity and parity inequalities. More precisely, the initial connectivity constraints considered are associated with the singletons that must be visited, that is, $S = \{i\}$, $i \in V_R$, and with the components that contain no potential facility, that is, $S = V_k$, $k \in K$, with $V_k \cap D = \emptyset$. The initial set of parity constraints is restricted to those associated with R-odd singletons; that is, for the three-index formulations, constraints (3) are initially replaced with (16) and (17), and the only parity constraints initially included are the inequalities (4) associated with R-odd singletons $S = \{v\}$, with $|\delta_R(v)|$ odd. For the two-index formulations, constraints (30)–(32) are initially replaced with (44) and (45), the only parity constraints (31) initially included are those associated with R-odd singletons $S = \{v\}$ with $|\delta_R(v)|$ odd, and all logical inequalities (46) are added.

RtFCs (32) are handled as lazy constraints, so they are only separated at the nodes with an integer LP solution. In contrast, all other families of relaxed inequalities are separated whenever the current LP solution is fractional. We then first apply a heuristic separation and resort to the exact separation only when the heuristic fails in finding any violated cut. Below we detail the separation procedures that are applied in each case.

4.1. Separation of Inequalities for the Three-Index Formulations

Let $(\bar{x}, \bar{y}, \bar{z})$ denote the current LP solution, and let $G(\bar{x}, \bar{y})$ be the support graph associated with $(\bar{x}, \bar{y})$ at any iteration of the algorithm. For each facility $d \in D$, we denote by $(\bar{x}^d, \bar{y}^d)$ the partial LP solution associated with the potential facility d, and by $G^d_{\bar{x},\bar{y}} = (V^d, E_{\bar{x}^d,\bar{y}^d})$ its corresponding support graph, which can be obtained from G by eliminating all edges in E with $\bar{x}^d_e = 0$

and all vertices that are not incident with any edge of $E_{\bar{x}^d,\bar{y}^d}$.

4.1.1. Separation of the Connectivity Constraints (3). For each potential facility $d \in D$, we check whether $G^d_{\bar{x},\bar{y}}$ is connected. If not, each connected component C with vertex set $V(C) \subseteq V^d \setminus \{d\}$ defines a violated connectivity constraint (3). When the current LP solution is integer, then $\bar{z}_d = 1$, and the above separation procedure is exact. However, when the current LP solution is fractional, it may fail to find a violated constraint (3) even if one exists. Therefore, when $G^d_{\bar{x},\bar{y}}$ contains one single connected component, we search for connected components in the subgraph of $G^d_{\bar{x},\bar{y}}$ that contains only those edges with values $\bar{x}^d_e + \bar{y}^d_e \geq \varepsilon$, where ε is a given parameter. We then compute the current value of $(\bar{x}^d_e + \bar{y}^d_e)(V(C))$ for each connected component C with vertex set $V(C) \subseteq V^d \setminus \{d\}$. If for some edge $e \in \gamma(V(C))$ the inequality $(\bar{x}^d + \bar{y}^d)(\delta(V(C))) < 2x^d_e$ is satisfied, then the connectivity inequality (3) associated with $V(C)$ is violated by $(\bar{x}^d, \bar{y}^d)$. Finally, if no violated constraint has been found with the above heuristic, we build the tree of min cuts T^d of $G^d_{\bar{x},\bar{y}}$ with capacities given by $\bar{x}^d_e + \bar{y}^d_e$. For each edge $e = (u,v)$ in $E_{\bar{x}^d,\bar{y}^d}$ with $u,v \neq d$, the minimum cut $\delta(S)$ such that $e \in \gamma(S)$ is easily obtained from the min-cut tree T^d. If the value of the min cut is smaller than $2\bar{x}^d_e$, then the inequality (3) associated with S and d is violated by $(\bar{x}^d, \bar{y}^d)$. The above separation procedure is exact and similar to that applied in Fernández and Rodríguez-Pereira (2017) to the connectivity constraints of the three-index formulation for the MDRPP. The complexity of this separation procedure is dominated by that of solving the max-flow problems, which allow determining the min cuts. Thus, the overall complexity is $O(n^4)$.

4.1.2. Separation of the Parity Inequalities (18). Because the initial formulation includes all parity constraints (4) associated with singletons, for integer solutions $(\bar{x}, \bar{y}, \bar{z})$, the reinforced parity inequalities (18) are always satisfied. When $(\bar{x}, \bar{y}, \bar{z})$ is not integer, we first apply a heuristic, and we resort to the exact separation only if the heuristic fails. The heuristic and exact method for inequalities (18) are adaptations of those applied in Fernández and Rodríguez-Pereira (2017) to the simple parity constraints (4) of the three-index formulation for the MDRPP, where now the right-hand side of the inequality is $2 - z_d$, instead of 1.

Concerning the heuristic for each potential facility $d \in D$, we find the connected components of the subgraph $G^d(\bar{x}, \bar{y})$ induced by edges with values $b^d_e = \min\{(\bar{x}^d_e - \bar{y}^d_e), 1 - (\bar{x}^d_e - \bar{y}^d_e)\} > \varepsilon$, where ε is a given parameter. Then, if $S \subset V$ is the vertex set of one of the components, its associated edge set is $H = \{e \in \delta(S) \mid 1 - (\bar{x}^d_e - \bar{y}^d_e) < \bar{x}^d_e - \bar{y}^d_e\}$. If $b^d(\delta(S)) < 2 - \bar{z}_d$ and $|H|$ is odd, then the parity constraint (18) associated with S and H is violated by $(\bar{x}^d, \bar{y}^d, \bar{z}^d)$. If $|H|$ is even, we obtain an odd set $|H|$ by either removing one edge from $|H|$ [and transferring it to $\delta(S) \setminus H$] or by adding to H one edge currently in $\delta(S) \setminus H$. In particular, the smallest increment is obtained with

$$\Delta = \min\Big\{\min\{\bar{x}^d_e - \bar{y}^d_e : e \in \delta(S) \setminus H\},\ \min\{1 - (\bar{x}^d_e - \bar{y}^d_e) : e \in H\}\Big\}.$$

Then, if $b^d(\delta(S)) + \Delta < 2 - \bar{z}_d$, the parity constraint (18) associated with S and the updated set H is violated by $(\bar{x}^d, \bar{y}^d)$. Otherwise, the heuristic fails to find a constraint violation.

The exact method constructs, for each $d \in D$, the tree of min cuts T^d of the support graph G^d with capacities b^d. When T^d has a cut $\delta(S)$ of capacity smaller than $2 - \bar{z}_d$, that is, $b(\delta(S)) < 2 - \bar{z}_d$, we consider its vertex set S and the set of edges $H = \{e \in \delta(S) \mid (\bar{x}^d_e - \bar{y}^d_e) \geq 0.5\}$. If $|H|$ is odd, then H defines, together with S, a violated inequality of type (18). Otherwise, if $|H|$ is even, we update the set H to an odd set by moving an edge as mentioned above. When $b^d(\delta(S)) + \Delta < 2 - \bar{z}_d$, the updated set H defines a violated inequality (18) for d and S for the current solution $(\bar{x}^d, \bar{y}^d)$. The complexity of this separation procedure is the same as that of the connectivity constraints: $O(n^4)$.

4.2. Separation of Inequalities for the Two-Index Formulations

Let $G(\bar{x}, \bar{y})$ denote the support graph associated with the LP solution $(\bar{x}, \bar{y}, \bar{z})$ at any iteration of the algorithm.

4.2.1. Separation of the Connectivity Inequalities (30). The separation of constraints (30) is an adaptation of the procedure presented in Fernández, Laporte, and Rodríguez-Pereira (2018) for the connectivity constraints of the two-index formulation for the MDRPP. Now we need to take into account that the right-hand side is $2(1 - z(S))$ instead of 2. We first check whether $G(\bar{x}, \bar{y})$ is connected. If not, the vertex set of any component containing no depot defines a violated cut. As before, when $(\bar{x}, \bar{y}, \bar{z})$ is integer, the above separation is exact, but it may fail for fractional solutions. In such a case, the connected components are identified in the subgraph of $G(\bar{x}, \bar{y})$ with only those edges with $\bar{x}_e + \bar{y}_e \geq \varepsilon$, where ε is a given parameter. Then, the value $(\bar{x} + \bar{y})(\delta(V(C)))$ is computed for each component $V(C)$ and compared with $2(1 - \bar{z}(S))$. If $(\bar{x} + \bar{y})(\delta(V(C))) < 2(1 - z(S))$, the constraint (30) associated with $V(C)$ is violated by $(\bar{x}, \bar{y}, \bar{z})$.

For the exact separation, we build the tree of min cuts of $G(\bar{x}, \bar{y})$ with capacities given by $\bar{x}_e + \bar{y}_e$, and look for min cuts $\delta(S)$ of value $(\bar{x}, \bar{y})(\delta(S)) < 2$. When

$(\overline{x},\overline{y})(\delta(S)) < 2(1-z(S))$, the inequality (30) associated with S is violated by $(\overline{x},\overline{y},\overline{z})$.

4.2.2. Separation of the Parity Inequalities (31). We use the separation of constraints (31) presented in Fernández, Laporte, and Rodríguez-Pereira (2018) for the two-index formulation for the MDRPP. Because the initial formulation includes the inequalities associated with singletons, we separate them only in fractional solutions. We first apply a heuristic that finds the connected components in the subgraph $G(\overline{x},\overline{y},\overline{z})$ induced by edges with values $b_e = \min\{(\overline{x}_e - \overline{y}_e), 1-(\overline{x}_e-\overline{y}_e)\} > \varepsilon$, where ε is a given parameter. Then, if $S \subset V$ is the vertex set of one of the components, we proceed as above to identify its associated edge set H. If $b(\delta(S)) < 1$ and $|H|$ is odd, then the parity constraint (31) associated with S and H is violated by $(\overline{x},\overline{y})$. Otherwise, if $b(\delta(S)) + \Delta < 1$, the parity constraint (31) associated with S and the updated set H is violated by $(\overline{x},\overline{y})$. If $|H|$ is odd and $b(\delta(S)) + \Delta \geq 1$, then it is necessary to apply the exact method.

For the exact separation, we construct the tree of min cuts T of $G_{\overline{x},\overline{y},\overline{z}}$ with capacities given by b_e. When T^b has a cut $\delta(S)$ of capacity smaller than one, we consider its vertex set S and the set of edges $H = \{e \in \delta(S) \mid (\overline{x}_e - \overline{y}_e) \geq 0.5\}$. If $|H|$ is odd, a violated inequality is defined by H and S. When $|H|$ is even, an updated odd set H can be identified by moving an edge. Then, if $b(\delta(S)) + \Delta < 1$, the updated set H defines a violated cut (31) for S.

4.2.3. Separation of the Return-to-Facility Inequalities (32). RtFCs (32) are handled as lazy constraints, so they are separated only when the LP solution $(\overline{x},\overline{y},\overline{z})$ is integer. In such a case, violated inequalities can be easily identified by first finding a tour decomposition of the current solution (see, for instance, Hierholzer 1873) and then checking whether any of the tours contain a path $P_{d_1d_2}$ connecting two (consecutive) open facilities. If so, $D' = \{d_1, d_2\}$ and $S = V(P_{d_1d_2}) \setminus D'$ define a violated cut. The complexity of the separation procedure is dominated by that of finding the tour decomposition, which is $O(m)$.

5. Computational Experiments

In this section, we present the results of computational experiments we have conducted to assess the behavior of our formulations on the different LARPs studied.

5.1. Description of the Instances

The sets of instances used in the computational experiments were adapted from MDRPP benchmark instances used in Fernández, Laporte, and Rodríguez-Pereira (2018) and Fernández and Rodríguez-Pereira (2017), which, in turn, were adapted from the following sets of well-known RPP instances: the "ALB" set from Corberán and Sanchis (1994, 1998); the "P" set from Christofides et al. (1981); four sets of instances from each of three classes from Hertz, Laporte, and Nanchen Hugo (1999), instances with vertices of degree four (labeled "D"), grid instances (labeled "G"), and randomly generated instances (labeled "R"); and larger sets of instances ("ALB2," "GRP," "MAD," "URP5," and "URP7") from Corberán, Plana, and Sanchis (2007). We preserved from each original instance the set of required edges and the routing cost function c. The maximum number of facilities to be located has been fixed to $p = 4$. The potential locations for the facilities were randomly chosen from the set of vertices, ensuring that no component had more than one potential location. To define the potential locations for the facilities, we first considered the connected components of the input graph, where each vertex nonincident to any required edge defines one component. Then, potential locations were assigned to components according to some weights p_k, $k \in K$, defined as the sum of a fixed parameter $r = 0.2$, plus a parameter based on the required edges, defined as the ratio between the number of required edges in that component and the total number of required edges; that is $p_k = 0.2 + |R_k|/|R|$, for all $k \in K$. For the considered set of benchmark instances, the resulting values were always smaller than one. Then, for each component, a number r_k was randomly drawn from a continuous uniform distribution $U[0,1]$, and the component was allocated a potential site when $p_k \leq r_k$. In that case, the vertex of V_k where the potential location was actually located was obtained by randomly generating a number v from a discrete uniform distribution $U[1,|V_k|]$. To generate the setup costs of the potential locations, for each instance I, we took from Fernández, Laporte, and Rodríguez-Pereira (2018) the optimal values of the instance solved as an MDRPP with two and four depots, V_I^2 and V_I^4, respectively. Then, the value $V_I = |(V_I^2 - V_I^4)|/2$ was taken as the average setup cost for that instance, and the values f_d, $d \in D$ for instance I were randomly generated from a discrete uniform distribution $U[V_I/2, 3V_I/2]$. Finally, the capacity of each potential location, b_d, was randomly generated from a discrete uniform distribution $U[|R|/4, 3|R|/4]$. Note that, on average, four open facilities are sufficient to serve all the demand, which is consistent with the selected value of p.

Table 2 shows the characteristics of the instances. The column headings represent the number of instances in the set ("# of instances"), the number of vertices ("$|V|$"), the number of edges ("$|E|$"), the number of required edges ("$|R|$"), the number of connected components in the graph induced by the required edges ("$|K|$"), and the number of potential locations ("$|D|$"). In each column, when not all the instances of the group have the same value, the minimum and maximum values are given.

Table 2. Characteristics of the Instances

	No. of instances	$\|V\|$	$\|E\|$	$\|R\|$	$\|K\|$	$\|D\|$
ALB	2	90–102	143–159	88–99	10–11	5
P	24	7–50	10–183	4–78	2–8	4–6
D16	9	8–16	12–30	3–16	2–5	4–6
D36	9	25–36	52–71	10–38	4–11	4–10
D64	9	40–63	92–120	27–75	5–15	5–16
D100	9	76–100	161–197	50–121	9–22	5–17
G16	9	11–16	15–24	3–13	3–5	4–7
G36	9	22–36	34–60	11–35	5–9	4–7
G64	9	45–63	74–110	24–68	4–14	5–15
G100	9	69–100	121–180	41–113	4–20	5–18
R20	5	13–15	24–72	4–7	3–4	5–8
R30	5	15–23	28–99	7–11	4–6	5–8
R40	5	24–32	58–161	8–18	5–9	7–13
R50	5	23–39	82–169	13–20	6–12	5–17
ALB2	15	78–114	133–172	44–122	2–23	4–26
GRP	10	77–113	138–171	52–126	4–34	5–23
MAD	15	149–195	274–318	86–238	2–42	4–33
URP5	7	298–493	597–1,403	206–671	19–99	5–37
URP7	8	452–744	915–2,089	321–1,003	15–140	7–43

5.2. Experimental Results

The branch-and-cut algorithm was implemented in C++, and experiments were run on a 2.80 GH Intel Core i7 machine with 16 GB of memory. We used IBM CPLEX 12.7 Concert Technology with default parameters, except for the cuts generated by CPLEX, which were disabled because preliminary testing indicated that activating the CPLEX cuts produced worse results. The maximum computing time was set to 4 hours for instances in groups D, G, R, P, ALB, ALB2, GRP, and MAD, and to 24 hours for the larger instances in groups URP5 and URP7. Connectivity and parity cuts were separated at all nodes of the enumeration tree for all the tested formulations. As mentioned, the RtFCs (32) used in the two-index formulations are handled as lazy constraints.

Tables 3 and 4 show, for the MC-p-LARP and MC-LARP, respectively, the aggregated results obtained, for each group of instances, with the 3IF, its reinforcement with the optimality condition O5 (3IF-O5), and the 2IF. Columns "#Opt$_0$" and "Gap$_0$" report the number of instances in the group that were optimally solved at the root node and the average percentage gap at the root node with respect to the optimal or best known solution at termination. Similarly, the next two columns, "#Opt" and "Gap," give the same information at termination: the number of instances solved to optimality and the average percent gap with respect to the optimal or best known solution. Columns "Nodes" represent the average number of nodes explored in the search tree. Finally, the columns "CPU" give the average of the total computing times in seconds.

Note that the last two sets corresponding to the large instances were solved only with the two-index formulations. Furthermore, for these sets, we also increased the maximum computing time to 24 hours.

Our results show that, for both MC-p-LARP and MC-LARP, the two-index formulation is more efficient and faster than the two three-index formulations. The 2IF allowed us to solve all the small instances within a few minutes, reducing the computing times of the 3IF by 98%. In contrast, the 3IF and 3IF-O5 could not find optimal solutions on 18 instances within the limit time of four hours (15 MC-p-LARP instances and 3 MC-LARP instances). Moreover, with the 2IF, we could also solve all medium instances and one-third of the large ones. Finally, note that the number of nodes in the search tree is also smaller with the 2IF. Comparing Tables 3 and 4, it can be observed that the results are quite similar regardless of whether the number of facilities to be opened is restricted or setup costs are included in the objective function.

The superiority of the 2IF relative to the 3IF and 3IF-O5 is also reflected in the number of cuts required by each type of formulation, which is remarkably smaller on the two-index formulations. Whereas the number of connectivity cuts generated with the 3IF for the 100-node instances in groups D100 and G100 is, on average, around 25,000 (both for the MC-p-LARP and MC-LARP), with the 2IF, this number is usually smaller than 100 for the same instances. The situation is similar, although less extreme, with the parity cuts. For the mentioned instances, the 3IF requires, on average, 2,500 to 3,000 parity cuts, whereas the 2IF generates around 200 such cuts. One difference that can be observed between the two types of formulations is that three-index formulations require many more connectivity cuts than parity cuts, whereas the formulations with two index variables are much more balanced in terms of the number of cuts of each type that are generated, although they tend to generate more parity cuts than connectivity cuts. Concerning the RtFC inequalities used in the 2IF to guarantee that routes return to their starting depots, we have observed that they are very seldom needed. The vast majority of the instances were optimally solved without generating any RtFC. Only one or two RtFCs were generated for about 10% of the considered instances.

Comparing the two three-index formulations, it is easy to see that the 3IF-O5 outperforms the 3IF, in terms of the number of instances solved to optimality and, particularly, in terms of computing time. Nevertheless, as mentioned before, the original two-index formulations still outperform the three-index formulations even when these are reinforced with condition O5.

Tables 5 and 6 show the results for the models MC-p-LARP-UD and MC-LARP-UD, respectively, which extend the previous models by including a cardinality constraint on the number of users that can be served from

Table 3. Computational Results for the MC-p-LARP

	3IF						3IF-O5						2IF					
	#Opt$_0$	Gap$_0$ (%)	#Opt	Gap (%)	Nodes	CPU (s)	#Opt$_0$	Gap$_0$(%)	#Opt	Gap (%)	Nodes	CPU (s)	#Opt$_0$	Gap$_0$ (%)	#Opt	Gap (%)	Nodes	CPU (s)
D16	9/9	0			0	0.10	8/9	0.08	9/9	0	0.22	0.11	9/9	0			0	0.02
D36	3/9	1.62	9/9	0	51.56	17.35	3/9	1.63	9/9	0	31.22	9.81	8/9	0.52	9/9	0	1.22	0.14
D64	0/9	2.50	9/9	0	164.00	519.19	0/9	2.68	9/9	0	108.56	328.62	8/9	0.05	9/9	0	0.56	0.48
D100	0/9	4.00	2/9	2.39	551.56	12,361.85	0/9	3.59	4/9	1.31	639.56	11,230.97	2/9	0.63	9/9	0	11.22	12.41
G16	6/9	3.01	9/9	0	3.33	0.26	7/9	2.31	9/9	0	2.33	0.21	9/9	0			0	0.03
G36	3/9	2.31	9/9	0	13.00	10.08	3/9	2.13	9/9	0	4.33	2.87	8/9	0.49	9/9	0	0.33	0.10
G64	2/9	3.69	8/9	0.29	349.33	2,031.50	3/9	1.89	9/9	0	113.44	186.33	7/9	0.36	9/9	0	9.11	0.33
G100	1/9	2,551	2/9	24.96	24.96	12,846.15	1/9	2.06	5/9	1.20	173.56	8,931.24	5/9	0.36	9/9	0	2.78	0.95
R20	3/5	3.96	5/5	0	5.80	0.27	4/5	10.38	5/5	0	0.80	0.22	5/5	0			0	0.05
R30	2/5	2.55	5/5	0	4.80	1.77	3/5	0.13	5/5	0	1.80	1.75	5/5	0			0	0.08
R40	2/5	1.01	5/5	0	24.20	44.01	3/5	0.78	5/5	0	52.80	54.43	3/5	0.23	5/5	0	1.40	0.30
R50	3/5	1.16	5/5	0	6.40	31.32	4/5	0.40	5/5	0	4.20	35.80	4/5	0.07	5/5	0	0.40	0.21
P	13/24	1.56	24/24	0	19.13	11.19	13/24	1.31	24/24	0	5.38	2.62	17/24	0.31	24/24	0	2.00	0.20
ALB	0/2	2.10	2/2	0	369.50	4,739.15	0/2	2.24	2/2	0	70.50	1,015.20	2/2	0			0.50	2.38
ALB2	0/15	35.99	4/15	35.19	257.60	11,410.03	0/15	36.30	4/15	35.32	132.00	10,858.05	5/15	1.11	15/15	0	42.87	17.23
GRP	0/10	42.18	3/10	41.33	153.90	12,485.84	0/10	41.88	4/10	40.63	210.70	12,799.46	2/10	2.46	10/10	0	276.50	59.83
MAD	0/15	93.49	0/15	93.48	2.07	14,405.61	0/15	87.01	0/15	86.99	5.13	14,446.39	8/15	0.35	15/15	0	74.07	251.52
URP5													0/7	2.65	3/7	0.58	482.29	60,832.93
URP7													1/8	18.41	2/8	18.04	56.00	70,099.98

Table 4. Computational Results for the MC-LARP

	3IF						3IF-O5						2IF					
	#Opt$_0$	Gap$_0$ (%)	#Opt	Gap (%)	Nodes	CPU (s)	#Opt$_0$	Gap$_0$ (%)	#Opt	Gap (%)	Nodes	CPU (s)	#Opt$_0$	Gap$_0$ (%)	#Opt	Gap (%)	Nodes	CPU (s)
D16	5/9	2.39	9/9	0	3.33	0.16	5/9	1.37	9/9	0	1.22	0.12	9/9	0.00			0	0.08
D36	3/9	2.13	9/9	0	19.44	6.20	3/9	1.84	9/9	0	17.78	7.96	5/9	0.88	9/9	0	5.44	0.29
D64	0/9	3.02	9/9	0	80.44	180.43	0/9	3.01	9/9	0	45.22	127.89	3/9	0.51	9/9	0	4.44	1.19
D100	0/9	3.91	8/9	0.28	482.56	6,610.96	0/9	3.78	8/9	0.23	467.00	5,888.12	0/9	1.38	9/9	0	24.22	17.35
G16	8/9	0.79	9/9	0	0.56	0.17	8/9	0.85	9/9	0	0.22	0.12	7/9	0.73	9/9	0	0.78	0.06
G36	9/9	1.94	9/9	0	5.33	2.48	6/9	1.39	9/9	0	4.11	2.33	4/9	1.46	9/9	0	1.89	0.19
G64	3/9	1.78	9/9	0	11.67	28.01	2/9	1.94	9/9	0	11.44	29.86	3/9	0.97	9/9	0	5.11	0.50
G100	0/9	3.61	7/9	0.50	154.22	6,303.97	0/9	2.45	8/9	0.10	128.33	4,202.78	1/9	1.37	9/9	0	45.56	6.93
R20	4/5	0.14	5/5	0	0.20	0.13	4/5	0.40	5/5	0	0.40	0.23	5/5	0			0	0.06
R30	4/5	0.21	5/5	0	0.80	1.08	3/5	0.42	5/5	0	0.80	1.15	2/5	0.67	5/5	0	1.00	0.13
R40	3/5	0.43	5/5	0	17.00	33.97	3/5	0.43	5/5	0	18.80	28.68	2/5	1.20	5/5	0	12.60	0.78
R50	3/5	0.67	5/5	0	10.00	55.23	3/5	0.68	5/5	0	12.60	54.80	2/5	3.24	5/5	0	12.40	0.56
P	10/24	2.27	24/24	0	13.25	4.33	9/24	3.26	24/24	0	12.08	5.12	8/24	2.52	24/24	0	11.38	0.43
ALB	0/2	2.45	2/2	0	131.50	1,225.94	0/2	2.59	2/2	0	57.00	882.60	1/2	1.12	2/2	0	3.50	2,09
ALB2	0/15	23.96	9/15	15.34	162.00	8,673.03	0/15	23.18	9/15	20.91	130.60	7,945.71	4/15	1.66	15/15	0	93.87	36.75
GRP	0/10	33.28	5/10	31.03	145.10	9,111.73	0/10	32.36	5/10	30.72	174.20	8,217.41	1/10	2.99	10/10	0	260.70	104.90
MAD	0/15	80.83	0/15	80.61	8.60	14,410.02	0/15	80.49	0/15	80.48	7.93	14,403.95	4/15	0.44	15/15	0	92.53	547.17
URP5													0/7	1.78	4/7	0.50	449.71	45,425.08
URP7													0/8	21.43	2/8	21.09	23.50	61,583.30

each open facility. As mentioned above, these models had to be treated with the three-index formulation to recreate the routes from each facility once the values of the decision variables were known. As before, the behavior is similar for the two models MC-p-LARP-UD and MC-LARP-UD. However, comparing the results with those for the corresponding version without cardinality constraints, we can see, as expected, that the cardinality version is more difficult. This translates into a lower number of instances optimally solved, a larger number of explored nodes, and an increase in the computing time.

Tables 7 and 8 report the results obtained with the two-index formulation for the models in which the min-max objective function is considered. Dealing with this kind of objective is typically difficult. Consequently, the results obtained for these models are the worst ones, with the lowest number of instances optimally solved and the largest computing time. In spite of this, the proposed algorithm found a proven optimal solution for the 62% of the tested instances.

As mentioned, in the benchmark instances that we have generated, there is no component with more than one potential facility. As we explain below, this characteristic has very little effect on the results we have obtained. On the one hand, for the models where optimality condition O5 holds, instances with more than one potential facility in some component can be a priori transformed into equivalent instances with at most one potential facility per component by arbitrarily selecting one potential facility for components with several candidates in the case of the MC-p-LARP, or by identifying the candidate facility of the component with the minimum setup cost for the MC-LARP. On the other hand, for the models where the optimality condition O5 does not hold, the results reported in Table 9 suggest that allowing for several potential facilities in some of the components would have no significant effect on the performance of our algorithms. From Table 9 it can be seen that for MC-p-LARP-UD and MC-LARP-UD, the average number of open facilities is smaller than the number of potential facilities. Moreover, for the MC-p-LARP-UD, this number is smaller than the parameter p. The lack of effect of having at most one potential facility per component has been confirmed with an additional computational test using the instances of group D16, but with more potential locations in some components. The results with the modified instances show that the difference in the behavior of the algorithm in terms of computational time or number of nodes on the exploration tree is negligible.

Table 5. Computational Results for the MC-p-LARP-UD

	$\#\text{Opt}_0$	Gap_0 (%)	#Opt	Gap (%)	Nodes	CPU (s)
D16	4/9	8.01	9/9	0	7.11	0.49
D36	0/9	9.97	9/9	0	951.56	579.66
D64	0/9	15.14	3/9	11.04	2,079.11	10,947.88
D100	0/9	50.23	0/9	48.58	859.11	14,401.60
G16	4/9	6.74	9/9	0	21.56	0.78
G36	0/9	6.42	9/9	0	1,022.67	673.96
G64	1/9	22.53	4/9	19.95	1,668.78	10,712.95
G100	0/9	51.00	1/9	50.48	583.11	12,957.32
R20	2/5	4.19	5/5	0	10.20	0.75
R30	1/5	1.75	5/5	0	33.20	5.48
R40	1/5	7.18	5/5	0	551.40	1,130.02
R50	0/5	6.61	5/5	0	234.20	460.44
P	7/24	4.66	23/24	0.44	207.67	612.74
ALB	0/2	53.13	½	50.00	967.00	7,863.31

Table 6. Computational Results for the MC-LARP-UD

	$\#\text{Opt}_0$	Gap_0 (%)	#Opt	Gap (%)	Nodes	CPU (s)
D16	2/9	8.01	9/9	0	16.44	0.62
D36	0/9	8.20	9/9	0	934.22	570.73
D64	0/9	16.76	4/9	12.75	2,151.56	10,355.26
D100	0/9	44.04	0/9	42.45	871.78	14,403.16
G16	3/9	6.93	9/9	0	18.22	1.11
G36	0/9	6.60	9/9	0	819.11	532.60
G64	0/9	24.11	4/9	21.50	1,185.44	9,107.25
G100	0/9	45.51	0/9	44.95	725.11	14,402.83
R20	3/5	3.17	5/5	0	5.80	0.68
R30	2/5	2.52	5/5	0	42.00	7.01
R40	1/5	8.84	4/5	1.25	517.60	2,985.65
R50	0/5	7.43	5/5	0	160.60	545.64
P	6/24	4.61	23/24	0.52	269.96	620.02
ALB	0/2	51.03	½	50.00	642.00	7,810.20

5.3. Analysis of the Solutions: Cross Comparison of the Models

We close the computational experiments section by analyzing some characteristics of the solutions produced by the different models. The results concerning the number of facilities open in the optimal solutions of the different formulations are summarized in the Table 9. As could be expected, when the objective takes into account the overall routing costs, models with facility setup costs (MC-LARP and MC-LARP-UD)

Table 7. Computational Results for the MM-p-LARP

	$\#\text{Opt}_0$	Gap_0 (%)	#Opt	Gap (%)	Nodes	CPU (s)
D16	1/9	34.65	9/9	0	114.67	5.56
D36	0/9	51.08	5/9	1.37	8,729.11	9,586.92
D64	0/9	55.09	1/9	42.94	1,352.56	13,620.39
D100	0/9	100.00	0/9	100.00	121.11	14,401.83
G16	1/9	37.78	9/9	0	60.89	5.07
G36	0/9	39.62	5/9	3.31	2,377.22	7,530.66
G64	0/9	56.92	0/9	42.25	530.78	14,400.59
G100	0/9	100.00	0/9	100.00	20.44	14,401.47
R20	0/5	57.03	5/5	0	286.20	16.13
R30	0/5	54.30	5/5	0	482.60	69.92
R40	0/5	64.63	3/5	9.81	377.00	6,496.05
R50	0/5	69.24	2/5	47.86	7,716.00	9,633.41
P	2/24	27.75	18/24	2.62	2,169.04	3,715.40
ALB	0/2	75.79	0/2	69.66	329.50	14,401.47

Table 8. Computational Results for the MM-p-LARP-UD

	#Opt$_0$	Gap$_0$ (%)	#Opt	Gap (%)	Nodes	CPU (s)
D16	0/9	45.37	9/9	0	204.78	12.20
D36	0/9	51.24	6/9	1.65	7,676.11	9,670.18
D64	0/9	55.48	1/9	40.34	1,400.00	13,208.01
D100	0/9	100.00	0/9	100.00	201.11	14,405.84
G16	1/9	37.76	9/9	0	50.22	6.21
G36	0/9	47.13	6/9	12.68	1,405.00	6,726.02
G64	0/9	58.91	0/9	46.28	2,551.89	14,400.05
G100	0/9	100.00	0/9	100.00	9.11	14,400.61
R20	0/5	55.93	5/5	0	272.20	22.47
R30	0/5	54.16	5/5	0	844.20	259.71
R40	0/5	64.66	4/5	8.82	3,311.20	6,300.43
R50	0/5	68.19	0/5	62.26	1,440.80	11,523.14
P	0/24	33.39	19/24	2.38	2,681.96	3,487.50
ALB	0/2	100.00	0/2	100.00	370.00	14,400.09

produce, in general, solutions with a smaller number of open facilities than the models where the maximum number of open facilities is limited only by the parameter p (MC-p-LARP, MC-LARP). In particular, MC-LARP produces solutions that, on average, have 33% fewer open facilities than those for the MC-p-LARP. This reduction is not so evident for the corresponding models with unit demands and capacity constraints, where the MC-LARP-UD produces solutions that, on average, have around 13% fewer facilities than those for the MC-p-LARP-UD. Similarly, models with unit demands (MC-p-LARP-UD, MC-LARP-UD) produce, in general, optimal solutions with more open facilities than their nondemand counterparts (MC-p-LARP, MC-LARP). On the contrary, it can be observed that unit demand constraints have very little effect on the number of open facilities in the optimal solutions of models with a makespan objective. The MM-p-LARP and MM-p-LARP-UD produce solutions with a very similar number of open facilities; there are only 5 instances out of 98 where the optimal MM-p-LARP-UD solution opens one more facility than the optimal MM-p-LARP solution.

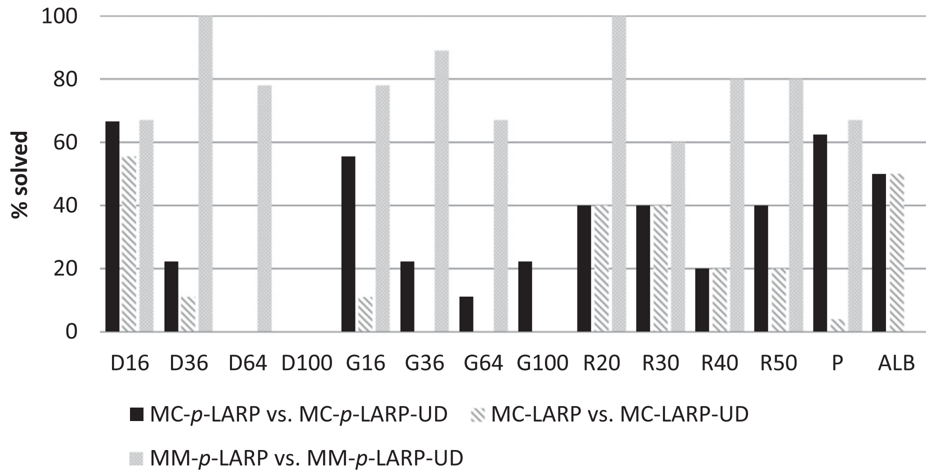
Because the models with capacity constraints have been shown to be notably more difficult to solve than their uncapacitated counterparts, we have also investigated how often optimal solutions to models without capacity constraints are feasible (and therefore optimal) for their capacitated versions. Figure 3 illustrates that the makespan model is clearly more successful in this respect, producing a percentage of feasible solutions for its capacitated counterpart that ranges from 60 to 100, depending on the type and size of the instances. In contrast, the capability of producing feasible solutions for their capacitated versions of the models that include the overall routing costs in their objective is quite small, particularly for the more time-consuming instances. It is worth noting that no optimal solution to the MC-p-LARP or MC-LARP was feasible for the MC-p-LARP-UD or MC-LARP-UD, respectively, with the D64 and D100 sets of instances.

We also analyze the *robustness* of the uncapacitated models (MC-p-LARP, MC-LARP, MM-p-LARP), measured in terms of their capability of producing good quality solutions for the other models. For this, the optimal solutions to each model in $\mathcal{F} = \{MC-\text{p}-LARP, MC-LARP, MM-\text{p}-LARP\}$ has been evaluated relative to the objectives of the other models and compared with their optimal values. In particular, let $\overline{x}^i$ denote an optimal solution to formulation $i \in \mathcal{F}$ for a given instance, and $\overline{v}^i$ its optimal value. Also let v^{ij} denote the objective function value of solution $\overline{x}^i$, relative to the objective function of formulation $j \in \mathcal{F}, j \neq i$. Table 10 gives, for each model $i \in \mathcal{F}$, the averages of the percentages $100(v^{ij} - \overline{v}^i)/\overline{v}^i$ over all the instances of each set of benchmark instances, for each model $j \neq i$.

Table 9. Average Number of Open Facilities in the Optimal Solutions of the Different Models

	MC-p-LARP	MC-LARP	MC-p-LARP-UD	MC-LARP-UD	MM-p-LARP	MM-p-LARP-UD
D16	3.33	3.11	3.56	3.22	3.67	4.00
D36	2.56	1.56	4.00	3.44	4.00	4.00
D64	2.22	1.44	3.67	3.00	3.56	4.00
D100	3.67	1.89	2.44	2.71		
G16	2.33	1.22	3.11	2.78	3.78	4.00
G36	2.56	1.22	3.67	2.78	4.00	4.00
G64	1.78	1.00	3.38	2.50	4.00	4.00
G100	2.56	1.11	3.00	2.50		
R20	2.00	2.00	2.60	2.60	4.00	3.80
R30	2.60	2.20	3.20	3.20	4.00	4.00
R40	2.80	2.40	3.60	3.60	4.00	4.00
R50	3.20	2.40	3.60	3.20	3.80	4.00
P	3.38	1.13	3.46	2.67	3.67	3.58
ALB	3.50	3.00	4.00	3.00	3.00	
Average	2.75	1.83	3.38	2.94	3.79	3.94

Figure 3. Percentage of Optimal Solutions of Uncapacitated Models That Are Feasible for the Capacitated Counterpart



As can be seen from Table 10, the models that include the overall routing costs produce, in general, solutions that are not good for the makespan objective. This is particularly true for the MC-LARP, which includes the facility setup cost in the objective. The converse holds because the makespan model also produces optimal solutions that, in general, are not of good quality for the MC-p-LARP or the MC-LARP. On the other hand, not surprisingly, the MC-p-LARP produces, in general, optimal solutions that are good for the MC-LARP, and vice versa. In this sense, the obtained results show a slight superiority of the MC-LARP over the MC-p-LARP.

Finally, Tables 11 and 12 show the impact of parameters p and f_d on the characteristics of optimal solutions. The average number of facilities opened, the total cost of all the routes in the group, and the average computing times in seconds are given in columns under the headings #D, "Total cost," and "CPU (s)," respectively. As was expected, reducing the maximum number of open facilities increases the total cost (see Table 11). Table 12 illustrates the effect of the magnitudes of the setup costs on the relation between the number of facilities opened and the total cost. Smaller setup costs allow the opening of a larger number of facilities without increasing the overall cost. In contrast, the instances with larger setup costs produce solutions with fewer opened facilities, thus increasing the total cost. Furthermore, from Tables 11 and 12, it can be seen that the behavior of the algorithm in terms of computing time remains stable when varying the values of the analyzed parameters.

6. Conclusions

We have modeled and solved several LARPs with different characteristics. The models differ from each other in their objective functions, on whether the number of facilities to be located is upper bounded, or on whether the facilities are capacitated. We have considered min-cost objectives aiming at minimizing the overall routing costs (possibly incorporating facility setup costs as well) and min–max objectives aiming at minimizing the makespan. Some of the studied models assume that there are no capacity limitations, whereas other models

Table 10. Cross Comparison of Optimal Values of the Different Models

	MC-p-LARP		MC-LARP		MM-p-LARP	
	MC-LARP	MM-p-LARP	MC-p-LARP	MM-p-LARP	MC-p-LARP	MC-LARP
D16	1.20	12.63	3.25	20.00	7.86	8.35
D36	2.66	128.20	1.34	178.05	21.82	29.92
D64	1.08	104.65	0.50	138.48	14.15	18.07
D100	2.04		0.56			
G16	9.61	63.89	5.00	187.96	30.57	66.67
G36	8.60	85.40	1.96	168.47	14.13	30.58
G64	3.36	130.23	0.51	159.87	36.80	45.82
G100	3.59		2.17			
R20	1.38	55.06	1.78	55.06	49.46	54.44
R30	2.50	52.10	1.33	93.45	36.09	40.56
R40	1.40	118.75	1.19	138.99	20.63	21.82
R50	1.08	29.49	2.94	51.90	30.83	30.92
P	14.79	36.21	15.23	197.06	14.18	26.23

Table 11. Sensitivity Analysis on the Value of p

	#D			Total cost			CPU (s)		
	$p = 4$	$p = 3$	$p = 2$	$p = 4$	$p = 3$	$p = 2$	$p = 4$	$p = 3$	$p = 2$
MC-p-LARP	3.33	2.89	1.89	5,274	5,486	6,215	0.02	0.10	0.27
MC-p-LARP-UD	3.56	3.00	2.00	6,181	6,533	3,262[a]	0.40	0.64	1.67

[a]Only four instances are feasible with $p = 2$.

Table 12. Sensitivity Analysis on the Setup Costs f

	#D			Total cost			CPU (s)		
	f_d	$\frac{1}{2}f_d$	$2f_d$	f_d	$\frac{1}{2}f_d$	$2f_d$	f_d	$\frac{1}{2}f_d$	$2f_d$
MC-p-LARP	3.00	3.33	2.56	6,120	5,710	6,854	0.08	0.08	0.27
MC-p-LARP-UD	3.22	3.56	3.00	7,060	6,637	7,821	0.62	0.76	0.80

include a cardinality constraint on the number of users that can be served from an open facility.

Three-index variable formulations have been presented for all the models. The polyhedral analysis carried out for the three-index formulation of the uncapacitated models proves that the main families of constraints are facet defining. Moreover, a two-index variable formulation was introduced for the min-cost models without capacity constraints, which incorporates a new set of constraints forcing the routes return to their departing facility. All the formulations exploit optimality conditions, which allows using binary decision variables only.

Exact and heuristic separation procedures have been studied for the large-size families of inequalities and an exact branch-and-cut solution algorithm was implemented for the solution of the proposed formulations. Our numerical results demonstrate the good behavior of the algorithm, which was tested on several sets of benchmark instances. For the uncapacitated min-cost models, all instances involving up to 200 depots, as well as most instances involving up to 744 vertices, were solved to optimality. Despite the difficulty of the models with a makespan objective or with capacity constraints, instances with up to 100 vertices were optimally solved for the makespan objective and for the capacitated versions of the min-cost models. When comparisons are possible, our results show the superiority of the two-index formulation in terms of efficiency and speed with respect to the three-index formulations.

We believe that developments similar to those presented in this paper can be carried out for some problems where the demand is located at the nodes of the input graph. A promising avenue of research is to develop two-index variable formulations for some node-routing problems with multiple routes. The limitations of such an approach would be similar to those of this paper. Thus, in principle, it seems viable for problems with min-cost objectives and without capacity constraints like the m-traveling salesman problem, uncapacitated multidepot vehicle routing problems, or some location node routing problems. In all these cases, constraints guaranteeing that the routes return to their starting depots would be needed to ensure the validity of the formulations with the two-index variables.

Acknowledgments

The authors thank the referees for their valuable comments.

References

Albareda-Sambola M (2015) Location-routing and location-arc routing. Laporte G, Nickel S, Saldanha da Gama F, eds *Location Science* (Springer, Berlin), 399–418.

Amaya A, Langevin A, Trépanier M (2007) The capacitated arc routing problem with refill points. *Oper. Res. Lett.* 35(1):45–53.

Arbib C, Servilio M, Archetti C, Speranza MG (2014) The directed profitable location rural postman problem. *Eur. J. Oper. Res.* 236(3):811–819.

Baldacci R, Maniezzo V (2006) Exact methods based on node-routing formulations for undirected arc-routing problems. *Networks* 47(1):52–60.

Christofides N, Campos V, Corberán Á, Mota E (1981) An algorithm for the rural postman problem. Technical Report IC.O.R.81.5, Imperial College London, London.

Corberán Á, Laporte G (2014) *Arc Routing: Problems, Methods, and Applications*. MOS-SIAM Series on Optimization, vol. 20 (Society for Industrial and Applied Mathmatics, Philadelphia).

Corberán Á, Sanchis JM (1994) A polyhedral approach to the rural postman problem. *Eur. J. Oper. Res.* 79(1):95–114.

Corberán Á, Sanchis JM (1998) The general routing problem polyhedron: Facets from the RPP and GTSP polyhedra. *Eur. J. Oper. Res.* 108(3):538–550.

Corberán Á, Plana I, Sanchis JM (2007) A branch & cut algorithm for the windy general routing problem. *Networks* 49(4):245–257.

Drexl M, Schneider M, Scheneker DB (2013) A survey of location-routing problems. Technical report, Institute for Business Studies, Department of Business Administration, Economics and Law, Darmstadt Technical University, Darmstadt, Germany.

Fernández E, Rodríguez-Pereira J (2017) Multi-depot rural postman problems. *TOP* 25(2):340–372.

Fernández E, Laporte G, Rodríguez-Pereira J (2018) A branch-and-cut algorithm for the multi-depot rural postman problem. *Transportation Sci.* 52(2):353–369.

Ghiani G (1998) Arc routing, allocation-arc routing and location-arc routing problems. Unpublished doctoral dissertation, University of Naples Federico II, Naples, Italy.

Ghiani G, Laporte G (1999) Eulerian location problems. *Networks* 34(4):291–302.

Ghiani G, Laporte G (2000) A branch-and-cut algorithm for the undirected rural postman problem. *Math. Programming* 87(3): 467–481.

Ghiani G, Improta G, Laporte G (2001) The capacitated arc routing problem with intermediate facilities. *Networks* 37(3): 134–143.

Hashemi Doulabi SH, Seifi A (2013) Lower and upper bounds for location-arc routing problems with vehicle capacity constraints. *Eur. J. Oper. Res.* 224(1):189–208.

Hertz A, Laporte G, Nanchen Hugo P (1999) Improvement procedures for the undirected rural postman problem. *INFORMS J. Comput.* 11(1):53–62.

Hierholzer C (1873) Über die Mögglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechnung zu umfahren. *Math. Ann.* 6(1):30–32.

Levy L, Bodin LD (1989) The arc oriented location routing problem. *INFOR* 27(1):74–94.

Longo H, Poggi de Aragão M, Uchoa E (2006) Solving capacitated arc routing problems using a transformation to the CVRP. *Comput. Oper. Res.* 33(6):1823–1837.

Lopes RB, Plastria F, Ferreira C, Santos BS (2014) Location-arc routing problem: Heuristic approaches and test instances. *Comput. Oper. Res.* 43:309–317.

Min H, Jayaraman V, Srivastava R (1998) Combined location-routing problems: A synthesis and future research directions. *Eur. J. Oper. Res.* 108(1):1–15.

Nagy G, Salhi S (2007) Location-routing: Issues, models and methods. *Eur. J. Oper. Res.* 177(2):649–672.

Pearn WL, Assad AA, Golden BL (1987) Transforming arc routing into node routing problems. *Comput. Oper. Res.* 14(4):285–288.

Pia A, Filippi C (2006) A variable neighborhood descent algorithm for a real waste collection problem with mobile depots. *Internat. Trans. Oper. Res.* 13(2):125–141.

Prodhon C, Prins C (2014) A survey of recent research on location-routing problems. *Eur. J. Oper. Res.* 238(1):1–17.

Salazar-Aguilar MA, Langevin A, Laporte G (2013) The synchronized arc and node routing problem: Application to road marking. *Comput. Oper. Res.* 40(7):1708–1715.