

New path elimination constraints for multi-depot routing problems

Tolga Bektas¹, Luís Gouveia², Daniel Santos²

¹ Centre for Operational Research, Management Science and Information Systems (CORMSIS)
Southampton Business School, University of Southampton
Southampton, Highfield, SO17 1BJ, United Kingdom
`t.bektas@soton.ac.uk`

² Centro de Matemática, Aplicações Fundamentais e Investigação Operacional (CMAF-CIO)
DEIO, Faculdade de Ciências, Universidade de Lisboa
C6 - Piso 4, 1749-016, Lisbon, Portugal
`legouveia@fc.ul.pt`, `drsantos@fc.ul.pt`

Abstract

Multi-depot routing problems arise in distribution logistics where a set of vehicles based at several depots are used to serve a number of clients. Most variants of this problem have the basic requirement that the route of each vehicle starts and ends at the same depot. This paper describes new inequalities, namely multi-cut constraints (MCC), that enforce this requirement in mathematical programming formulations of multi-depot routing problems. The MCCs are exponential in size, and are equivalent to a compact three-index formulation for the problem in terms of the associated linear programming relaxations. The paper describes how a generalization of the MCCs can be obtained, in a similar manner, by using a stronger version of the three-index formulation. The connection between the compact and the exponential formulations implies a separation procedure based on max-flow/min-cut computations, which has reduced complexity in comparison with a previously known set of constraints described for the same purpose. The new inequalities are used in a branch-and-cut algorithm. Computational results are presented for instances with up to 300 clients and 60 depots.

Keywords: multi-depot routing; branch-and-cut; separation; traveling salesman; integer linear programming; reformulation.

1 Introduction

Multi-depot routing problems (MDRPs) arise in serving a set of clients from at least two depots, with at least one vehicle available at each depot, and where the aim is to minimize the total cost of the routing, often measured by the total distance traveled by all the vehicles. The depot locations can either be fixed, or are alternatively chosen from a potential set of locations. The two types of restrictions that are typically found in such problems are that (i) each client is served in one and only one route, and (ii) each route contains exactly one depot. The latter set of constraints require further elaboration. In particular, if a route does not contain a depot, it will be one of client nodes alone and disconnected from the depots. Inequalities that prevent the formation of such routes are known as subtour elimination constraints. Alternatively, if a route contains two or more depots, this implies that there exists a vehicle traveling on a path between two different depots. When a vehicle needs to return to its home base after the travel, as is typically required in MDRPs, any such paths that exist between pairs of depots will not be acceptable. We refer to such paths as infeasible paths, and to inequalities that disallow such paths as path elimination constraints.

Valid mathematical programming formulations for MDRPs need to include sets of constraints to prohibit both subtours and infeasible paths. Whilst the literature on subtour elimination constraints is rich (see, e.g., Öncan et al. 2009, Godinho et al. 2011, Roberti & Toth 2012, for surveys on the topic), this is not the case for path elimination constraints within multi-depot routing problems. Earlier work on such constraints goes as far back as Laporte et al. (1983) in the context of location-routing problems, who describe the so-called chain barring constraints. Later, Laporte et al. (1986) build on the chain barring constraints, leading to an improved set of path elimination constraints which have been further improved by Benavent & Martínez-Sykora (2013). These inequalities have also been used in a recent study by Sundar & Rathinam (2017). A slightly different set of path elimination constraints, based on a new set of variables, was proposed by Belenguer et al. (2011).

Benavent & Martínez-Sykora (2013) describe one type of an MDRP that is very similar to one of the variants investigated in this paper. The authors present the multi-depot multiple TSP (MDMTSP) and study the related polyhedron, where they show that the path elimination constraints used in their problem are facet-defining under very mild conditions. In addition, they present the conditions under which inequalities that are facet-defining for the TSP are also facet-defining for the MDMTSP. They also present a new set of comb-like inequalities that utilize the multi-depot characteristic of the problem and prove that they are also facet-defining. All inequalities, along with exact and heuristic separation algorithms, are used within a branch-and-cut algorithm for the MDMTSP that allows them to solve instances of up to 280 nodes and 25 depots within 20 minutes of computational time.

This paper extends the line of work on inequalities that eliminate infeasible paths that might arise in multi-depot routing problems. We study two main variants of the problem, one with fixed depots and the other where depots can be chosen from within a set of candidate sites, both in their simplest setting. In particular, we do not consider any additional constraints on, e.g., capacity or time, to be able to study the effect of the new inequalities in isolation from the potential impact of any other additional constraint. The contributions of this paper are as follows: (i) we propose new path elimination inequalities, called multi-cut constraints (MCCs), valid for both variants of the problem and for extensions involving multiple vehicles per depot; (ii) using projection, we show the connections between the basic version of the MCCs and a compact three-index formulation (i.e., with a polynomial number of constraints and variables), and then present a generalization of the MCCs by using a stronger version of the three-index formulation; (iii) we present theoretical and computational comparisons between the MCCs and the existing path elimination constraints, and (iv) we describe a branch-and-cut algorithm that uses the MCCs and present computational results on a variety of instances with up to 300 clients and 60 depots.

2 Routing Problems with Multiple Depots

The MDRP is defined on a directed graph $G = (V, A)$, with a set $V = \{1, \dots, n\}$ of nodes, a set $A = \{(i, j) : i, j \in V, i \neq j\}$ of arcs, and a cost function c associated to each arc. To simplify the presentation, we will assume that G is a complete graph, however, this is not required as the ensuing exposition can easily be adapted to incomplete graphs by simply not considering the pairs (i, j) such that $(i, j) \notin A$ in the mathematical expressions. The node set V is partitioned into a set D of potential depots (with $|D| = p$) and a set C of clients (with $|C| = n - p$). There are k_d vehicles available at each depot $d \in D$. The objective of the MDRP is to find a minimum cost set of circuits such that each node in C is in one and only one circuit, and that at most k_d circuits depart from and

end at depot $d \in D$.

We consider two variants which arise depending on whether or not every depot must be part of a route, or equivalently, at least one vehicle from each depot $d \in D$ must be used. If the latter constraint is imposed, we denote the resulting variant as Fixed-MDRP (F-MDRP), which requires that all depots must be used. Otherwise, the former variant arises where additional decisions need to be taken as to whether a depot should be used, which we denote as the Location-MDRP (L-MDRP). The L-MDRP is also known as the Location-Routing Problem (see, e.g., Albareda-Sambola 2015) where there is often a fixed cost of using a depot, which we do not consider in this study for reasons similar to excluding additional constraints, but such costs can easily be incorporated to the models described in the ensuing exposition.

As the arcs between depots can never be used, the size of the arc set A can be reduced by removing every arc that links two depots. In order to simplify the text, we will still refer to this “improved” arc set as A . In addition, let $A^c = \{(i, j) \in A : i, j \in C, i \neq j\}$ be the set of arcs between client nodes, and for all $d \in D$ let $A^d = \{(d, i) \in A : i \in C\} \cup \{(i, d) \in A : i \in C\}$ be the set of arcs with an endpoint in d . Note that $A = A^c \cup (\cup_{d \in D} A^d)$.

We first consider a single-vehicle version of the problem which means that every depot houses one vehicle, which is assumed to have unlimited capacity (as in, e.g., parcel deliveries or postal distribution). In other words, we assume that $k_d = 1$ for every depot $d \in D$. In a later section we will show how to easily adapt the new formulations to variants of the problem where $k_d > 1$.

In the single-vehicle F-MDRP, the objective is to find a minimum cost set of exactly p disjoint circuits on graph G such that:

- Each route starts at one of the depots and ends at the same depot, and
- Each client in C is visited exactly once.

In the single-vehicle version of the L-MDRP, the objective is to find a minimum cost set of at most p disjoint circuits on graph G such that:

- Each route starts from and ends at the same depot (provided the vehicle at that depot is used), and
- Each client in C is visited exactly once.

Both problem variants can be formulated by using two sets of binary variables: variables x_{ij} indicating whether or not arc $(i, j) \in A$ is used in one of the circuits, and variables y_d indicating whether or not the vehicle on depot $d \in D$ is used in one of the circuits. The following is a generic valid integer linear programming formulation of the location variant of the problem:

$$\begin{aligned}
& \text{Minimize} && \sum_{(i,j) \in A} c_{ij} x_{ij} \\
& \text{subject to:} && \sum_{j \in C} x_{dj} = y_d && \forall d \in D && (1) \\
& && \sum_{j \in C} x_{jd} = y_d && \forall d \in D && (2) \\
& && \sum_{j \in V} x_{ij} = 1 && \forall i \in C && (3) \\
& && \sum_{j \in V} x_{ji} = 1 && \forall i \in C && (4) \\
& && \{(i, j) \in A : x_{ij} = 1\} \text{ contains no circuit with zero depots} && && (5) \\
& && \{(i, j) \in A : x_{ij} = 1\} \text{ contains no circuit with two or more depots} && && (6) \\
& && x_{ij} \in \{0, 1\} && \forall (i, j) \in A && (7) \\
& && y_d \in \{0, 1\} && \forall d \in D. && (8)
\end{aligned}$$

The fixed variant of the problem is obtained when we add constraints

$$y_d = 1 \quad \forall d \in D. \quad (9)$$

In this case, the variables y_d can be eliminated from the model. We can also consider a mix of the fixed and location variants by setting $y_d = 1$ for only a subset of D . For reasons of simplicity, we omit this variant in the remainder of the text.

The system (1)–(4) and (7) is quite similar to the well-known assignment relaxation that appears in formulations for the traveling salesman problem (TSP) and several of its variants. It is clear that any feasible solution of the assignment relaxation is a set of disjoint circuits. In the case of the F-MDRP every node is included in one such circuit whereas in the L-MDRP some depots may not be used in the solution. Such a solution, however, may not be feasible for the variant being modeled, namely the F-MDRP or the L-MDRP, since it may include circuits with no depots, or circuits with two or more depots. For this reason, the generic constraints (5) and (6) are used to forbid such solutions, respectively. In the following sections we discuss modeling approaches that address both situations.

Before initiating the discussion on sets of constraints to model (5) and (6) we introduce some notation to simplify the presentation.

$$\begin{aligned} x(S) &= \sum_{i,j \in S, i \neq j} x_{ij} \\ x(S_1, S_2) &= \sum_{i \in S_1, j \in S_2, i \neq j} x_{ij} \\ y(D') &= \sum_{d \in D'} y_d. \end{aligned}$$

For singleton subsets (of clients or depots), say $I = \{i\}$, we write i instead of $\{i\}$. We denote the linear programming (LP) relaxation of a formulation F by F_L , and use $v(F)$ to denote the optimal value of formulation F . Finally, we use $\mathcal{P}_{x,y}(P)$ to denote the projection of a polyhedron P onto the space of the (x, y) variables.

2.1 Eliminating circuits without any depot

In this section, we briefly describe inequalities that can be used in place of (5) to eliminate circuits with zero depots, or, equivalently, client-only circuits. These inequalities have been widely studied in the context of the TSP and its variants, and can be easily adapted for the MDRP. Consider the following subtour elimination constraints:

$$x(S) \leq |S| - 1 \quad \forall S \subseteq C, S \neq \emptyset. \quad (10)$$

By using the assignment constraints (4), inequalities (10) can be equivalently written in cut form as follows:

$$x(D \cup (C \setminus S), S) \geq 1 \quad \forall S \subseteq C, S \neq \emptyset. \quad (11)$$

Observe that in the case of the fixed variant, the subset S in the two previous constraints can be restricted to $|S| \leq |C| - |D|$ since each depot must be linked to at least one client node.

Although exponential in size, inequalities (10) (or (11)) can be combined with a cutting plane approach provided that there exists a procedure for implicitly adding them to the formulation. In fact, it is well-known that inequalities (11) can be separated in polynomial time by max-flow/min-cut computations (see, e.g., Ahuja et al. 1993). In addition, there exists a compact flow based formulation that is equivalent in terms of the corresponding linear programming relaxation bound to the formulation involving the inequalities (11). Again, this is a direct consequence of the max-flow/min-cut theorem and the corresponding flow based formulation is closely associated to the separation routine for the cut inequalities.

The literature on the TSP and its variants is rich, and includes surveys (see, e.g., Öncan et al. 2009, Godinho et al. 2011, Roberti & Toth 2012) on subtour elimination constraints and their equivalent flow based formulation, as well as other types of constraints to deal with client-only circuits, both compact and exponential in size. These surveys also include comparisons of polyhedra, and present different models with possibly weaker linear programming relaxation bounds but which can be used to solve medium-sized problems for practical purposes.

Given the extensive research that exists on modeling approaches for eliminating client-only circuits in routing problems, we will restrict our attention to those used to eliminate circuits with two or more depots, which are more relevant to the multi-depot routing problems. Further details are given below.

2.2 Eliminating circuits with more than one depot

In this section we present several modeling approaches for eliminating paths between different depots, or, equivalently, circuits with more than one depot, that is, approaches to model the generic inequalities (6).

2.2.1 Multi-cut inequalities

We start by presenting a new set of inequalities that involve only the x and the y variables.

Proposition 1. *The following multi-cut constraints are valid for the MDRP and eliminate circuits with two or more depots:*

$$x(C \setminus S, d) + x(C \setminus S, S) + x(d, S) \geq y_d \quad \forall d \in D, \forall S \subseteq C. \quad (12)$$

Proof. Constraints (12) are clearly valid if $y_d = 0$ or if $S = C$ or $S = \emptyset$. Assume then that $y_d = 1$, let $S' = C \setminus S$ and consider first the situation where $x(S', d)$, $x(S', S)$ and $x(d, S)$ are all 0. Due to the assignment constraints (1) and (2), there must exist an arc connecting the depot d to a node in S' (since $x(d, S) = 0$) and another arc connecting a node in S to the depot d (because $x(S', d) = 0$). But then, since $x(S', S) = 0$ there is no way of closing the circuit. To see why circuits with two or more depots are eliminated consider, without loss of generality, that a circuit contains two depots, d_1 and d_2 , from D (obviously, such that $y_{d_1} = y_{d_2} = 1$), and define S' as the set of client nodes which are in the path between depot d_1 and depot d_2 . Thus, $x(S', d_1) = x(d_1, S) = x(S', S) = 0$ and so the inequality (12) for depot d_1 is violated. \square

We will refer to inequalities (12) as 1-MCC. It is also easily shown that with these constraints alone we obtain a valid representation of (6). We denote by $F^=$ the model defined by (1)–(4), (11), (12), (7) and (8).

Given the form in which inequalities (12) are presented, it might be more appropriate to view them as path forcing constraints rather than path elimination constraints, in that they force paths to start and end at the same depot. In Section 2.4, we will show an alternative form of these inequalities that is more typical of elimination inequalities. We believe that this alternative representation is a unique characteristic of the cuts we present in this paper.

2.2.2 Connections with a 3-index formulation

Arguably the most intuitive formulation to ensure circuits that only have one depot (i.e., routes starting from and ending at the same depot) is based on binary variables z_{ij}^d which have value 1 if arc $(i, j) \in A$ is used in the circuit of depot d , and 0 otherwise. Four such similar formulations are given by Albareda-Sambola et al. (2005), Bektas (2012), Fernández & Rodríguez-Pereira (2016) and by Hill & Voß (2016) for variants of multi-depot problems. Clearly, an arc with an endpoint in a depot $d \in D$ cannot be used in the circuit of depot $d' \in D \setminus \{d\}$ or else we could have a path between depots d and d' , hence we can define variables z_{ij}^d for any $d \in D$ only for arcs $(i, j) \in A^c \cup A^d$.

It is easy to see that the following relation between the “old” x variables and the “new” z variables holds:

$$z_{ij}^d \leq x_{ij} \quad \forall d \in D, \forall (i, j) \in A^c \cup A^d. \quad (13)$$

Consider the following constraints

$$\sum_{j \in C} z_{dj}^d = y_d \quad \forall d \in D \quad (14)$$

$$\sum_{j \in C} z_{jd}^d = y_d \quad \forall d \in D \quad (15)$$

$$\sum_{j \in d \cup C} z_{ji}^d = \sum_{j \in d \cup C} z_{ij}^d \quad \forall d \in D, \forall i \in C \quad (16)$$

$$z_{ij}^d \in \{0, 1\} \quad \forall d \in D, \forall (i, j) \in A^c \cup A^d, \quad (17)$$

and let $(3I)$ denote the equation system (13)–(17), (7) and (8). It is easy to see that (13)–(17) is a valid representation of (6). Furthermore, the constraints (17) can be relaxed to $z_{ij}^d \geq 0$ without violating the validity.

This follows because we can view $(3I)$ as a path model in a 3-layered graph where the set D is replicated into set \bar{D} , with each node $\bar{d} \in \bar{D}$ being the copy of the original node $d \in D$. With respect to the arc set defined in the new graph, the only difference is that arcs entering node d in the original graph now enter node \bar{d} . In this representation, a feasible circuit in the original graph corresponds to a path from a node d to its copy \bar{d} , and hence a feasible solution to the problem defined in the original graph corresponds to at most p paths (exactly p paths in the F-MDRP variant) from each node in D to its copy in \bar{D} in this 3-layered graph.

To the best of our knowledge, this 3-layered graph approach was first proposed by Albareda-Sambola et al. (2005), for a capacitated version of the L-MDRP. Later, Bektaş (2012) proposed a similar approach, albeit in a stronger form (see the next subsection), for a more general version of the the fixed variant of the problem. The whole system defined by $(3I)$ can be viewed as a flow system guaranteeing that y_d units of flow are sent from each node d to its copy, the z_{ij}^d variables can then be interpreted as indicating whether arc (i, j) is used to send flow from node d to its copy, and this path interpretation shows that the integrality requirement on the z variables can be relaxed.

Let $(3I)_L$ denote the LP relaxation of $(3I)$, more precisely, obtained from $(3I)$ by replacing constraints (17), (7) and (8) with:

$$z_{ij}^d \geq 0 \quad \forall d \in D, \forall (i, j) \in A^c \cup A^d \quad (18)$$

$$x_{ij} \geq 0 \quad \forall (i, j) \in A \quad (19)$$

$$y_d \geq 0 \quad \forall d \in D. \quad (20)$$

We can now relate the 1-MCC with the 3-index formulation as follows.

Proposition 2. $\mathcal{P}_{x,y}((3I)_L)$ is given by the 1-MCC inequalities (12), (19) and (20).

Proof. This follows from the max-flow/min-cut theorem. Given the interpretation of the $(3I)_L$ system in the 3-layered graph, we redefine variables z_{id}^d and x_{id} as $z_{i\bar{d}}^d$ and $x_{i\bar{d}}$, respectively, where \bar{d} is the copy of node $d \in D$. The max-flow/min-cut theorem indicates that for each depot d , y_d units of flow are sent from d to its copy \bar{d} if and only if every cut separating d from \bar{d} has capacity at least y_d . This last constraint is $x(d \cup (C \setminus S), \bar{d} \cup S) \geq y_d$ for $S \subseteq C$. Redefining the variables $x_{i\bar{d}}$ into x_{id} , we obtain the desired multi-cut inequalities (12). \square

2.2.3 A stronger 3-index formulation and generalization of the 1-MCC inequalities

The linear programming relaxation of the system $(3I)$ can be strengthened by replacing constraints (13) with the following,

$$\sum_{d \in D} z_{ij}^d \leq x_{ij} \quad \forall (i, j) \in A^c \quad (21)$$

$$z_{ij}^d \leq x_{ij} \quad \forall (i, j) \in A^d, \quad (22)$$

since any arc $(i, j) \in A^c$ can only be used in at most one circuit given the requirement that the circuits for each depot are disjoint. Let $(S3I)$ denote the equation system $(3I)$ with (21) and (22) replacing (13). Consider now the following generalization of the 1-MCC inequalities.

$$x(C \setminus S, D') + x(C \setminus S, S) + x(D', S) \geq y(D') \quad \forall D' \subseteq D, \forall S \subseteq C. \quad (23)$$

In the remainder of the paper, we denote the inequalities given above for sets D' , where $|D'| = k$, by k -MCC. Their validity follows from similar arguments given for the case $|D'| = 1$. We also note that any k -MCC associated to a depot subset D' is stronger than the sum of m inequalities k_1 -MCC, \dots , k_m -MCC associated, respectively, to depot subsets D_1, \dots, D_m , where $D_i \cap D_j = \emptyset$, $\forall i, j \in \{1, \dots, m\}$, $i \neq j$ and $\cup_{i=1, \dots, m} D_i = D'$. As will be shown next, the k -MCC inequalities can be derived from the stronger 3-index formulation providing an indirect proof of their validity.

Proposition 3. $\mathcal{P}_{x,y}((S3I)_L)$ is contained in the polyhedron defined by the MCC inequalities (23), (19) and (20).

Proof. The proof of this Proposition follows from the max-flow/min-cut theorem and also from the fact that we can view the $(S3I)$ system in the 3-layered graph. Variables z_{id}^d (x_{id}) are redefined as $z_{i\bar{d}}^d$ ($x_{i\bar{d}}$) as in the proof

of Proposition 2. For each subset $D' \subseteq D$, the max-flow/min-cut theorem says that $y(D')$ units of flow are sent from D' to the subset of the copies of the depots in D' , say \bar{D}' , if and only if every cut separating D' from \bar{D}' has capacity at least $y(D')$. This last constraint is $x(D' \cup C \setminus S, \bar{D}' \cup S) \geq y(D')$ for $S \subseteq C$. Redefining the variables $x_{i\bar{d}}$ into x_{id} , we obtain the desired generalized multi-cut inequalities (23). \square

It is still an open question as to whether we have a strict inclusion or an equality on the result of Proposition 3. The 3-index formulation can still be “strengthened” by observing that equality holds in both of the previous inequalities (21) and (22) since, respectively, if an arc $(i, j) \in A^c$ is used then it needs to be used in one and only one of the circuits, and if an arc with an endpoint in a depot $d \in D$ is used then it needs to be used in the circuit of the same depot, that is, we can further strengthen the 3-index formulation by replacing the inequalities (21) and (22) by:

$$\sum_{d \in D} z_{ij}^d = x_{ij} \quad \forall (i, j) \in A^c \quad (24)$$

$$z_{ij}^d = x_{ij} \quad \forall (i, j) \in A^d. \quad (25)$$

We conjecture that the LP bound does not improve by using equalities instead of inequalities. In fact, this is easy to prove for the case of constraints (25). To see this, consider by contradiction an inequality (22), assume that $d \in D$ and suppose also that there exists $i' \in C$ such that $z_{di'}^d < x_{di'}$. Then, we obtain $y_d = \sum_{i \in C} z_{di}^d < \sum_{i \in C} x_{di} = y_d$, where the first equality is given by constraints (14) and the second equality is given by constraints (1). Thus, constraints (22) and (25) are equivalent.

Equivalence is still open with respect to replacing (21) with (24). In some computational experiments made, the LP bounds obtained were the same with either sets of inequalities. The point of using the equality version of these constraints is that we can eliminate the x variables from the model, and by redefining the z_{ij}^d variables as 0-1 variables, we obtain a valid formulation for the problem involving only the z_{ij}^d (and y_d variables) as in the four previous works by Albareda-Sambola et al. (2005), Bektaş (2012), Fernández & Rodríguez-Pereira (2016) and Hill & Voß (2016).

2.2.4 Separation of the k -MCC inequalities

We now present a separation procedure for the k -MCC inequalities that is strongly associated with the previously given relations between the flow systems (3I) and (S3I) and the new inequalities. Let (x', y') be the current solution of the linear programming relaxation. Consider an MCC constraint associated with a subset of depots D' and let us build the corresponding 3-layered graph with an additional source node s and a target node t , new arcs connecting s to every node in D' and new arcs connecting the copies of the nodes in D' to t . All the new arcs have a flow capacity of 1, whereas all the remaining arcs (i, j) have a flow capacity that is equal to the value of x'_{ij} . The separation algorithm consists in checking whether a flow from the source s to the target t of value $y(D')$ exists in the extended 3-layered graph or not. If the maximum flow is less than $y(D')$, then we have found a violated MCC inequality which is identified by the corresponding minimum cut.

Observe that repeating this procedure for all subsets D' does not lead to a polynomial-time algorithm, although in practice it may not be as inefficient as one would expect, since typically MDRP instances involve few depots. Also, as we shall show in the next subsection, we need not separate all the MCC inequalities since more than half of them are redundant in the presence of the others. Finally, in such a procedure we can start by separating the constraints associated to the cases where $|D'| = 1$, which would provide a polynomial-time algorithm. Observe that, as pointed out before, the 1-MCC inequalities suffice in providing a valid representation of (6) and thus, the separation of k -MCC inequalities for $k > 1$ is completely optional. In the computational results we evaluate the potential usefulness of the more general MCC.

2.3 Properties of the multi-cut inequalities

This section presents two properties of the multi-cut inequalities, one related to the strength of the LP relaxation of the main formulation using these inequalities, and another on a feature of these constraints that implies a reduction in the number of inequalities that need to be separated. We start with the first result.

Proposition 4. *Consider an instance with symmetric costs and a solution (x^*, y^*) that satisfies the depot assignment constraints (1) and (2). Then, there exists a solution (x', y') that satisfies (1), (2) and all MCC inequalities (23), and has the same cost as (x^*, y^*) .*

Proof. Let (x', y') be defined as follows:

$$\begin{aligned} x'_{ij} &= \frac{1}{2}x_{ij}^* + \frac{1}{2}x_{ji}^* & \forall (i, j) \in A \\ y'_d &= y_d^* & \forall d \in D. \end{aligned}$$

Clearly, both (x^*, y^*) and (x', y') have the same objective function value since costs are symmetric (that is, $c_{ij} = c_{ji}$) and we can easily see that (x', y') satisfies (1) and (2). Consider a subset of client nodes $S \subseteq C$ and a subset of depots $D' \subseteq D$. Let $S' = C \setminus S$. We have the following:

$$\begin{aligned} x'(S', D') + x'(S', S) + x'(D', S) &= \frac{1}{2} (x^*(S', D') + x^*(D', S') + x^*(S', S) + x^*(S, S') + x^*(D', S) + x^*(S, D')) \\ &= \frac{1}{2} (x^*(C, D') + x^*(D', C) + x^*(S', S) + x^*(S, S')) \\ &= \frac{1}{2} (y(D') + y(D') + x(S', S) + x(S, S')) \\ &\geq y(D'). \end{aligned}$$

Note that $x(D', C) = y(D') = x(C, D')$ results from adding up inequalities (1) and (2) for every depot $d \in D'$, and that both $x(S', S)$ and $x(S, S')$ are non-negative. Thus, we have proved that any MCC inequality (23) is satisfied by the new solution (x', y') . \square

The implication of Proposition 4 is that, for symmetric instances, $v(F_L^-)$ remains the same irrespective of whether F_L^- includes the multi-cut inequalities or not. We note, however, that even in this case the MCC are needed to provide a valid integer programming formulation for the problem, at least in what concerns the simpler set given by the 1-MCC inequalities. We also note that this result only holds for symmetric instances. As will be evidenced by the computational results, the multi-cut constraints may improve the value of F_L^- in the case of asymmetric instances.

Our second result concerns the redundancy of a subset of k -MCCs, which has implications on the reduction of the computational effort needed to separate these inequalities. We start by presenting an equality that holds for F_L^- and is relevant for the proof of Proposition 5.

Lemma. *The assignment constraints (1)–(4) imply the following equality:*

$$\begin{aligned} x(C \setminus S, D_1) + x(C \setminus S, S) + x(D_1, S) + y(D \setminus D_1) &= y(D_1) + x(S, D \setminus D_1) + x(S, C \setminus S) + x(D \setminus D_1, C \setminus S), \\ \forall S \subseteq C, \forall D_1 \subseteq D. \end{aligned} \tag{26}$$

Proof. Let $S' = C \setminus S$ and $D_2 = D \setminus D_1$. The proof is based on four valid equalities that are obtained by adding the assignment constraints (1)–(4) for adequate subsets and then combining the resulting equalities.

- i) by adding the assignment constraints (1) for $d \in D_1$ and partitioning C into S and S' , we obtain $x(D_1, S) + x(D_1, S') = y(D_1)$;
- ii) similarly, by adding constraints (2) for $d \in D_2$ we obtain $x(S', D_2) + x(S, D_2) = y(D_2)$;
- iii) by adding constraints (3) for $i \in S'$ and partitioning V into D_1 , D_2 , S and S' , we obtain $x(S', D_1) + x(S', D_2) + x(S', S) + x(S') = |S'|$;
- iv) similarly, by adding constraints (4) for $i \in S'$ we obtain $x(D_1, S') + x(D_2, S') + x(S, S') + x(S') = |S'|$.

By combining the two last equalities we obtain $x(S', D_1) + x(S', D_2) + x(S', S) = x(D_1, S') + x(D_2, S') + x(S, S')$. In this last equality we now use the first two equalities obtained under i) and ii) and replace $x(S', D_2)$ by $y(D_2) - x(S', D_1)$ and $x(D_1, S')$ by $y(D_1) - x(S, D_2)$ to obtain (26). \square

Proposition 5. *In the presence of the assignment constraints (1)–(4), every $|D|$ -MCC inequality is redundant and for each subset $S \subseteq C$ and each subset $D_1 \subset D$, with $|D_1| = k$, the corresponding k -MCC inequality is equivalent to the $(|D| - k)$ -MCC inequality written for $S' = C \setminus S$ and $D_2 = D \setminus D_1$.*

Proof. The first statement is a direct consequence of the equality (26) written for $D_1 = D$ and the fact that $x(S, S') \geq 0$. From equality (26) we also obtain that $\forall S \subseteq C, \forall D_1 \subset D$,

$$x(C \setminus S, D_1) + x(C \setminus S, S) + x(D_1, S) \geq y(D_1)$$

if and only if

$$x(S, D \setminus D_1) + x(S, C \setminus S) + x(D \setminus D_1, C \setminus S) \geq y(D \setminus D_1).$$

□

The previous result indicates that, in practice, it suffices to separate only up to about half of the k -MCCs, as those that are not separated will be implied by the ones that are. For example, for an instance with six depots, we only need to search for violated 1-MCCs, 2-MCCs and half of the 3-MCCs. This reduction does not yield a polynomial-time algorithm for separating all MCCs. However, and as pointed out before, in a branch-and-cut procedure we can start by separating the 1-MCCs, which would provide a polynomial-time algorithm, and extend the search to k -MCCs with $k > 1$ if we so decide. Finally, due to Proposition 4, this result is only interesting for asymmetric instances.

2.4 Comparison of the MCC inequalities with existing inequalities

In this section, we investigate how the MCCs relate with similar sets of inequalities described earlier in the literature. To the best of our knowledge, this work is the first to describe path elimination constraints on a directed graph, using variables defined for arcs instead of edges. Previous studies describe formulations on undirected graphs using edge variables only, in which the use of binary edge variables automatically excludes two-node cycles (i.e., cycles with a depot and a client) unless suitable modifications are performed. In particular, Benavent & Martínez-Sykora (2013) use a variable u_{ij} , defined as binary for every pair $i, j \in C$ to indicate whether the edge $\{i, j\}$ between clients i and j is used in the solution, and defined as $\{0, 1, 2\}$ for every pair $d \in D, i \in C$ indicating, respectively, whether the edge linking depot d and client i is not used or if it is used once or twice, with the latter case forming a two-node cycle. The edge variables in Benavent & Martínez-Sykora (2013) can be related with the directed variables x_{ij} through the equalities $u_{ij} = x_{ij} + x_{ji}$ for every edge $\{i, j\}$. Observe that the directed model satisfies the inequalities $x_{ij} + x_{ji} \leq 1$ for $i, j \in C$, which are a subset of the subtour elimination constraints (10), and thus the inequality $u_{ij} \leq 1$ for $i, j \in C$ is satisfied. This relation allows us to write the path elimination constraints described by Benavent & Martínez-Sykora (2013) using directed variables and therefore allows a comparison with the new ones proposed in this work.

To establish the connection, we start by observing that by using the assignment constraints (2) and (4), the MCC inequalities (23) can be rewritten in packing form as follows:

$$x(D \setminus D', S) + x(S) + x(S, D') \leq |S| \quad \forall D' \subseteq D, \forall S \subseteq C. \quad (27)$$

As previously pointed out, some of the constraints (27) are redundant. For simplicity, however, we assume the entire set of MCC inequalities in the following exposition. Consider, now, the following weaker version of (27)

$$x(D \setminus D', i) + x(S) + x(j, D') \leq |S| \quad \forall D' \subseteq D, \forall i, j \in S \subseteq C, i \neq j, \quad (28)$$

and add up the inequality (28) written for sets D' and S with the same inequality written for sets $D \setminus D'$ and the same set S , where i and j are swapped. This operation results in the following inequality:

$$x(D \setminus D', i) + x(i, D \setminus D') + 2x(S) + x(j, D') + x(D', j) \leq 2|S| \quad \forall D' \subseteq D, \forall i, j \in S \subseteq C, i \neq j. \quad (29)$$

Clearly this constraint is redundant since it is obtained by adding two valid inequalities. In fact, we can even observe that from an integer point of view, it does not even eliminate circuits with more than one depot. However,

if $|S| \geq 3$ we can reduce the right-hand side of constraint (29) to $2|S| - 1$ and obtain the following valid inequality for the problem:

$$x(D \setminus D', i) + x(i, D \setminus D') + 2x(S) + x(j, D') + x(D', j) \leq 2|S| - 1 \quad \forall D' \subseteq D, \forall i, j \in S \subseteq C : |S| \geq 3, i \neq j. \quad (30)$$

The inequalities above, under the relation between the undirected variables u and the directed variables x , correspond to the directed version of the path elimination constraints used by Benavent & Martínez-Sykora (2013) and the proof of their validity follows from arguments similar to the ones used for the undirected case, as in the previously cited paper.

As observed above, the inequalities (30) are not valid when $S = \{i, j\}$ and thus circuits of the type $(d_1, i, j, d_2, p, q, d_1)$, in which d_1 and d_2 are depots and i, j, p and q are clients, are not guaranteed to be cut off. To prohibit such solutions, we consider the following inequalities, that are also the directed version of similar inequalities for the undirected case described in Benavent & Martínez-Sykora (2013):

$$x(D \setminus D', i) + x(i, D \setminus D') + 3x_{ij} + 3x_{ji} + x(j, D') + x(D', j) \leq 4 \quad \forall D' \subseteq D, \forall i, j \in C, i \neq j. \quad (31)$$

One observation concerns particular circuits of the type $\bar{\sigma} = (d_1, i_1, d_2, i_2, \dots, d_{m-1}, i_{m-1}, d_m, i_m, d_1)$, that is, infeasible circuits which alternate between a depot and a client. As has been shown in Laporte et al. (1986), circuits of the type $\bar{\sigma}$ are never optimal if we consider symmetric costs, since there always exists a feasible solution that is cheaper or at most as costly. Here, we provide a proof for the sake of completeness. Consider a solution σ_1 that consists of m single-client return trips as (d_j, i_j, d_j) for $j \in \{1, \dots, m\}$. Consider now another solution σ_2 , also consisting of m return trips, but now in the form (d_j, i_{j-1}, d_j) for $j \in \{2, \dots, m\}$ and (d_1, i_m, d_1) . We claim that the cheapest of these two feasible solutions σ_1 and σ_2 is cheaper than the alternating circuit $\bar{\sigma}$. Let \bar{c} be the cost of circuit $\bar{\sigma}$, and c_1, c_2 be the costs of solutions σ_1 and σ_2 , respectively. Since costs are symmetric, then $2\bar{c} = c_1 + c_2$. Suppose, without loss of generality, that $c_1 \leq c_2$. Then $2\bar{c} \geq 2c_1$, hence $\bar{c} \geq c_1$.

Infeasible depot-client alternating circuits are always eliminated by the 1-MCC inequalities (12) for single node sets S . Whilst eliminating circuits of this type is redundant for symmetric instances, the 1-MCCs are needed in the case of asymmetric instances, since solutions containing circuits of this type are not necessarily dominated by the two alternative solutions with m return trips mentioned above.

The reason why we have included the constraints (30) and the version for two client sets (31) in this paper is that they also provide a valid representation of (6) in the generic model but only in the case of symmetric cost instances, since the infeasible depot-client alternating circuits are not eliminated by (30) nor (31). In addition, for symmetric instances, and in contrast to what happens with the MCC inequalities (23) as shown by Proposition 4, these inequalities may improve the LP bound of the $F^=$ model as our computational results shall show. In order to solve asymmetric cost instances, inequalities (30) and (31) do not suffice, so an additional set of inequalities (e.g, the 1-MCC inequalities (12) for single node sets S , as pointed out before) will need to be considered.

As a final conclusion to this section, we note that another set of undirected path elimination constraints is presented by Belenguer et al. (2011), however they are based on a different set of binary variables to model two-node cycles. For this reason, a direct comparison between their inequalities and inequalities (30) and (31) is not straightforward and requires further investigation.

2.5 Variations

In this section, we look at two variations of the generic model, one where the depot assignment constraints are relaxed, and the other to cater for multiple vehicles at each depot.

2.5.1 Analyzing the impact of the depot assignment constraints

This section is motivated by the formulations described in the two papers by Belenguer et al. (2011) and Benavent & Martínez-Sykora (2013) where no information is given on the value of the degree of the depot nodes. We first observe that solutions composed by chains such as $(d_1, i_1, \dots, i_m, d_2)$, where d_1 and d_2 are two different depots and i_1, \dots, i_m are clients, are feasible for the models proposed by Belenguer et al. (2011) and Benavent & Martínez-Sykora (2013) without the path elimination constraints. However, such solutions are not feasible for the generic model proposed in Section 2 without the path elimination constraints, since the assignment constraints (1) and (2) require that, for each depot, the in-degree and the out-degree must be equal to one another.

We observe that when $S = C$ and $S = \emptyset$, the 1-MCC inequalities (12) are equivalent, respectively, to constraints:

$$\sum_{j \in C} x_{dj} \geq y_d \quad \forall d \in D \quad (32)$$

$$\sum_{j \in C} x_{jd} \geq y_d \quad \forall d \in D. \quad (33)$$

Therefore, when the 1-MCCs are used in model $F^=$, it is possible to replace constraints (1) and (2) by the following weaker set of constraints

$$\sum_{j \in C} x_{dj} \leq y_d \quad \forall d \in D \quad (34)$$

$$\sum_{j \in C} x_{jd} \leq y_d \quad \forall d \in D, \quad (35)$$

and still obtain a valid model for the MDRP. In particular, we will refer to the model $F^=$ where (1) and (2) are replaced by (34) and (35) as F^{\leq} .

The main observation for the formulation F^{\leq} is that its LP relaxation without the 1-MCC inequalities (12) admits paths that connect depots but that are not necessarily part of circuits, and the corresponding LP relaxation bound may be weaker when compared with the LP relaxation bound given by model $F^=$. This may lead to a difference in the performance of each model within the branch-and-cut algorithm described in Section 3.

2.5.2 Extensions to the multi-vehicle case

In this section we show how to extend the models to the cases where some depots may have more than one vehicle available to use, that is, $k_d > 1$ for some $d \in D$.

For the F-MDRP variant, bearing in mind that at least one vehicle from each depot is needed, the decisions in the multi-vehicle case would entail choosing between 1 and k_d vehicles to use at each depot. To this end, the binary y_d variable would be changed to be a general integer, to indicate the number of vehicles used at depot $d \in D$. The domain constraints for the y variables would then be changed as $y_d \in \{1, \dots, k_d\}$, $\forall d \in D$. As for the L-MDRP variant, the new integrality requirements would be as $y_d \in \{0, 1, \dots, k_d\}$, $\forall d \in D$, as one would use any number of vehicles between 0 and k_d . In addition, for both variants, the depot assignment constraints (1) and (2) are written in the same way but, owing to the new definition of the y variables, they now state that the out-degree and the in-degree of each depot should be equal to the number of vehicles used from that depot.

For both variants, the 1-MCC inequalities (12) would be written in the same way and are still valid considering the new definition of the y variables. Also, generalizations of the 1-MCC inequalities involving subsets of depots, that is, the k -MCC inequalities (23), can also be adapted in a straightforward manner.

The relation between the 1-MCC inequalities (respectively, the k -MCC inequalities) and the 3-index formulation (respectively, the stronger 3-index formulation) is also valid in the multi-vehicle per depot case, with the difference that we can now have between 0 and k_d paths (1 and k_d paths in the F-MDRP variant) from each depot $d \in D$ to its replica $\bar{d} \in \bar{D}$. Given the interpretation of the $(3I)_L$ (respectively, the $(S3I)_L$) system in the 3-layered graph, we can once more use the max-flow/min-cut theorem to prove a result similar to Proposition 2 (respectively, Proposition 3). Additionally, given the new definition of the y variables, both Proposition 4 and Proposition 5 are also valid, since their proofs depend only on the assignment constraints, which are unchanged in terms of their mathematical expression, and not on the fact that the y variables were defined as 0-1 variables.

We omit the formal proofs in the multi-vehicle per depot case since our objective in this section is merely to state that the new proposed path elimination constraints are still applicable in more general multi-depot routing problems which may involve other sets of constraints (e.g., capacities) that may require the use of more than one vehicle per depot.

3 Computational Results

This section describes computational experiments conducted to address the four aims listed below:

- i) To assess the performance of a branch-and-cut algorithm that uses the 1-MCC inequalities (12). We also analyze the effect of imposing the 1-MCC inequalities (12) on the integer solutions.
- ii) To compare the performance of $F^=$ with F^{\leq} within the branch-and-cut algorithm.
- iii) To evaluate the effect of introducing the k -MCC inequalities (23) for $k > 1$ on the LP relaxation bound for asymmetric instances.
- iv) To numerically compare the quality of the 1-MCC inequalities (12) with the quality of the inequalities (30) and (31), in terms of LP relaxation bounds, for symmetric as well as asymmetric instances.

3.1 Instances

Three sets, A, B and C, of instances are used in the experiments. The first set A is a subset of the symmetric benchmark location-routing problem instances available at <http://prodhonc.free.fr/>, also used by Benavent & Martínez-Sykora (2013), comprising a total of nine instances, six with 100 clients and with either 5 or 10 depots, and three with 200 clients and 10 depots. In these instances, we are given coordinates for each node and, following Benavent & Martínez-Sykora (2013), the costs c_{ij} and c_{ji} are determined as the Euclidean distance between nodes i and j multiplied by 100 and rounded up to the nearest integer. These instances are named as $|C|-|D|-t$, where $t = 1, 2, 3$ indicates how the points are distributed, where 1 represents uniform distribution, and 2 and 3 denote two and three clusters, respectively.

The instances in set B are also symmetric, where the node coordinates have been generated by uniformly placing as many points as needed in a 200×200 continuous grid. The set includes a total of 17 instances, comprising nine with 100 clients, three with 200 clients and a five with 300 clients. For the first subset, we initially generate an instance with 100 clients and 20 depots. Removing 10 depots from this instance gives rise to another instance with 10 depots. Finally, a further removal of five depots results in an instance with five depots. The construction is used to obtain a geographically interrelated set of instances with the same set of clients, and where the depot locations have a nested structure. By repeating this process twice more, we obtain a total of nine 100-client instances. The same process has been used to generate three instances with 200 clients and either 10, 20 or 40 depots, and five instances with 300 clients and either 10, 20, 30, 40 or 60 depots. The instances with 100 clients are named as $\text{bgs-100-}|D|-t$, where $t = 1, 2, 3$ denotes the instance number, whereas the instances with 200 or 300 clients are named as $\text{bgs-}|C|-|D|$.

The instances in set C are asymmetric and are obtained by transforming the symmetric instances in set B. The transformation consists in increasing or decreasing the cost of an arc (i, j) by a percentage $p_{ij} \in [0.25, 0.75]$. More precisely, for each edge $\{i, j\}$, we randomly choose either arc (i, j) or arc (j, i) with equal probability. The cost of the chosen arc is increased, and the cost of the arc in the opposite direction is decreased by the specified percentage. The instances in set C are named as the original symmetric instance from which they derive with an added suffix “a”.

3.2 Description of the branch-and-cut algorithm

The branch-and-cut (B&C) algorithm operates on the generic formulation described in Section 2, within which two sets of cuts, namely the subtour elimination constraints (11) and the 1-MCC inequalities (12), are separated.

In the B&C algorithm, in each node of the branch-and-bound tree, we first separate the subtour elimination constraints (11) by using a standard heuristic procedure (see, e.g., Fischetti et al. 1997), which consists in finding client-only connected components induced by the fractional solution x^* by considering only the arcs $(i, j) \in A$ such that $x_{ij}^* > 0$. If no heuristic subtour elimination constraints are found then we proceed by applying exact separation algorithms. Both the subtour elimination and the 1-MCC constraints are separated exactly by using standard algorithms (see, e.g., Ahuja et al. 1993) for the former and the procedure described in Section 2.2.4 for the latter. A maximum of 20 inequalities of both types are added before re-optimizing the LP, including subtour elimination constraints found with the heuristic procedure, but, if more than five violated inequalities (11) are found, then the separation of 1-MCCs is skipped and the re-optimization phase begins immediately.

In the computational experiments related with evaluating the k -MCC inequalities for $k > 1$, the k -MCCs are separated in an ascending order of k but, as in the standard branch-and-cut, the algorithm does not attempt to separate a k -MCC if more than five violated $(k - 1)$ -MCCs are found. The separation procedure used for separating these inequalities was described in Section 2.2.4.

In the computational experiments where inequalities (30) and (31) are used, the branch-and-cut functions in a similar way but uses the separation algorithm described by Belenguer et al. (2011) and adapted for arc variables, which will be discussed further in Section 3.5.

The B&C algorithm was ran on an Intel Core i7-4790 3.6GHz processor with 8GB of RAM, within which CPLEX 12.6.1 Concert Technology for C++ was used. A time limit of 10800 seconds (three hours) has been imposed on the solution time of each instance.

3.3 Evaluating the effect of the 1-MCC inequalities (12)

We now present computational results to assess the performance of the B&C, for both the F-MDRP and L-MDRP, in solving the symmetric and asymmetric instances to optimality. In addition, we compare the integer solutions obtained with and without 1-MCC inequalities, in order to study the impact of these inequalities on the feasibility of the resulting solutions. Finally, we compare the performance of the B&C for both $F^=$ and F^{\leq} so as to analyze the effect of imposing additional information on the degree constraints of the depots, as was discussed in Section 2.5.1. Due to the result given in Proposition 4 we divide this study in two subsections, one concerning the symmetric instances and the other for the asymmetric instances.

3.3.1 Symmetric instances

The results for the sets A and B of instances are given in Table 1 for the F-MDRP and Table 2 for the L-MDRP. Each table presents the results separately for formulations $F^=$ and F^{\leq} . The columns are divided into two groups, one for the B&C without 1-MCCs, indicating that only the subtour elimination constraints (11) are separated, and the other where the 1-MCCs were included in the B&C. For each instance, the columns LP and t_L show the linear programming relaxation bound of the corresponding formulation and the associated computation time, and the columns OPT and t show the optimal value of the problem and the associated computational time. When the instance was not solved within the 10800 second time limit, this is shown by a “*”, in which case the OPT column shows the best lower and upper bounds obtained. The column named ‘Feas?’ indicates whether the solution identified without the 1-MCCs is feasible with respect to the path elimination constraints. Finally, columns #CC and #MC show the total number of subtour elimination constraints and 1-MCCs separated in the B&C tree, respectively. The additional column DU in Table 2 shows the number of depots used in the optimal solution identified by the algorithm.

	B&C without 1-MCCs							B&C with 1-MCCs							
Name	LP	t_L	#CC	OPT	t	#CC	Feas?	LP	t_L	#CC	#MC	OPT	t	#CC	#MC
Using model $F^=$															
100-5-1	37506	0	191	38116	25	957	yes	37506	0	191	0	38116	46	1425	32
100-5-2	33679.5	0	167	33976	14	779	no	33679.5	0	167	0	34018	30	911	39
100-5-3	32622.5	0	139	32979	4	289	no	32622.5	0	139	0	33024	10	336	8
100-10-1	40847.5	0	159	41119	3	258	no	40847.5	0	159	7	41991	39	774	403
100-10-2	38529	1	77	38868	2	125	no	38529	0	77	0	39126	9	492	91
100-10-3	33015	0	241	33364	3	368	no	33015	1	241	0	33719	18	522	40
200-10-1	52834.5	5	292	53620	1981	2730	no	52834.5	1	292	2	53739	8283	3984	1489
200-10-2	46758.5	1	252	47354	2059	2479	no	46758.5	2	252	0	47441	5639	5472	506
200-10-3	47381.5	2	317	47669	191	1669	no	47381.5	2	317	4	47828	2414	4606	377
bgs-100-5-1	1578.5	0	196	1587	1	248	yes	1578.5	0	196	2	1587	1	248	0
bgs-100-10-1	1576	1	239	1582	1	281	no	1576	1	239	2	1597	8	365	187
bgs-100-20-1	1656.5	0	126	1669	4	283	no	1656.5	1	131	4	1687	45	508	679
bgs-100-5-2	1728.5	1	179	1744	9	416	yes	1728.5	0	179	2	1744	7	420	35
bgs-100-10-2	1754	0	207	1767	3	197	no	1754	0	207	2	1777	53	436	386
bgs-100-20-2	1779.5	0	99	1805	3	137	no	1779.5	0	99	10	1835	133	378	1767
bgs-100-5-3	1517	1	156	1553	45	1187	yes	1517	1	156	0	1553	43	1131	11
bgs-100-10-3	1513	0	123	1555	68	801	no	1513	0	123	0	1558	107	866	163
bgs-100-20-3	1563	1	94	1599	2	152	no	1563	0	94	4	1604	3	180	28
bgs-200-10	2290.75	4	432	2294	10	378	no	2290.75	4	415	7	2301	704	1395	568
bgs-200-20	2300.33	2	247	2311	35	519	no	2300.33	2	247	0	[2327, 2333]	10800*	2347	6365
bgs-200-40	2375.5	3	214	2397	83	373	no	2375.5	3	214	14	2417	7838	944	11464
bgs-300-10	2698.5	7	347	[2705, 2742]	10800*	8178	no	2698.5	7	350	2	[2708, 2767]	10801*	9200	1555
bgs-300-20	2715	5	294	[2722, 2773]	10800*	6919	no	2715	5	294	0	[2728, 2787]	10801*	6009	4493
bgs-300-30	2753	8	447	2773	8994	2304	no	2753	9	447	4	[2767, 2826]	10801*	3176	7863
bgs-300-40	2793	8	356	[2809, 2815]	10800*	1858	no	2793	11	356	4	[2808, 2861]	10802*	2000	11951
bgs-300-60	2882	10	298	2904	3358	962	no	2882	18	273	21	[2906, 2994]	10801*	1355	14863
Using model F^{\leq}															
100-5-1	37190.5	0	207	37319	3	294	no	37506	1	265	9	38116	25	931	43
100-5-2	30456.5	0	158	30618	60	1556	no	33679.5	1	202	9	34018	31	937	52
100-5-3	31922	0	190	32087	2	235	no	32622.5	1	210	11	33024	3	398	14
100-10-1	39051.5	0	168	39142	3	384	no	40847.5	0	200	23	41991	254	795	934
100-10-2	30121	0	196	30392	22	706	no	38529	1	309	15	39126	18	678	117
100-10-3	31775	0	279	31775	1	199	no	33015	2	283	17	33719	36	673	88
200-10-1	52291.5	1	260	52690	3031	3331	no	52834.5	2	260	18	[53617, 53739]	10801*	3721	2178
200-10-2	38963.8	2	397	39137	360	2420	no	46758.5	4	531	16	[47314, 47441]	10800*	4092	444
200-10-3	44535.5	2	409	44983	7114	5605	no	47381.5	4	606	18	47828	1303	2702	158
bgs-100-5-1	1542	0	283	1552	5	433	no	1578.5	0	315	11	1587	2	288	8
bgs-100-10-1	1499.5	1	278	1501	3	482	no	1576	1	371	18	1597	27	632	400
bgs-100-20-1	1484.5	0	165	1486	1	150	no	1656.5	0	253	36	1687	26	551	459
bgs-100-5-2	1696.5	0	223	1698	1	248	no	1728.5	1	223	9	1744	21	814	55
bgs-100-10-2	1687.5	1	259	1692	1	220	no	1754	1	274	15	1777	33	462	291
bgs-100-20-2	1653.5	1	189	1657	1	188	no	1779.5	1	196	39	1835	74	403	1675
bgs-100-5-3	1508.5	0	135	1516	2	241	no	1517	0	140	7	1553	25	1369	10
bgs-100-10-3	1471	0	87	1471	0	0	no	1513	0	107	22	1558	41	806	146
bgs-100-20-3	1433.5	0	148	1438	1	123	no	1563	0	167	35	1604	5	193	139
bgs-200-10	2246.5	2	486	2259	3084	5223	no	2290.75	3	540	16	2301	620	1974	732
bgs-200-20	2224.5	4	499	2232	74	827	no	2300.33	4	541	30	2332	9320	2235	7424
bgs-200-40	2163.5	2	317	2168	53	548	no	2375.5	4	438	95	[2409, 2417]	10801*	1435	24137

*Not solved to optimality within the limit of three hours.

Table 1: Evaluating the branch-and-cut for symmetric instances (F-MDRP)

	B&C without 1-MCCs							B&C with 1-MCCs								
Name	LP	t_L	#CC	OPT	t	#CC	Feas?	LP	t_L	#CC	#MC	OPT	t	DU	#CC	#MC
Using model $F^=$																
100-5-1	37190.5	0	207	37943	22	943	yes	37190.5	1	207	0	37943	22	4	943	0
100-5-2	30456.5	0	137	30832	6	374	yes	30456.5	0	137	0	30832	6	2	374	0
100-5-3	31922	0	190	32850	20	1346	yes	31922	0	190	0	32850	24	4	1346	0
100-10-1	39051.5	1	268	39608	10	730	yes	39051.5	0	268	0	39608	20	5	820	1
100-10-2	30121	0	189	30641	25	921	yes	30121	0	189	0	30641	30	2	1135	4
100-10-3	31775	0	281	32521	3	420	yes	31775	0	281	0	32521	4	5	420	0
200-10-1	52291.5	2	334	53047	2182	4172	yes	52291.5	2	334	0	53047	1253	5	4672	9
200-10-2	38963.8	2	386	39663	6574	8858	yes	38963.8	3	386	0	39663	5221	4	7856	2
200-10-3	44535.5	4	456	45289	4452	5541	yes	44535.5	3	456	0	45289	4078	3	4893	1
bgs-100-5-1	1542	0	315	1569	148	2379	yes	1542	1	315	0	1569	117	2	1798	10
bgs-100-10-1	1499.5	0	273	1524	19	934	yes	1499.5	0	273	0	1524	20	6	934	2
bgs-100-20-1	1484.5	0	264	1509	6	599	yes	1484.5	0	264	0	1509	5	9	516	1
bgs-100-5-2	1696.5	1	328	1705	1	283	yes	1696.5	0	328	0	1705	1	1	283	0
bgs-100-10-2	1687.5	1	293	1702	6	310	yes	1687.5	0	293	0	1702	2	2	310	0
bgs-100-20-2	1653.5	0	216	1695	13	1059	yes	1653.5	1	216	0	1695	9	5	605	5
bgs-100-5-3	1508.5	0	133	1553	56	2672	yes	1508.5	0	133	0	1553	58	5	2672	0
bgs-100-10-3	1471	0	125	1530	121	3293	yes	1471	0	125	0	1530	124	7	3293	0
bgs-100-20-3	1433.5	0	165	1494	18	745	yes	1433.5	0	165	0	1494	14	12	609	2
bgs-200-10	2246.5	3	358	2266	4928	9700	yes	2246.5	2	358	0	2266	5997	3	10502	42
bgs-200-20	2224.5	2	419	2250	1536	5074	yes	2224.5	3	419	0	2250	1110	7	4272	8
bgs-200-40	2163.5	1	249	2211	8284	10941	yes	2163.5	1	249	0	2211	8134	16	10805	55
bgs-300-10	2663.5	21	638	[2678, 2823]	10800*	12111	yes	2663.5	22	638	0	[2675, 2826]	10800*	3	11074	6
bgs-300-20	2645	18	549	[2665, 2807]	10801*	9810	yes	2645	17	549	0	[2666, 2870]	10802*	8	9604	33
bgs-300-30	2631	13	632	[2664, 2713]	10801*	9874	no	2631	12	632	0	[2664, 2717]	10802*	12	10137	48
bgs-300-40	2622.5	9	417	[2662, 2693]	10801*	11704	yes	2622.5	9	417	0	[2662, 2693]	10802*	11	10036	84
bgs-300-60	2610.5	9	394	[2653, 2682]	10934*	6138	no	2610.5	9	394	0	[2650, 2729]	10802*	16	5834	89
Using model F^{\leq}																
100-5-1	37190.5	1	207	37319	23	692	no	37190.5	1	207	6	37943	15	4	1026	7
100-5-2	30456.5	0	159	30618	48	1165	no	30456.5	0	164	6	30832	11	2	740	6
100-5-3	31922	1	189	32087	2	124	no	31922	0	189	8	32850	49	4	1797	14
100-10-1	39051.5	1	168	39142	3	397	no	39051.5	0	168	14	39608	37	5	910	19
100-10-2	30121	1	218	30392	25	775	no	30121	0	218	6	30641	54	2	1177	11
100-10-3	31775	1	277	31775	2	200	no	31775	1	277	9	32521	24	5	718	18
200-10-1	52291.5	2	257	52690	1304	1397	no	52291.5	3	257	18	[52831, 53086]	10800*	6	13296	124
200-10-2	38963.8	1	412	39137	81	836	no	38963.8	3	412	5	39663	8580	4	10104	24
200-10-3	44535.5	2	411	44983	1113	4037	no	44535.5	3	411	5	45289	2932	3	4671	22
bgs-100-5-1	1542	1	315	1552	3	442	no	1542	1	292	4	1569	72	2	1416	11
bgs-100-10-1	1499.5	0	275	1501	2	224	no	1499.5	0	261	10	1524	20	6	1102	17
bgs-100-20-1	1484.5	0	217	1486	1	202	no	1484.5	0	215	16	1509	11	9	968	34
bgs-100-5-2	1696.5	0	217	1698	1	292	no	1696.5	1	218	5	1705	3	1	377	5
bgs-100-10-2	1687.5	0	268	1692	2	227	no	1687.5	0	258	10	1702	3	2	380	14
bgs-100-20-2	1653.5	0	241	1657	1	236	no	1653.5	0	241	25	1695	84	5	1651	41
bgs-100-5-3	1508.5	0	144	1516	1	184	no	1508.5	0	144	6	1553	34	5	1217	11
bgs-100-10-3	1471	0	90	1471	1	89	no	1471	1	83	13	1530	116	7	3644	19
bgs-100-20-3	1433.5	1	166	1438	0	127	no	1433.5	0	166	23	1494	13	12	649	36
bgs-200-10	2246.5	2	447	2259	450	2209	no	2246.5	3	447	7	2266	6735	3	12480	42
bgs-200-20	2224.5	3	341	2232	80	710	no	2224.5	4	356	15	2250	5452	7	9648	99
bgs-200-40	2163.5	1	248	2168	40	525	no	2163.5	5	278	38	[2192, 2219]	10801*	15	7838	167

*Not solved to optimality within the limit of three hours.

Table 2: Evaluating the branch-and-cut for symmetric instances (L-MDRP)

There are three main sets of implications of the results for symmetric instances:

- We initially restrict our attention to the results obtained with model $F^=$, reported in the upper half of Tables 1 and 2. A comparison of the LP bounds of model $F^=$ with and without 1-MCCs supports the theoretical result of Proposition 4, both for the F-MDRP and the L-MDRP. Additionally, when compared purely in terms of value, the optimal solutions identified with 1-MCC inequalities (12) are not very different from those obtained without the use of the 1-MCC inequalities. However, as the column ‘Feas?’ indicates, most of the former sets of solutions are not feasible for the F-MDRP. This is rather obvious given that path elimination constraints are an integral part of the model to ensure feasibility. The situation, however, is

rather different for the L-MDRP, where every solution obtained without the 1-MCCs is feasible, excluding two of the instances with 300 clients (although these remain unsolved, hence we cannot be sure that the optimal solution will not be also feasible). This can be explained by the fact that the model will not choose to use all available depots, and only select those that are sufficiently distant from one another, reducing the likelihood that circuits containing two depots or more will be formed. Nevertheless, separating the 1-MCC inequalities can be seen to improve the solution time to optimality even when the instances are feasible without the use of these inequalities. This is evident from the results shown for instances 200-10-1, 200-10-2, 200-10-3, bgs-200-20 and bgs-200-40, for which the solution time to optimality decreased by up to about 30% with the use of the 1-MCC inequalities. Finally, as for the L-MDRP instances, we observe the expected behavior of reduction of the cost when more depots are allowed.

- We now analyze the results obtained with model F^{\leq} reported in the lower half of the tables. In this case, we find that all the solutions obtained without the use of the 1-MCC inequalities yield path-infeasible solutions. In addition, the B&C usually identifies additional 1-MCCs in comparison to model $F^=$, particularly for instances with 200 clients. The discussion provided earlier in Section 2.5.1 explains the reasons behind these findings, in particular that the relaxed depot assignment constraints allow infeasible paths that are not contained in circuits, and that feasible (and optimal) solutions cannot be obtained without enforcing the 1-MCCs. The conclusion is that providing more information on the degree of the depots prevents the formation of many path-infeasible solutions, and avoids the use of many (or in some cases all) path elimination constraints that would otherwise be needed. However, the degree information itself cannot guarantee that optimal solutions obtained without path elimination constraints would be path-feasible, as will be seen in the case of asymmetric instances.
- When models $F^=$ and F^{\leq} are compared in terms of the strength of the LP relaxation, it can be seen that, in the case of the F-MDRP, the LP bounds provided by the former are substantially better when 1-MCCs are not separated. This is also an indication that model $F^=$ permits a significantly less number of infeasible solutions, particularly as the LP value of model $F^=$ without 1-MCC inequalities is generally larger than the optimal value of model F^{\leq} without 1-MCC inequalities. This, for example, is evidenced across all instances in the case of the F-MDRP. Nevertheless, these observations do not necessarily suggest that the B&C based on model F^{\leq} should not be considered. In fact, the B&C based on model F^{\leq} slightly outperformed the B&C based on model $F^=$ in some cases (e.g., instance 200-10-3). However, the results suggest that, on average, the B&C based on model $F^=$ seems to be the best option for solving both the F-MDRP and the L-MDRP. For this reason, we did not try to solve the instances with 300 clients using the B&C based on model F^{\leq} .

In conclusion, the performance of the B&C approach is such that it is able to solve all the symmetric instances tested here with 100 clients and up to 20 depots within a minute of computation time. As for the instances with 200 clients and up to 40 depots, the B&C based on model $F^=$ is able to solve all instances within the time limit of three hours, except for the instance bgs-200-20 in the F-MDRP case, which is, however, solved by the B&C based on model F^{\leq} . Finally, we were unable to solve any of the instances with 300 clients and up to 60 depots within the time limit.

3.3.2 Asymmetric instances

This section reports the results obtained on the set C of instances, separately for the F-MDRP in Table 3 and for the L-MDRP in Table 4, much in the same way as in the previous section.

	B&C without 1-MCCs							B&C with 1-MCCs							
Name	LP	t_L	#CC	OPT	t	#CC	Feas?	LP	t_L	#CC	#MC	OPT	t	#CC	#MC
Using model $F^=$															
bgs-100-5-1a	898.241	2	1048	911	4	1100	no	898.349	3	1058	2	912	10	1470	38
bgs-100-10-1a	908.78	3	960	915	9	1329	no	911.35	4	964	25	921	15	1446	60
bgs-100-20-1a	948.75	1	811	953	3	635	no	969.471	3	843	113	979	19	749	942
bgs-100-5-2a	967.833	1	471	979	7	1300	no	967.948	1	475	4	980	8	1021	65
bgs-100-10-2a	986.5	1	431	992	3	559	no	988.778	1	450	13	998	8	778	176
bgs-100-20-2a	1033.75	0	403	1036	3	682	no	1055.8	2	429	195	1066	10	778	477
bgs-100-5-3a	917.25	6	1486	927	7	1241	no	917.25	6	1486	0	928	6	1219	8
bgs-100-10-3a	924.083	2	942	933	6	1070	no	939.419	3	986	53	955	46	1628	591
bgs-100-20-3a	947.6	2	1023	952	4	734	no	981.054	11	1056	351	994	70	984	2969
bgs-200-10a	1311.29	1	94	1315	5	210	no	1319.7	4	153	38	1325	19	355	62
bgs-200-20a	1340	1	105	1346	6	247	no	1348.97	6	112	82	1361	196	705	1558
bgs-200-40a	1464.58	2	77	1468	4	113	no	1492.26	53	135	709	1504	507	315	4840
bgs-300-10a	1631.12	6	201	[1642, 1651]	10802*	7570	no	1632.95	13	222	24	[1643, 1660]	10801*	12299	1543
bgs-300-20a	1647.35	9	285	1661	2064	2848	no	1656.83	42	339	120	[1668, 1696]	10801*	6696	10526
bgs-300-30a	1667.01	6	203	1679	547	1205	no	1681.12	74	255	431	[1695, 1720]	10802*	2686	15495
bgs-300-40a	1689.03	4	113	1697	196	682	no	1709.94	122	204	772	[1718, 1753]	10802*	1840	15993
bgs-300-60a	1764.83	4	79	1770	29	179	no	1802.05	1043	159	3213	[1815, 1821]	10803*	861	17823
Using model F^{\leq}															
bgs-100-5-1a	864.602	0	563	877	13	1765	no	898.349	1	598	9	912	5	851	30
bgs-100-10-1a	851.972	1	774	856	4	871	no	911.35	2	836	39	921	6	940	45
bgs-100-20-1a	839.5	4	1152	846	6	1048	no	969.471	9	1186	143	979	24	1176	578
bgs-100-5-2a	939.25	0	403	945	3	403	no	967.948	1	451	11	980	8	857	95
bgs-100-10-2a	929.9	1	491	932	2	483	no	988.778	1	638	27	998	5	716	82
bgs-100-20-2a	915.167	1	761	917	3	864	no	1055.8	5	1041	222	1066	18	1097	723
bgs-100-5-3a	891.548	3	1188	898	9	1340	no	917.25	3	1232	7	928	15	1684	35
bgs-100-10-3a	882.854	3	1102	888	8	1086	no	939.419	6	1173	85	955	62	2033	786
bgs-100-20-3a	858.667	9	1760	860	21	2274	no	981.054	26	1939	428	994	103	2591	3019
bgs-200-10a	1284.23	1	119	1291	8	342	no	1319.7	5	205	57	1325	36	463	106
bgs-200-20a	1273.75	2	118	1279	10	242	no	1348.97	9	221	143	1361	487	1208	3733
bgs-200-40a	1266.91	1	137	1274	18	457	no	1492.26	78	255	1235	1504	1246	415	7628

*Not solved to optimality within the limit of three hours.

Table 3: Evaluating the branch-and-cut for asymmetric instances (F-MDRP)

	B&C without 1-MCCs							B&C with 1-MCCs								
Name	LP	t_L	#CC	OPT	t	#CC	Feas?	LP	t_L	#CC	#MC	OPT	t	DU	#CC	#MC
Using model $F^=$																
bgs-100-5-1a	865	0	597	879	5	747	yes	865	1	597	0	879	5	2	747	0
bgs-100-10-1a	856.25	1	712	872	11	1420	no	856.25	1	712	0	873	20	4	1914	17
bgs-100-20-1a	848.847	6	1420	859	14	1817	no	848.847	6	1420	0	861	25	11	2223	17
bgs-100-5-2a	948.542	1	620	954	4	884	yes	948.542	0	620	0	954	5	1	884	0
bgs-100-10-2a	945.367	1	556	952	5	805	yes	945.367	1	556	0	952	5	3	805	0
bgs-100-20-2a	931	1	704	933	3	722	no	931	2	704	2	933	4	6	751	17
bgs-100-5-3a	893.417	2	978	904	12	1914	yes	893.417	1	978	0	904	13	2	1934	4
bgs-100-10-3a	890.917	2	1167	904	23	2684	yes	890.917	3	1167	0	904	28	3	2541	33
bgs-100-20-3a	871.067	27	2550	878	24	2502	no	871.067	29	2550	0	879	30	7	2774	21
bgs-200-10a	1291.88	1	138	1296	5	207	no	1291.88	1	138	0	1296	5	4	211	3
bgs-200-20a	1290.14	1	145	1295	6	299	no	1290.14	2	145	2	1295	5	6	282	1
bgs-200-40a	1288.33	1	122	1293	5	235	no	1288.33	1	122	0	1293	12	8	308	6
bgs-300-10a	1609.54	11	311	1626	10590	6648	no	1609.54	10	311	3	[1624, 1627]	10800*	2	6641	50
bgs-300-20a	1604.09	11	252	1621	4556	4257	no	1604.09	9	249	6	1622	10387	5	5729	100
bgs-300-30a	1599.22	7	224	1616	8380	3802	no	1599.22	10	224	5	[1614, 1617]	10800*	8	5604	399
bgs-300-40a	1593.11	6	227	1606	420	1172	no	1593.11	16	249	8	1606	799	11	1596	71
bgs-300-60a	1582.98	6	184	1591	134	487	no	1583.37	21	237	18	1592	533	16	883	249
Using model F^{\leq}																
bgs-100-5-1a	864.602	1	721	877	6	1045	no	865	1	734	4	879	11	2	1366	8
bgs-100-10-1a	851.972	2	920	856	3	853	no	856.25	2	940	9	873	25	4	2289	40
bgs-100-20-1a	839.5	4	1044	846	6	1165	no	848.847	4	1078	29	861	11	11	1445	39
bgs-100-5-2a	939.25	1	306	945	2	464	no	948.542	0	362	6	954	3	1	666	6
bgs-100-10-2a	929.9	1	646	932	2	612	no	945.367	2	776	12	952	4	3	872	14
bgs-100-20-2a	915.167	4	1341	917	2	542	no	931	6	1458	33	933	3	6	613	40
bgs-100-5-3a	891.548	4	1321	898	9	1529	no	893.417	4	1327	4	904	15	2	1937	6
bgs-100-10-3a	882.854	6	1389	888	6	1171	no	890.917	6	1448	17	904	24	3	2245	27
bgs-100-20-3a	858.667	7	1585	860	14	1884	no	871.067	7	1649	34	879	26	7	2397	67
bgs-200-10a	1284.23	1	116	1291	16	469	no	1291.88	3	235	16	1296	11	4	353	21
bgs-200-20a	1273.75	1	116	1279	7	270	no	1290.14	4	324	35	1295	13	6	422	38
bgs-200-40a	1266.91	2	127	1274	10	297	no	1288.33	5	265	59	1293	20	8	501	59

*Not solved to optimality within the limit of three hours.

Table 4: Evaluating the branch-and-cut for asymmetric instances (L-MDRP)

The results show that most of the observations made for the symmetric instances also apply here. One exception is the difference in the LP relaxation bounds obtained by the B&C with and without 1-MCC inequalities for the F-MDRP, which is expected given that Proposition 4 does not hold in the case of asymmetric instances. In particular, for 10 out of the 17 F-MDRP instances tested with the B&C based on model $F^=$, the LP bound obtained with the 1-MCC inequalities is larger than the optimal integer value obtained without them. This is the same for all instances tested with the formulation F^{\leq} . However, the LP bound values obtained with and without 1-MCC inequalities do not show any difference for the L-MDRP, when tested with formulation $F^=$, except in instance bgs-300-60a.

In addition, and in slight contrast to the observations made for symmetric instances on the integer solutions obtained without the use of 1-MCCs for the L-MDRP variant, there were four out of nine instances with 100 clients, three out of three instances with 200 clients and five out of five instances with 300 clients where the solution obtained is not feasible with respect to the 1-MCCs.

3.4 Evaluating the effect of the k -MCCs

In this section, we report our computational experience with using the more general k -MCC inequalities (23) for $k > 1$ to further improve the LP bounds provided by model $F^=$ with the B&C on instances with 100 clients. Due to Proposition 4, we have only tested the asymmetric instances in set C. The results are shown in Table 5 for the F-MDRP variant and Table 6 for the L-MDRP variant, where the k -MCC inequalities were separated for up to $k = 5$.

Name	OPT	B&C with 1-MCCs		B&C with k -MCCs up to $k = 5$	
		LP	t	LP	t
bgs-100-5-1a	912	898.349	3	898.349	0
bgs-100-10-1a	921	911.35	4	911.438	15
bgs-100-20-1a	979	969.471	3	969.762	819
bgs-100-5-2a	980	967.948	1	967.948	1
bgs-100-10-2a	998	988.778	1	988.778	17
bgs-100-20-2a	1066	1055.8	2	1055.8	894
bgs-100-5-3a	928	917.25	6	917.25	1
bgs-100-10-3a	955	939.419	3	939.633	13
bgs-100-20-3a	994	981.054	11	983.133	762

Table 5: Comparison results of the 1-MCC inequalities with the k -MCC inequalities up to $k = 5$ (F-MDRP)

Name	OPT	B&C with 1-MCCs		B&C with k -MCCs up to $k = 5$	
		LP	t	LP	t
bgs-100-5-1a	879	865	1	865	0
bgs-100-10-1a	873	856.25	1	856.25	10
bgs-100-20-1a	861	848.847	6	848.847	652
bgs-100-5-2a	954	948.542	0	948.542	0
bgs-100-10-2a	952	945.367	1	945.367	11
bgs-100-20-2a	933	931	2	931	931
bgs-100-5-3a	904	893.417	1	893.417	1
bgs-100-10-3a	904	890.917	3	890.917	16
bgs-100-20-3a	879	871.067	29	871.067	653

Table 6: Comparison results of the 1-MCC inequalities with the k -MCC inequalities up to $k = 5$ (L-MDRP)

The results show that only minor improvements in the LP bounds were obtained for the F-MDRP, namely for instances bgs-100-10-1a, bgs-100-20-1a, bgs-100-10-3a and bgs-100-20-3a, and no improvements in the case of the L-MDRP. The latter result is not surprising, given the earlier results with the 1-MCCs. The computational time requirements to solve the LP with k -MCCs with up to $k = 5$ increases considerably with the number of depots, which is expected.

The results speak in favor of not using generalizations in the form of k -MCCs for $k \geq 2$. In particular, the results suggest that there seems to be no computational benefit in terms of the LP bounds and reduction in the time to solve the instances to optimality. It would therefore suffice to use 1-MCCs which have reduced separation complexity and provide just as good bounds as their generalizations.

3.5 Comparing the 1-MCC with inequalities (30) and (31)

The discussion in Section 2.4 suggests that there should not be any dominance relationship between the 1-MCC inequalities (12) and the inequalities (30) and (31), at least in the space of the directed x variables. In this section, we compare the LP bounds obtained using the formulation $F^=$ by using either the 1-MCC inequalities (12) or using the inequalities (30) and (31), for both the F-MDRP and the L-MDRP. For the comparison within asymmetric instances, and for reasons explained at the end of Section 2.4, we have added the 1-MCC inequalities (12) for single node sets S to the inequalities (30) and (31) in order to obtain a valid formulation. Table 7 presents the comparison results for the F-MDRP variant and Table 8 for the L-MDRP. In our experiments, we used exact separation for the two sets of inequalities.

Name	OPT	With ineq. (30)–(31)		With 1-MCCs	
		LP	t	LP	t
100-5-1	38116	37560.5	3	37506	0
100-5-2	34018	33681.1	2	33679.5	0
100-5-3	33024	32622.5	1	32622.5	0
100-10-1	41991	41178.8	4	40847.5	0
100-10-2	39126	38680.7	4	38529	0
100-10-3	33719	33111.3	4	33015	1
200-10-1	53739	52847.8	37	52834.5	1
200-10-2	47441	46758.5	73	46758.5	2
200-10-3	47828	47454.2	86	47381.5	2
bgs-200-10	2301	2290.97	120	2290.75	4
bgs-200-20	2332	2306.97	103	2300.33	2
bgs-200-40	2417	2384.19	190	2375.5	3
bgs-100-5	1587	1579.5	3	1578.5	0
bgs-100-10	1597	1580.2	5	1576	1
bgs-100-20	1687	1660.75	4	1656.5	1
bgs-100-5-1a	912	898.241	1	898.349	3
bgs-100-10-1a	921	910.456	3	911.35	4
bgs-100-20-1a	979	958.188	4	969.471	3
bgs-100-5-2a	980	967.902	2	967.948	1
bgs-100-10-2a	998	988.689	2	988.778	1
bgs-100-20-2a	1066	1045.87	6	1055.8	2
bgs-100-5-3a	928	917.25	2	917.25	6
bgs-100-10-3a	955	935.762	4	939.419	3
bgs-100-20-3a	994	975.28	6	981.054	11
bgs-200-10a	1325	1317.62	75	1319.7	4
bgs-200-20a	1361	1346	90	1348.97	6
bgs-200-40a	1504	1479.14	195	1492.26	53

Table 7: Comparison results of the 1-MCC inequalities with inequalities (30) and (31) (F-MDRP)

Name	OPT	With ineq. (30)–(31)		With 1-MCCs	
		LP	t	LP	t
100-5-1	37943	37199.6	4	37190.5	1
100-5-2	30832	30456.5	2	30456.5	0
100-5-3	32850	31933.5	2	31922	0
100-10-1	39608	39078.5	4	39051.5	0
100-10-2	30641	30127.8	3	30121	0
100-10-3	32521	31860.8	4	31775	0
200-10-1	53047	52300.8	53	52291.5	2
200-10-2	39663	38963.8	55	38963.8	3
200-10-3	45289	44546.4	90	44535.5	3
bgs-200-10	2266	2246.5	92	2246.5	2
bgs-200-20	2250	2225	182	2224.5	3
bgs-200-40	2211	2168.21	307	2163.5	1
bgs-100-5	1569	1544	2	1542	1
bgs-100-10	1524	1500	2	1499.5	0
bgs-100-20	1509	1486	3	1484.5	0
bgs-100-5-1a	879	865	1	865	1
bgs-100-10-1a	873	856.25	2	856.25	1
bgs-100-20-1a	861	848.847	2	848.847	6
bgs-100-5-2a	954	948.542	2	948.542	0
bgs-100-10-2a	952	945.367	1	945.367	1
bgs-100-20-2a	933	931	3	931	2
bgs-100-5-3a	904	893.417	2	893.417	1
bgs-100-10-3a	904	890.917	2	890.917	3
bgs-100-20-3a	879	871.067	2	871.067	29
bgs-200-10a	1296	1291.88	29	1291.88	1
bgs-200-20a	1295	1290.14	49	1290.14	2
bgs-200-40a	1293	1288.33	46	1288.33	1

Table 8: Comparison results of the 1-MCC inequalities with inequalities (30) and (31) (L-MDRP)

The results concerning the symmetric instances indicate that there are improvements in the LP bound, in some cases significant, when inequalities (30) and (31) are used. This is the case for both variants of the problem, and is in contrast to using the 1-MCC inequalities (12). These results also suggest that for sparser instances with only a few client nodes linked to the depots, the previously known inequalities (30) could be more advantageous. However, the exact separation procedure used to identify violated inequalities (30) and (31) is more time-consuming than the separation procedure for finding violated 1-MCC inequalities. More precisely, if MF is the complexity of the max-flow algorithm, then separating the 1-MCC inequalities has a worst-case computational complexity of $O(MF \times |D|)$, since one max-flow has to be calculated for each depot, whereas separating inequalities (30) and (31) has a worst-case computational complexity of $O(MF \times |C|^2)$, since a max-flow has to be calculated for each pair of clients. In fact, the expression defining inequalities (30) and (31), even suggests that a max-flow has to be calculated for each pair of clients and each subset of depots. However, the exact separation procedure described in Belenguer et al. (2011) indicates a clever way of determining, for all inequalities associated to a given pair of client nodes i and j , the most violated inequality, if any exists, for all subsets of depots. This strategy was also used in our computational experiment.

For asymmetric instances we observe the opposite situation, but quite surprisingly, only for the F-MDRP instances. For the L-MDRP instances, the two sets of inequalities provide the same bounds.

A word of caution on the results above is in order. As previously pointed out, the comparison was done using exact separation, leading to the suggestion that the multi-cut constraints prove to be much easier and computationally more efficient on average to separate. The comparative findings may change, however, if other bespoke separation routines were to be devised and used for each of the set of inequalities, as was done by Benavent & Martínez-Sykora (2013) for inequalities (30) and (31).

4 Conclusions

Although multi-depot routing problems have long been studied, those that investigate the problem in its simplest setting are few and far between. Our study belongs to the latter category. In this paper, we have described new path elimination constraints for multi-depot routing problems, with and without location considerations, studied

their properties and connections between existing inequalities, and presented computational results to numerically assess their relative strength. The results suggest that the basic version of the multi-cut constraints, namely 1-MCCs, are computationally efficient and effective.

Through computational experiments, we have offered some insight into the nature of the problem and the way in which it is formulated. In particular, whilst the path elimination constraints are needed to ensure feasibility of the resulting solutions, we empirically showed that the way that the depot assignment constraints are modeled significantly affects the number of path elimination constraints required within the B&C algorithm, as well as the time spent to solve the problem to optimality.

We have also provided a generalization of the 1-MCC inequalities, namely the k -MCCs, however computational results do not indicate that they are computationally effective. Nevertheless, it is still an open question as to whether there might be other classes of instances, or variants of multi-depot problems, for which the more general k -MCCs might yield more significant improvements both in the LP relaxation bounds or in the computational times, in which case more efficient separation procedures for these constraints may be needed.

Acknowledgments

We thank the two reviewers whose comments have helped to improve the paper. L. Gouveia and D. Santos were supported by Portuguese National Funding from Fundação para a Ciência e a Tecnologia, under projects UID/MAT/04561/2013 and PTDC/MAT NAN/2196/2014. D. Santos was also supported by the PhD grant “Programa de Bolsas de Doutoramento da Universidade de Lisboa”.

References

- Ahuja, R. K., Magnanti, T. L. & Orlin, J. B. (1993), *Network flows: theory, algorithms, and applications*, Prentice-Hall, New Jersey.
- Albareda-Sambola, M. (2015), Location-routing and location-arc routing, in G. Laporte, S. Nickel & F. Saldanha da Gama, eds, ‘Location Science’, Springer, Cham, chapter 15, pp. 399–418.
- Albareda-Sambola, M., Díaz, J. A. & Fernández, E. (2005), ‘A compact model and tight bounds for a combined location-routing problem’, *Computers & Operations Research* **32**(3), 407–428.
- Bektaş, T. (2012), ‘Formulations and Benders decomposition algorithms for multidepot salesmen problems with load balancing’, *European Journal of Operational Research* **216**(1), 83–93.
- Belenguer, J.-M., Benavent, E., Prins, C., Prodhon, C. & Calvo, R. W. (2011), ‘A branch-and-cut method for the capacitated location-routing problem’, *Computers & Operations Research* **38**(6), 931–941.
- Benavent, E. & Martínez-Sykora, A. (2013), ‘Multi-depot multiple TSP: a polyhedral study and computational results’, *Annals of Operations Research* **207**(1), 7–25.
- Fernández, E. & Rodríguez-Pereira, J. (2016), ‘Multi-depot rural postman problems’, *TOP* pp. 1–33.
- Fischetti, M., Salazar-González, J.-J. & Toth, P. (1997), ‘A branch-and-cut algorithm for the symmetric generalized traveling salesman problem’, *Operations Research* **45**(3), 378–394.
- Godinho, M. T., Gouveia, L. & Pesneau, P. (2011), On a time-dependent formulation and an updated classification of ATSP formulations, in A. R. Mahjoub, ed., ‘Progress in Combinatorial Optimization’, ISTE-Wiley, p. 223–254.
- Hill, A. & Voß, S. (2016), ‘Optimal capacitated ring trees’, *EURO Journal on Computational Optimization* **4**(2), 137–166.
- Laporte, G., Nobert, Y. & Arpin, D. (1986), ‘An exact algorithm for solving a capacitated location-routing problem’, *Annals of Operations Research* **6**, 293–310.

- Laporte, G., Nobert, Y. & Pelletier, P. (1983), ‘Hamiltonian location problems’, *European Journal of Operational Research* **12**(1), 82–89.
- Öncan, T., Altinel, İ. K. & Laporte, G. (2009), ‘A comparative analysis of several asymmetric traveling salesman problem formulations’, *Computers & Operations Research* **36**(3), 637–654.
- Roberti, R. & Toth, P. (2012), ‘Models and algorithms for the asymmetric traveling salesman problem: an experimental comparison’, *EURO Journal on Transportation and Logistics* **1**(1), 113–133.
- Sundar, K. & Rathinam, S. (2017), ‘Multiple depot ring star problem: a polyhedral study and an exact algorithm’, *Journal of Global Optimization* **67**(3), 527–551.