Compact Formulations for Multi-depot Routing Problems: Theoretical and Computational Comparisons

Tolga Bektaş[1], Luís Gouveia[2*], Daniel Santos[3†]

[1]University of Liverpool Management School
Chatham Street, Liverpool, L69 7ZH, United Kingdom
t.bektas@liverpool.ac.uk

[2]Centro de Matemática, Aplicações Fundamentais e Investigação Operacional
Faculdade de Ciências, Universidade de Lisboa
C6 - Piso 4, 1749-016 Lisboa, Portugal
legouveia@fc.ul.pt

[3]Centro de Estudos de Gestão do Instituto Superior Técnico
Instituto Superior Técnico, Universidade de Lisboa
Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal
daniel.rebelo.santos@tecnico.ulisboa.pt

Abstract

Multi-depot routing problems mainly arise in distribution logistics where a fleet of vehicles are used to serve clients from a number of (potential) depots. The problem concerns deciding on the routes of each vehicle and the depots from which the vehicles depart, so as to minimize the total cost of travel. This paper reviews a number of existing compact formulations, and proposes new ones, for two types of multi-depot routing problems, one that includes the depot selection decisions, and the other where depots are pre-selected. The formulations are compared theoretically in terms of the strength of their linear programming relaxation, and computationally in terms of the running time needed to solve the instances to optimality.

Keywords: vehicle routing; multi-depot; location-routing; integer linear programming.

*Corresponding author.

†Most of the work by this author was done while affiliated with Centro de Matemática, Aplicações Fundamentais e Investigação Operacional, Faculdade de Ciências, Universidade de Lisboa.

1 Introduction

Multi-depot routing problems (MDRPs) involve routing a fleet of vehicles from several depots at minimal cost (e.g., distance) to serve a number of clients, where each vehicle returns back to the depot it has originated from. The locations of the depots may either be fixed *a priori*, the case which we will name as the Fixed-MDRP (F-MDRP), or alternatively chosen from a set of potential locations, which we will denote by the Location-MDRP (L-MDRP) in the ensuing exposition. MDRPs are generalizations of single-depot routing problems such as the traveling salesman problem (see, e.g., Lawler et al. 1985, Applegate et al. 2006) and the vehicle routing problem (see, e.g., Toth & Vigo 2014).

Formulations hitherto proposed for the MDRP contain two types of constraints, one requiring that each client is served in exactly one vehicle route, and the other guaranteeing that each route must contain exactly one depot. The latter is usually modelled by two complementary sets of constraints to ensure that each route contains (i) at least one, and (ii) at most one depot, respectively. If condition (i) is not satisfied, then there exists a route, or a *subtour*, disconnected from a depot and formed within clients only. Subtours are avoided by using the classical subtour elimination constraints (see, e.g., Öncan et al. 2009, Godinho et al. 2011, Roberti & Toth 2012, for surveys on the topic). As for condition (ii), if it is not satisfied it implies a vehicle traveling on a path between two different depots. Routes of such nature are prohibited by the use of the so-called path elimination constraints. One such class called chain-barring constraints is described by Laporte et al. (1983), later improved by Laporte et al. (1986). Improvements of chain-barring constraints are described by Belenguer et al. (2011) and Benavent & Martínez-Sykora (2013) later used in the work by Sundar & Rathinam (2017). A new set of multi-cut-like path elimination constraints, which differ from what was previously used in the literature, is described by Bektaş et al. (2017). All the path elimination constraints proposed in the studies just cited are exponential in size and therefore require separation routines for use in algorithms such as branch-&-cut. In contrast, Albareda-Sambola et al. (2005), Bektaş (2012) and Burger et al. (2018) present path elimination constraints that are polynomial in size.

Our motivation for this paper stems from the fact that compact formulations, i.e., those that are polynomial in the number of variables and constraints, are of practical significance as they can be readily used in combination with off-the-shelf optimization software. This is in contrast to other types of formulations described for the MDRP, some of which were described in the previous paragraph, that involve the use of exponentially sized sets of constraints. However, algorithms that are based on such formulations require the use of specialized methods, such as constraint separation, which may not always be easy to understand, implement and use. For this reason, in this paper, we exclusively focus on compact formulations for the MDRP. We extend and build on the MDRP literature by presenting compact sets of path elimination constraints, which we combine with compact sets of subtour elimination constraints to derive several valid formulations for the MDRP, to then compare them both theoretically and computationally. The contributions of this paper may be summarized as follows: (i) we review, propose and compare sets of compact path elimination constraints that can be used for both the F-MDRP and the L-MDRP, (ii) we describe formulations for the two variants of the MDRP obtained by combining different sets of path elimination and subtour elimination inequalities, (iii) we compare the formulations theoretically in terms of the strength of the linear programming relaxation, and computationally in terms of the time requirements for obtaining optimal solutions on a number of benchmark and randomly generated instances.

The following section presents a generic formulation for the MDRP, followed by a description of compact subtour elimination constraints in Section 3 and path elimination constraints in Section 4, the latter including theoretical comparison results. Details of the computational experimentation along with the results are described in Section 5. Extensions of the models and the results to a more general case of the MDRP are discussed in Section 6. The paper concludes in Section 7.

2 A generic formulation

We define the MDRP on a directed graph $G = (V, A)$. The set $V = \{1, 2, \ldots, n\}$ of nodes is partitioned into a set D of depots and a set C of clients. We assume, without loss of generality, that $D = \{1, \ldots, |D|\}$. Direct travel between two depots is not allowed, that is, no arcs between pairs of depots exist, so $A = \{(i, j) : i, j \in V \text{ and } \{i, j\} \cap C \neq \emptyset\}$ is the set of arcs. Any additional link $(i, j) \notin A$ that does not exist (e.g., loops) can simply be omitted from the formulations presented below. To simplify notation, we define the following partition of A:

$A^C = \{(i,j) \in A : i,j \in C,\ i \neq j\}$ is the set of arcs between client nodes, $A_O^d = \{(d,i) \in A : i \in C\}$ is the set of arcs emanating from depot $d \in D$, and $A_I^d = \{(i,d) \in A : i \in C\}$ is the set of arcs terminating at depot $d \in D$. For simplification we also define $A_O^D = \cup_{d \in D} A_O^d$ and $A_I^D = \cup_{d \in D} A_I^d$. Finally, we assume a non-negative cost c_{ij} associated with traversing arc $(i,j) \in A$.

The objective of the F-MDRP (L-MDRP) is to find a minimum cost set of exactly $|D|$ (at most $|D|$) disjoint circuits on graph G such that (i) each route starts and ends at the same depot, and (ii) each client in C is visited exactly once. To model both variants we define one binary variable x_{ij} that equals 1 if arc $(i,j) \in A$ is used in one of the circuits, and that equals 0 otherwise, and another binary variable y_d that equals 1 if depot $d \in D$ is used in one of the circuits, and that equals 0 otherwise. Using the two sets of variables, a valid and a generic integer linear programming formulation for the L-MDRP is given as follows.

$$\text{Minimize} \sum_{(i,j) \in A} c_{ij} x_{ij} \tag{1}$$

$$\text{subject to:} \sum_{j \in C} x_{dj} = y_d \qquad \forall d \in D \tag{2}$$

$$\sum_{j \in C} x_{jd} = y_d \qquad \forall d \in D \tag{3}$$

$$\sum_{j \in V} x_{ij} = 1 \qquad \forall i \in C \tag{4}$$

$$\sum_{j \in V} x_{ji} = 1 \qquad \forall i \in C \tag{5}$$

$$\{(i,j) \in A : x_{ij} = 1\} \text{ contains no circuit with zero depots} \tag{6}$$

$$\{(i,j) \in A : x_{ij} = 1\} \text{ contains no circuit with two or more depots} \tag{7}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{8}$$

$$y_d \in \{0,1\} \qquad \forall d \in D. \tag{9}$$

The following constraints are to be added to the above formulation for the F-MDRP,

$$y_d = 1 \qquad \forall d \in D, \tag{10}$$

in which case the y variables can be eliminated from the formulation. For simplicity, we will use the generic formulation of the L-MDRP in the remainder of the paper as basis, but point out the differences for the F-MDRP, if any exist.

Any solution to the system of inequalities (2)–(5) and (8)–(9) defines a set of disjoint circuits on graph G, which may not be feasible since circuits with zero depots and/or circuits with more than one depot may exist. The generic constraints (6) and (7) forbid these two situations, where the former correspond to subtour elimination constraints and the latter model the path elimination constraints. The two following sections present these constraints in more detail.

3 Subtour elimination constraints

This section describes the subtour elimination constraints that will later be used in deriving the MDRP formulations. Whilst different sets of subtour eliminations constraints have earlier been proposed for various routing problems, some with a stronger linear programming relaxation, we are guided by the comparative study of such constraints by Roberti & Toth (2012) in choosing the two more practically appealing ones for our purposes. These are (i) the single commodity flow model by Gavish & Graves (1978) and (ii) the improved Miller-Tucker-Zemlin constraints (see Miller et al. 1960) proposed by Desrochers & Laporte (1991). As these constraints are well-known from the literature, the rest of this section only presents their adaptions to the L-MDRP and the F-MDRP.

3.1 Single commodity flow formulations

The intuition behind the single commodity flow formulation of Gavish & Graves (1978), originally proposed for the traveling salesman problem, is to send a commodity from the depot to the clients through the underlying

graph, where one unit of flow is left at each client. Connectivity between the depot and every client node is therefore ensured, and subtours are prevented. In adapting this formulation to the MDRP, we define a set of continuous variables f_{ij}, each of which represents the amount of flow on arc $(i,j) \in A^C \cup A^D_O$. The following system of inequalities then models the generic subtour elimination constraints (6):

$$\sum_{d \in D} \sum_{j \in C} f_{dj} = |C| \tag{11}$$

$$\sum_{j \in V} f_{ji} - \sum_{j \in C} f_{ij} = 1 \qquad \forall i \in C \tag{12}$$

$$x_{dj} \leq f_{dj} \leq |C|\, x_{dj} \qquad \forall (d,j) \in A^D_O \tag{13}$$

$$x_{ij} \leq f_{ij} \leq (|C|-1)\, x_{ij} \qquad \forall (i,j) \in A^C. \tag{14}$$

Constraints (11) state that $|C|$ units of flow leave the set of depots. Constraints (12) ensure that one unit of flow is left at each client, where lower and upper bounds for the amount of flow on each arc are defined by constraints (13) and (14).

The upper bounds imposed on the f variables in constraints (13) and (14) can be tightened for the F-MDRP. In particular, since each depot must be used, then each depot must be serving at least one client, based on which constraints (13) and (14) can be replaced by:

$$x_{dj} \leq f_{dj} \leq (|C| - |D| + 1)\, x_{dj} \qquad \forall (d,j) \in A^D_O \tag{15}$$

$$x_{ij} \leq f_{ij} \leq (|C| - |D|)\, x_{ij} \qquad \forall (i,j) \in A^C. \tag{16}$$

We denote both the system of inequalities (11)–(14) when addressing the L-MDRP and the system of inequalities (11)–(12) and (15)–(16) when addressing the F-MDRP as simply SCF.

Variants on the SCF formulation, namely by using different linking constraints (e.g., without the inequalities on the lower bounds on the flow variables and/or by decreasing by -1 the value of the maximum flow in the arcs after the depot in the upper bound inequalities), are known from the literature. Our results indicate that no significant differences are observed by using these different variants. The variant we have chosen for defining the SCF model provides, in general, the (slightly) better results. To make a focused analysis of the advantages and disadvantages of the models we omit the results from these variants.

3.2 Improved Miller-Tucker-Zemlin formulations

Desrochers & Laporte (1991) presented improvements to the Miller-Tucker-Zemlin subtour elimination constraints originally proposed for the traveling salesman problem (see Miller et al. 1960). These constraints use a continuous variable u_i associated to each client $i \in C$ which can be interpreted as the relative position of node i in its tour. The adaptation of these constraints to the L-MDRP are as follows:

$$u_i - u_j + |C|\, x_{ij} + (|C|-2)\, x_{ji} \leq |C| - 1 \qquad \forall i,j \in C,\ i \neq j. \tag{17}$$

These inequalities ensure that (i) if arc (i,j) or arc (j,i) is used in a route, then the positions of nodes i and j differ by exactly 1, and (ii) if neither arc (i,j) nor (j,i) is used in a route, then the difference between u_i and u_j is at most $|C|-1$. This value is consistent with the position interpretation, particularly when i is the last and j is the first node on a single existing route.

Using a reasoning similar to what has been used to tighten (13) and (14) into (15) and (16), constraints (17) can be tightened for the F-MDRP as follows:

$$u_i - u_j + (|C| - |D| + 1)\, x_{ij} + (|C| - |D| - 1)\, x_{ji} \leq |C| - |D| \qquad \forall i,j \in C,\ i \neq j. \tag{18}$$

For simplification we denote inequalities (17) (L-MDRP) or (18) (F-MDRP) as DL.

It is well-known for related problems in the literature, such as the asymmetric traveling salesman problem, that the linear programming relaxation of the SCF system of inequalities implies the original Miller-Tucker-Zemlin constraints by (Miller et al. 1960) but not the improved ones, the DL inequalities, proposed by Desrochers & Laporte (1991). In fact, it is possible to see that the DL inequalities, (17) or (18), imply the subtour elimination constraints for two client nodes, namely

$$x_{ij} + x_{ji} \leq 1 \qquad \forall (i,j) \in A^C, \tag{19}$$

which the SCF system of inequalities does not.

4 Path elimination constraints

This section is structured in two parts. We first describe different types of path elimination constraints for the MDRP in Section 4.1. As path elimination constraints are much less studied in the literature as compared to their subtour elimination counterparts, we also present several theoretical comparison results in Section 4.2.

4.1 Formulations

This section describes four types of path elimination constraints for MDRPs, namely those that use (i) single index node variables that indicate the "value" of the depot that defines the circuit the node appears in, (ii) two-index arc variables that indicate the "value" of the depot that defines the circuit the arc appears in, (iii) two-index client-depot variables that associate a client with the depot that defines the circuit the client is in, and (iv) three-index arc-depot variables that associate an arc with the depot that defines the circuit the arc is in. Constraints of type (i) are an improved version of ones proposed earlier in the literature. Constraints of type (ii) are new. Constraints of type (iii) are presented here for the first time for MDRPs, although they have previously been used for modeling other related problems, and constraints of type (iv) are known from the literature.

4.1.1 Node current formulations

The study by Burger et al. (2018) proposes the use of so-called currents (e.g., as in electrical) to model path elimination constraints in MDRPs, with each depot assigned a unique value. Clients that are in the circuit of a given depot must therefore also have the same current as that depot. The formulation uses a continuous variable k_i that indicates the current of node $i \in C$ and is as follows:

$$k_i + (|D| - d)\, x_{di} \leq |D| \qquad \forall d \in D,\ \forall i \in C \tag{20}$$
$$k_i + (|D| - d)\, x_{id} \leq |D| \qquad \forall d \in D,\ \forall i \in C \tag{21}$$
$$1 + (d - 1)\, x_{di} \leq k_i \qquad \forall d \in D,\ \forall i \in C \tag{22}$$
$$1 + (d - 1)\, x_{id} \leq k_i \qquad \forall d \in D,\ \forall i \in C \tag{23}$$
$$k_i - k_j \leq (|D| - 1)\,(1 - x_{ij} - x_{ji}) \qquad \forall i, j \in C,\ i \neq j. \tag{24}$$

For a depot $d \in D$ and a client $i \in C$, constraints (20) and (22) guarantee that if arc (d, i) is in the solution then the current of node i is equal to d. In a symmetric way constraints (21) and (23) guarantee that if arc (i, d) is in the solution then the current of i is equal to d. Constraints (24) ensure that two clients that are linked must have the same current.

Two important improvements are presented compared to the original formulation by Burger et al. (2018):

1. Constraints (24) are a lifted version of a weaker set of constraints, namely,

$$k_i - k_j \leq (|D| - 1)\,(1 - x_{ij}) \qquad \forall i, j \in C,\ i \neq j \tag{25}$$
$$k_j - k_i \leq (|D| - 1)\,(1 - x_{ij}) \qquad \forall i, j \in C,\ i \neq j. \tag{26}$$

   The lifted constraints (24) are significantly less in number and thus are superior both in theory and in practice in comparison to constraints (25)–(26). A similar idea to the one used to derive constraints (24) (i.e., subtours are not permitted) cannot be used to lift constraints (20)–(23) since in this case circuits with a single depot and client would be forbidden. Although these lifted inequalities have already been suggested in the work of Burger et al. (2018), they have not been used in the computational experiments therein since the distinction of arcs between clients and arcs incident with depots was not made.

2. In the node current model proposed in this paper, the constraints assigning currents to the nodes directly linked to depots are different from the constraints proposed in Burger et al. (2018). In fact, the former work uses constraints $k_d = d$ for all $d \in D$ as well as constraints (25) and (26) also for arcs involving depot nodes. In our model, the strategy is to guarantee that a client node linked to a depot has the current of the depot, without explicitly needing to assign currents to the depot nodes. More importantly is the fact that constraints (20)–(23) are stronger than the ones in Burger et al. (2018). If we add $(d - 1)\, x_{di} \leq d - 1$ to constraint (20) we obtain $k_i + (|D| - 1)\, x_{di} \leq |D| + d - 1$, which can be rewritten as one of the original

constraints in Burger et al. (2018), namely $k_i - d \leq (|D| - 1)\,(1 - x_{di})$ which is constraint (26) for the arc (d, i) after using the equality $k_d = d$. In a similar manner we show that constraint (21) implies constraint (25) for arc (d, i). Concerning the other two sets of constraints, observe that if we add $(|D| - d)\,x_{di} \leq |D| - d$ to constraint (22) we obtain constraint (26) for the arc (i, d) after using the equality $k_d = d$. A similar reasoning shows that constraint (23) implies constraint (25) for arc (i, d).

We denote the system of inequalities (20)–(24) as NODE-DL and the system of inequalities (20)–(23) and (25)–(26) as NODE. The DL suffix for the stronger model follows from the fact that strengthening constraints (25)–(26) into (24) by adding the reversed arc is similar to the way that the Miller-Tucker-Zemlin constraints are strengthened into the DL inequalities (17). Finally, note that both NODE-DL and NODE are valid for either MDRP variant studied in this work.

4.1.2 Arc current formulation

The arc current formulation adopts the same idea of the node current formulation but associates the currents to arcs instead. The relationship between the two is akin to that between the single commodity flow formulation and the original Miller-Tucker-Zemlin constraints, where the interpretation of the extra "position" variables is associated with arcs in the former and nodes in the latter (see Section 3). The arc current formulation uses a continuous variable g_{ij} indicating the current of arc $(i, j) \in A$, using which path elimination constraints can be written as follows:

$$g_{di} = d\,x_{di} \qquad \forall d \in D,\ \forall i \in C \tag{27}$$

$$g_{id} = d\,x_{id} \qquad \forall d \in D,\ \forall i \in C \tag{28}$$

$$\sum_{j \in V} g_{ji} - \sum_{j \in V} g_{ij} = 0 \qquad \forall i \in C \tag{29}$$

$$x_{ij} \leq g_{ij} \leq |D|\,x_{ij} \qquad \forall (i, j) \in A^C. \tag{30}$$

Constraints (27) and (28) state that the current of the arcs leaving and entering a depot $d \in D$ is equal to d, respectively. Constraints (29) relate to flow conservation and ensure that the current of an arc that belongs to the circuit of a depot is the same as that of the depot. Finally, constraints (30) link the g and the x variables, and define lower and upper bounds on the arc currents for arcs between clients. We denote the system of inequalities (27)–(30) as ARC.

4.1.3 Client-depot assignment formulations

Consider the variables v_d^i that are equal to 1 if client $i \in C$ is assigned to (or is included in the circuit of) depot $d \in D$, and to 0 otherwise. Similar variables have been used to model other problems, such as the Hamiltonian p-median problem (Gollowitzer et al. 2014, Erdoğan et al. 2016, Bektaş et al. 2019) and data clustering (Benati et al. 2017). Variables v are naturally binary, however we may simply define them as continuous seeing as the system of inequalities which we present below implicitly guarantees that they only take either value 0 or 1 in any feasible solution. The following system of inequalities, which we denote as CDA, models path elimination constraints:

$$x_{di} \leq v_d^i \qquad \forall d \in D,\ \forall i \in C \tag{31}$$

$$x_{id} \leq v_d^i \qquad \forall d \in D,\ \forall i \in C \tag{32}$$

$$\sum_{d \in D} v_d^i = 1 \qquad \forall i \in C \tag{33}$$

$$v_d^i + x_{ij} \leq v_d^j + 1 \qquad \forall d \in D,\ \forall i, j \in C,\ i \neq j. \tag{34}$$

Constraints (31) ensure that $i \in C$ is included in the circuit of depot $d \in D$ (i.e., $v_d^i = 1$) if arc (d, i) is used. Constraints (32) model the same condition in the same manner for arc (i, d). Constraints (33) guarantee that every client is in one and only one circuit. These could be written as less than or equal to constraints and the resulting formulation would still be valid since the client degree constraints (4)–(5) and the subtour elimination constraints (6) guarantee that every client is visited. The reason we point out these two alternative versions of

(33) will be made clear in a subsequent section. Finally, constraints (34) state that if a client $i \in C$ is in the circuit of a depot $d \in D$ and if the arc (i, j) from client i to client $j \in C \setminus \{i\}$ is used, then j must also be in the circuit of depot d.

The CDA system can be strengthened in two different ways, one of them achieved through the lifting of inequalities (34), as described below:

1. The first is based on the observation that constraints similar to constraints (34) where the term x_{ij} is replaced with x_{ji} are valid. By adding these to the CDA system of inequalities, slight improvements are obtained in the linear programming relaxation value. However, we do not elaborate more on this model since it is dominated by the model discussed in the next point. Furthermore the enhanced model has fewer constraints.

2. The second is based on a set similar to the disaggregated Desrochers and Laporte constraints originally proposed by Gouveia & Pires (1999) for the precedence constrained traveling salesman problem, resulting in the following:
$$v_d^i + x_{ij} + x_{ji} \leq v_d^j + 1 \qquad \forall d \in D,\ \forall i, j \in C,\ i \neq j. \tag{35}$$
The validity of inequalities (35) easily follows from the fact that subtours between clients are not permitted. We will denote the system of inequalities (31)–(33) and (35) as CDA-DDL. Clearly, CDA-DDL has the same number of variables and constraints as CDA, but with a linear programming relaxation value theoretically at least as good, and better in practice.

   Although this paper exclusively concerns compact formulations, it is worth noting that constraints (35) can be generalized in two steps as described below:

   - The first generalization accommodates a larger subset $S \subset C$ of clients and results in the exponentially large set of constraints shown below, where $x(S)$ is defined as the summation of the x variables over all arcs involving nodes in S,
$$v_d^i + x(S) \leq v_d^j + |S| - 1 \qquad \forall d \in D,\ \forall i, j \in S \subset C,\ i \neq j, \tag{36}$$
     the validity of which follows straightforwardly from the fact that subtours involving only client nodes are not allowed. The generalization is in the same spirit as was done for the precedence constrained traveling salesman problem (see, e.g., Gouveia & Pires 1999, Gouveia & Pesneau 2006, Gouveia et al. 2018) as, in fact, the v variables may be interpreted as precedence variables. We denote inequalities (36) as generalized disaggregated Desrochers and Laporte (GDDL) inequalities, and the system of inequalities (31)–(33) and (36) as CDA-GDDL.
   - A further generalization incorporates a larger set of depots as shown below
$$\sum_{d \in D'} v_d^i + x(S) \leq \sum_{d \in D'} v_d^j + |S| - 1 \qquad \forall D' \subset D,\ \forall i, j \in S \subset C,\ i \neq j, \tag{37}$$
     which is similar to a generalization described for the Hamiltonian p-median problem (see Erdoğan et al. 2016). The validity of (37) follows straightforwardly from constraints (33) and the GDDL constraints (36). We denote the system of inequalities (31)–(33) and (37) as GCDA-GDDL. Observe that constraints (36) are a particular case of (37) obtained by setting $|D'| = 1$. Finally, as a side comment we also refer to the following weaker set of inequalities since they will appear in a proof to be presented later on:
$$\sum_{d \in D'} v_d^i + x_{ij} \leq \sum_{d \in D'} v_d^j + 1 \qquad \forall D' \subset D,\ \forall i, j \in C,\ i \neq j. \tag{38}$$
     For the purpose of formulation comparisons (see Section 4.2.3), we denote as GCDA the system of inequalities (31)–(33) and (38).

   A unique feature of the GDDL constraints is that, when used in a formulation such as CDA-GDDL or GCDA-GDDL, they serve to eliminate *both* paths and subtours, which renders any additional set of constraints from Section 3 redundant. Whilst exponential-size formulations, such as the ones above, are not of primary concern

in this paper and will therefore not be computationally tested, we leave the study of separation algorithms for constraints such as (36) and (37) for further research, in order that their performance in comparison with the state-of-the-art (see Belenguer et al. 2011, Benavent & Martínez-Sykora 2013, Bektaş et al. 2017) can be assessed.

4.1.4 Arc-depot assignment formulations

The arc-depot assignment formulations presented in this section, which have been used in related problems (see, e.g., Albareda-Sambola et al. 2005, Bektaş 2012, Fernández & Rodríguez-Pereira 2016, Hill & Voß 2016), use arc-depot assignment variables z_{ij}^d equal to 1 if arc $(i,j) \in A^C \cup A_O^d \cup A_I^d$ is used in the circuit of depot $d \in D$, and to 0 otherwise. Similarly to the client-depot assignment variables v, the z variables are defined as binary but, given their network flow interpretation (see Bektaş et al. 2017), they can be redefined as continuous. Path elimination constraints may be modeled using the following system of inequalities, denoted by 3I in the remainder of the paper:

$$\sum_{j \in C} z_{dj}^d = y_d \qquad \forall d \in D \tag{39}$$

$$\sum_{j \in C} z_{jd}^d = y_d \qquad \forall d \in D \tag{40}$$

$$\sum_{j \in \{d\} \cup C} z_{ji}^d = \sum_{j \in \{d\} \cup C} z_{ij}^d \qquad \forall d \in D,\ \forall i \in C \tag{41}$$

$$\sum_{d \in D} z_{ij}^d \leq x_{ij} \qquad \forall (i,j) \in A^C \tag{42}$$

$$z_{ij}^d = x_{ij} \qquad \forall d \in D,\ \forall (i,j) \in A_O^d \cup A_I^d \tag{43}$$

$$z_{ij}^d \geq 0 \qquad \forall d \in D,\ \forall (i,j) \in A^C. \tag{44}$$

Constraints (39) and (40) state that for each depot $d \in D$ that is used, there must be an arc outgoing d and an arc ingoing d which is used in the circuit of depot d, respectively. Constraints (41) ensure that the outdegree and the indegree of each client node is the same for every circuit. Finally, constraints (42)–(43) link the z and the x variables by ensuring that an arc is used in a given circuit if and only if the corresponding x variable for that is equal to 1. Recently, Bektaş et al. (2017) showed that this system of inequalities is closely related to a set of exponentially-many multi-cut path elimination inequalities that have been used in a branch-&-cut method described in the same work.

An alternative formulation, which we denote as 3I$^+$, is obtained if we replace (42) with

$$\sum_{d \in D} z_{ij}^d = x_{ij} \qquad \forall (i,j) \in A^C. \tag{45}$$

In the work by Bektaş et al. (2017) the question of whether 3I and 3I$^+$ are equivalent was left unanswered. We now prove that they are not equivalent in terms of the corresponding polyhedra, however the linear programming relaxation values are the same if the objective function depends only on the x and the y variables.

Proposition 1. *Consider two models, model A comprised of the assignment constraints* (2)–(5)*, the domain constraints for the x and the y variables* (8)–(9) *and the 3I system of inequalities* (39)–(44)*, and model B comprised of the assignment constraints* (2)–(5)*, the domain constraints for the x and the y variables* (8)–(9) *and the 3I$^+$ system of inequalities* (39)–(41) *and* (43)–(45)*. Given an objective function which only depends on the x and the y variables, both models provide the same linear programming relaxation value.*

Proof. The idea of this proof is to show that we can construct a solution of the linear programming relaxation of model B based on a solution of the linear programming relaxation of model A where the part on the x and the y variables is the same and, thus, the difference is only on the part on the z variables. For this reason, both solutions will have the same cost since we assume that the objective function only depends on the x and the y variables.

Let (x', y', z') be a solution of the linear programming relaxation of model A and, for all arcs $(i,j) \in A^C$, define $\epsilon_{ij} = x'_{ij} - \sum_{d \in D} z'^d_{ij}$. We start by proving that $\sum_{j \in C} \epsilon_{ji} = \sum_{j \in C} \epsilon_{ij}$ for any $i \in C$. First notice that from the client outdegree constraints (4) we have that:

$$\sum_{j \in C} x'_{ji} = \sum_{j \in C} x'_{ij} + \sum_{d \in D} x'_{id} - \sum_{d \in D} x'_{di}.$$

Additionally, from constraints (41) and (43), we can derive:

$$\sum_{d \in D} \sum_{j \in C} z'^d_{ji} = \sum_{d \in D} \sum_{j \in C} z'^d_{ij} + \sum_{d \in D} z'^d_{id} - \sum_{d \in D} z'^d_{di} = \sum_{d \in D} \sum_{j \in C} z'^d_{ij} + \sum_{d \in D} x'_{id} - \sum_{d \in D} x'_{di}.$$

Thus, by combining the two above equalities, we obtain:

$$\sum_{j \in C} \epsilon_{ji} = \sum_{j \in C} x'_{ji} - \sum_{d \in D} \sum_{j \in C} z'^d_{ji} = \sum_{j \in C} x'_{ij} - \sum_{d \in D} \sum_{j \in C} z'^d_{ij} = \sum_{j \in C} \epsilon_{ij}.$$

Consider now an arbitrary depot $k \in D$ and a new solution (x', y', z'') where $z''^d_{ij} = z'^d_{ij}$ for all $d \in D \setminus \{k\}$ and for all $(i,j) \in A^C \cup A^d_O \cup A^d_I$, where $z''^k_{ij} = z'^k_{ij} + \epsilon_{ij}$ for all $(i,j) \in A^C$, and where $z''^k_{ij} = z'^k_{ij}$ for all $(i,j) \in A^d_O \cup A^d_I$. Observe that (x', y', z') and (x', y', z'') have the same cost given an objective function which only depends on the x and the y variables. In order to complete the proof, we only need to show that (x', y', z'') is a feasible solution of the linear programming relaxation of model B.

Clearly, (x', y', z'') satisfies the assignment relaxation constraints (2)–(5) by definition. Additionally, and also by definition, (x', y', z'') satisfies constraints (39)–(40), the linking constraints (43) and the flow conservation constraints (41) for all $d \in D \setminus \{k\}$. Observe also that, by the definition of ϵ, the solution (x', z'') satisfies the linking constraints (45). Finally, in order to complete the proof, we now need to verify that (x', y', z'') satisfies the flow conservation constraints (41) for depot k.

Consider $i \in C$ and observe that

$$\sum_{j \in \{k\} \cup C} z''^k_{ji} = \sum_{j \in \{k\} \cup C} z'^k_{ji} + \sum_{j \in C} \epsilon_{ji} = \sum_{j \in \{k\} \cup C} z'^k_{ij} + \sum_{j \in C} \epsilon_{ij} = \sum_{j \in \{k\} \cup C} z''^k_{ij},$$

which follows from the fact that z' satisfies the flow conservation constraints (41) for depot k and from the result proved above for ϵ. This completes the proof since we have shown that (x', y', z'') is a solution of the linear programming relaxation of model B. □

Observe that the proof of Proposition 1 implicitly shows that the 3I$^+$ system may improve the linear programming relaxation values if other cost functions are used, more specifically if these are functions of the z variables, which means that the 3I and the 3I$^+$ system are not equivalent. However, for the objective function (1) used in the models described in this work, Proposition 1 shows that either system provides the same linear programming relaxation value. In the computational experiment we will only provide results for the 3I system seeing as having less than or equal constraints ensures the corresponding dual variables to be non-negative. This observation was confirmed in preliminary tests, where the 3I$^+$ system performed consistently worse than the 3I system.

We conclude this section by pointing out that a weaker (from a linear programming relaxation point of view) but still valid model can be obtained by replacing constraints (42) with their weaker version

$$z^d_{ij} \leq x_{ij} \qquad \forall d \in D,\ \forall (i,j) \in A^C. \tag{46}$$

These weaker constraints will be mentioned later on in the next section.

4.2 Theoretical comparisons of formulations

This section presents theoretical results comparing the several sets of path elimination constraints discussed in Section 4.1. Observe that the linear programming relaxation value of a formulation for the F-MDRP or the L-MDRP depends on both the subtour elimination and the path elimination constraints used. Therefore, we only present results focused on the path-elimination subset. These results will be independent of the set of subtour elimination constraints used.

4.2.1 Comparison of the node current and the arc current formulations

In order to compare the node current and the arc current formulations we start by relating the k and the g variables through the following equalities:

$$k_i = \sum_{j \in V} g_{ji} \qquad \forall i \in C. \tag{47}$$

Equivalently, under constraints (29), we can also write

$$k_i = \sum_{j \in V} g_{ij} \qquad \forall i \in C. \tag{48}$$

We now prove that the ARC system of inequalities dominates the NODE system of inequalities. For the proof we add the equalities (47) (or (48)) to the ARC system of inequalities. Observe that this addition does not change its linear programming relaxation value since these equalities are merely definitional constraints.

Proposition 2. *The system of inequalities defined by the assignment constraints* (2)–(5)*, the ARC system of inequalities* (27)–(30)*, the relationships* (47) *(or* (48)*) and* $0 \leq x_{ij} \leq 1,\ \forall (i,j) \in A$ *and* $0 \leq y_d \leq 1,\ \forall d \in D$ *implies the NODE system of inequalities* (20)–(23) *and* (25)–(26)*.*

Proof. Consider a client $i \in C$ and a depot $d \in D$. From the relationship (47) and constraints (27) we can derive:

$$k_i = \sum_{j \in V} g_{ji} = d\, x_{di} + \sum_{j \in V \setminus \{d\}} g_{ji}.$$

Then, from the upper bounds in constraints (30) and subsequently from the assignment constraints (5) we obtain

$$d\, x_{di} + \sum_{j \in V \setminus \{d\}} g_{ji} \leq d\, x_{di} + |D| \sum_{j \in V \setminus \{d\}} x_{ji} = d\, x_{di} + |D|\ (1 - x_{di}),$$

leading to

$$k_i \leq |D| - (|D| - d)\ x_{di},$$

which are constraints (20) written for i and d. Using similar arguments, we can show that constraints (21)–(23) are also implied by the ARC system of inequalities. As for constraints (25), observe that, for any $i, j \in C,\ i \neq j$, we have:

$$k_i - k_j = \sum_{l \in V} g_{il} - \sum_{l \in V} g_{lj} = \sum_{l \in V \setminus \{j\}} g_{il} - \sum_{l \in V \setminus \{i\}} g_{lj}.$$

By using the lower and the upper bound constraints (30) we can derive:

$$\sum_{l \in V \setminus \{j\}} g_{il} - \sum_{l \in V \setminus \{i\}} g_{lj} \leq |D| \sum_{l \in V \setminus \{j\}} x_{il} - \sum_{l \in V \setminus \{i\}} x_{lj}.$$

Finally, by using the assignment constraints (4)–(5) we obtain:

$$|D| \sum_{l \in V \setminus \{j\}} x_{il} - \sum_{l \in V \setminus \{i\}} x_{lj} = |D|\ (1 - x_{ij}) - (1 - x_{ij}) = (|D| - 1)\ (1 - x_{ij}).$$

Constraints (26) can be shown to be implied in a similar way. □

It was this result and the relationship between these two models that have motivated the improved constraints (20)–(23) presented in Section 4.1.1. Observe, however, that the arc current formulation does not imply the stronger constraints (24) of the NODE-DL model. This is easy to see since, because of these constraints, the NODE-DL formulation implies subtour elimination constraints for two client node subsets, that is, constraints (19), and the arc current formulation does not. This no dominance relation between the ARC formulation and the NODE-DL formulation is similar to the no dominance relationship between the SCF and the improved MTZ formulations mentioned at the end of Section 3.2. We believe that these observations explain why the NODE-DL formulation produces better lower bounds than the ARC formulation does when using the SCF as subtour eliminations constraints but worse lower bounds when using the improved MTZ subtour eliminations constraints (since these constraints already imply (19)).

4.2.2 Comparison of the arc current and the arc-depot assignment formulations

We now show that the 3I$^+$ system of inequalities implies the ARC system of inequalities. For the proof we add the following relationships between the g and the z variables

$$g_{ij} = \sum_{d \in D} d\, z_{ij}^d \qquad \forall (i,j) \in A^C \tag{49}$$

$$g_{ij} = d\, z_{ij}^d \qquad \forall d \in D,\ \forall (i,j) \in A_O^d \cup A_I^d, \tag{50}$$

to the system of inequalities (39)–(41) and (43)–(45), that define the 3I$^+$ system, without altering its linear programming relaxation value (as before, observe that the added inequalities are only definitional).

Proposition 3. *The system of inequalities defined by the assignment constraints* (2)–(5)*, the 3I$^+$ system of inequalities* (39)–(41) *and* (43)–(45)*, the relationships* (49)–(50) *and* $0 \le x_{ij} \le 1,\ \forall (i,j) \in A$ *and* $0 \le y_d \le 1,\ \forall d \in D$ *implies the ARC system of inequalities* (27)–(30)*.*

Proof. Constraints (27)–(28) follow from relationships (50) and constraints (43). As for the current conservation constraints (29), consider $i \in C$. If for each $d \in D$ we multiply the corresponding constraint (41) by d and add everything up we obtain

$$\sum_{d \in D} \sum_{j \in \{d\} \cup C} d\, z_{ji}^d = \sum_{d \in D} \sum_{j \in \{d\} \cup C} d\, z_{ij}^d,$$

which may be rewritten as:

$$\sum_{d \in D} d\, z_{di}^d + \sum_{j \in C} \sum_{d \in D} d\, z_{ji}^d = \sum_{d \in D} d\, z_{id}^d + \sum_{j \in C} \sum_{d \in D} d\, z_{ij}^d.$$

From relationships (49)–(50) the left-hand side of the above equalities may be written as $\sum_{j \in V} g_{ji}$ and the right-hand side as $\sum_{j \in V} g_{ij}$, thus implying constraints (29). Finally, for $(i,j) \in A^C$ observe that

$$g_{ij} = \sum_{d \in D} d\, z_{ij}^d \le |D| \sum_{d \in D} z_{ij}^d = |D|\, x_{ij},$$

and that

$$g_{ij} = \sum_{d \in D} d\, z_{ij}^d \ge \sum_{d \in D} z_{ij}^d = x_{ij},$$

both following from relationships (49) and constraints (45), which completes the proof. □

4.2.3 Comparison of the client-depot assignment and the arc-depot assignment formulations

To compare the client-depot assignment and the arc-depot assignment formulations we observe that the v variables, of the client-depot assignment formulations, and the z variables, of the arc-depot assignment formulations, can be related as follows:

$$\sum_{j \in \{d\} \cup C} z_{ji}^d = v_d^i \qquad \forall d \in D,\ \forall i \in C. \tag{51}$$

Once again, adding these equalities to the 3I$^+$ system does not alter its linear programming relaxation value. Also, by using the flow conservation constraints (41) we can derive the following equalities that are equivalent and similar to (51)

$$\sum_{j \in \{d\} \cup C} z_{ij}^d = v_d^i \qquad \forall d \in D,\ \forall i \in C. \tag{52}$$

The following result shows that the 3I$^+$ system of inequalities implies the GCDA system of inequalities.

Proposition 4. *The system of inequalities defined by the assignment constraints* (2)–(5)*, the 3I$^+$ system of inequalities* (39)–(41) *and* (43)–(45)*, the relationships* (51) *(or* (52)*) and* $0 \le x_{ij} \le 1,\ \forall (i,j) \in A$ *and* $0 \le y_d \le 1,\ \forall d \in D$ *implies the GCDA system of inequalities* (31)–(33) *and* (38)*.*

Proof. Regarding constraints (31) and (32) notice that from the relationships (51)–(52) for nodes $d \in D$ and $i \in C$ and the linking constraints between the z and the x variables (43) we get:

$$v_d^i = \sum_{j \in \{d\} \cup C} z_{ji}^d \geq z_{di}^d = x_{di}, \text{ and } v_d^i = \sum_{j \in \{d\} \cup C} z_{ij}^d \geq z_{id}^d = x_{id}.$$

For constraints (33), consider $i \in C$ and notice that if we add the relationships (51) for all depots we get:

$$\sum_{d \in D} v_d^i = \sum_{d \in D} \sum_{j \in \{d\} \cup C} z_{ji}^d = \sum_{d \in D} z_{di}^d + \sum_{d \in D} \sum_{j \in C} z_{ji}^d.$$

Then, by using the linking constraints between the z and the x variables (43) and (45) and, subsequently, constraints (5) we get:

$$\sum_{d \in D} z_{di}^d + \sum_{d \in D} \sum_{j \in C} z_{ji}^d = \sum_{d \in D} x_{di} + \sum_{j \in C} x_{ji} = 1.$$

In the case of constraints (38), consider a subset $D' \subset D$ and a client $j \in C$. If we add the relationships (51) for the depots in D' we obtain:

$$\sum_{d \in D'} v_d^j = \sum_{d \in D'} \sum_{k \in \{d\} \cup C} z_{kj}^d \geq \sum_{d \in D'} z_{ij}^d,$$

for some $i \in C \setminus \{j\}$. Now, if we add $\sum_{d \in D'} \sum_{k \in \{d\} \cup C:\, k \neq j} z_{ik}^d$ to both sides and use the relationships (52) for client i added up for the depots in D' we obtain:

$$\sum_{d \in D'} v_d^j + \sum_{d \in D'} \sum_{k \in \{d\} \cup C:\, k \neq j} z_{ik}^d \geq \sum_{d \in D'} \sum_{k \in \{d\} \cup C} z_{ik}^d = \sum_{d \in D'} v_d^i.$$

Then, by using the linking constraints (43) and (45) we derive:

$$\sum_{d \in D'} v_d^j + \sum_{k \in V \setminus \{j\}} x_{ik} \geq \sum_{d \in D'} v_d^i.$$

Finally, if we use the constraints (4) for client i we can rewrite the above inequality as

$$\sum_{d \in D'} v_d^j + 1 \geq x_{ij} + \sum_{d \in D'} v_d^i,$$

which is precisely a constraint (38). □

Three observations concerning the previous proof can be made:

i) The weaker linking constraints (46) would suffice to derive constraints (38);

ii) As observed in Section 4.1.3, the CDA system of inequalities and its lifted/generalized versions would still be valid if the equality constraints (33) are written as "less than or equal to" constraints. To obtain these constraints we could have used (42) instead of (45). In other words, the difference between having equality or inequality in constraints (33) is similar to the difference between the 3I and 3I$^+$ systems established in Proposition 1.

iii) A similar proof shows that the symmetric alternative variants of constraints (38), that is, with x_{ji} instead of x_{ij}, can also be shown to be implied. However, it is not difficult to show that the implication no longer holds if we use $x_{ij} + x_{ji}$ as in the DDL variants.

4.2.4 Comparison of the node current and the client-depot assignment formulations

To conclude the relationships between formulations presented in this paper, it remains to establish whether the client-depot assignment formulations dominate the node current formulations. Contrary to what might be expected this is not the case. Observe, first, that the k variables, of the node current formulations, and the v variables, of the client-depot assignment formulations, are related as follows:

$$k_i = \sum_{d \in D} d\, v_d^i \qquad \forall \in C. \tag{53}$$

In the following we will use as a basis of comparison the CDA-DDL and the NODE-DL systems, since these are the models that in practice perform better in the two classes of models, the client-depot assignment class and the node current class. It is possible to find solutions that are feasible for the linear programming relaxation of the CDA-DDL model that violate constraints of the NODE-DL formulation (rewritten with the v variables by using (53)). Nevertheless, we can provide an intuition on the nature of the relationship between the two classes of formulations.

Observe that if we add constraints (35) for every $d \in D$ and then use relationships (53), we obtain:

$$k_i + |D|\, x_{ij} + |D|\, x_{ji} \leq k_j + |D| \qquad \forall i, j \in C,\ i \neq j. \tag{54}$$

These inequalities are implied by the inequalities (24). To see this, observe that (54) is obtained by adding $x_{ij} + x_{ji} \leq 1$ to (24), which we already observed are implied by the NODE-DL system.

It is also interesting to enlarge this discussion to the inequalities associated to the arcs entering or leaving a depot. In fact, it could be easily shown that the client-depot assignment based systems imply the constraints (20)–(23). For simplicity we omit the "whole" formal proof of this statement but show how constraints (22) are implied. Observe that if we add constraints (31) $d-1$ times for client $i \in C$ and depot $d \in D$ and then add the constraint (33) for client i, we obtain the constraints $1 + (d-1)\, x_{ij} \leq d\, v_i^d + \sum_{d' \neq d} v_{d'}^i \leq \sum_{d' \in D} d'\, v_{d'}^i = k_i$, and as a consequence (22) are implied. Similar arguments can be used for the remaining constraints.

The point we want to make is that although the client-depot assignment based systems do not imply the node current systems, the former implies a weaker version of the latter.

For the purpose of the figure that will be shown in the next section, we denote as NODE-DL$^-$ the system of inequalities (20)–(23) and (54). Observe also that similar arguments could be used to compare the weaker CDA and NODE systems, however the "reversed" counterparts of (34), namely with x_{ji} instead of x_{ij}, would need to be added to CDA system given that the original node current based system proposed by Burger et al. (2018) used both constraints (25) and (26).

4.2.5 Summary

We conclude this section by presenting a visual summary of comparisons, as depicted in Figure 1, of the relationships established in the previous sections. In the figure, a vertical dashed line separates compact formulations that appear on the left hand side, from those on the right hand side that involve exponential sized sets of constraints. An arrow from a model A to a model B indicates that model B has at least as strong a linear programming relaxation value.

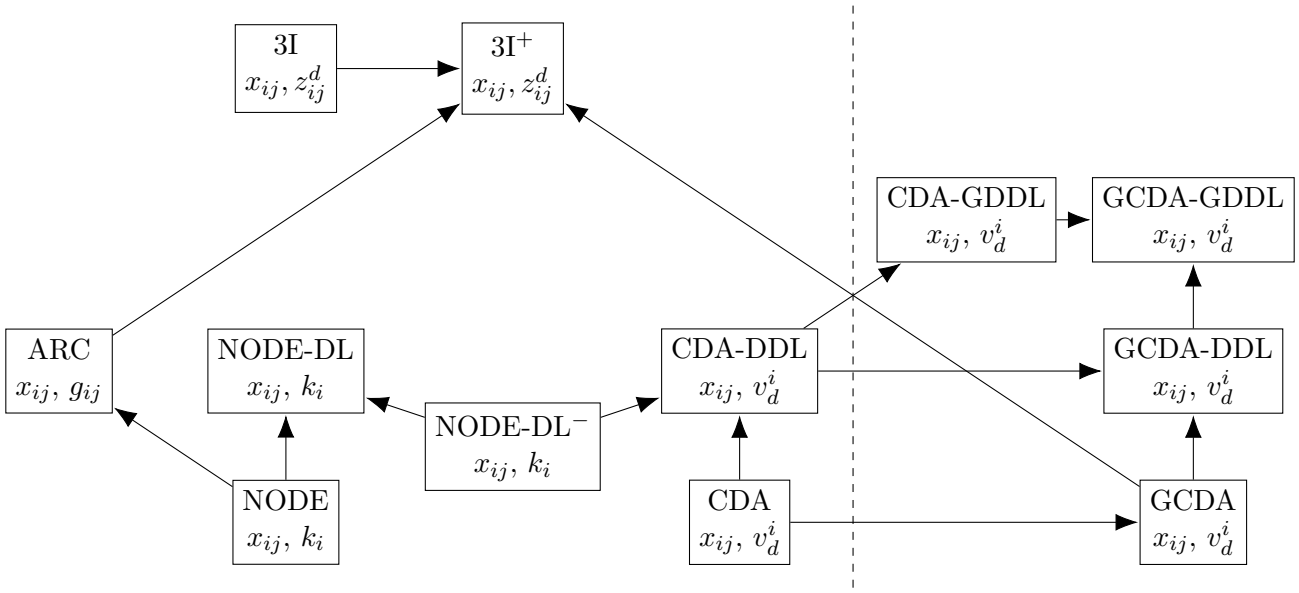


Figure 1: Established relationships between compact (to the left of the dashed line) and exponentially-sized (to the right of the dashed line) path elimination constraints.

5 Computational experiments

In this section we undertake computational experiments in order to computationally compare: (i) the linear programming relaxation bounds of the formulations presented earlier in the paper, and (ii) the time performance of the same formulations in solving the problem instances to optimality using an off-the-shelf optimizer. The experiments are intended to supplement the theoretical results, and whilst the comparisons conducted for item (i) above naturally complement the theoretical findings in terms of the strength of the linear programming bounds, this is not necessarily the case for the comparisons for item (ii). We provide further commentary on the latter in the concluding remarks. We start this section by first describing the test instances and the computational setting in Section 5.1, and then present the results in detail.

5.1 Instances and computational setting

The computational experiment uses a subset of the benchmark instances available at `http://prodhonc.free.fr/Instances/instances_us.htm` as well as a subset of the ones proposed by Bektaş et al. (2017). The former comprise symmetric cost instances with 100 clients and either 5 or 10 depots, whereas the latter are both symmetric and asymmetric cost instances with 100 clients and either 5, 10 or 20 depots.

Following the scheme proposed by Bektaş et al. (2017), we have generated two additional sets of instances, one with 50 clients and the other with 150 clients, similar to the sets with 100, 200 and 300 clients already used in the previous work, of which only the 100 client ones are used in this work. For clarity we explain the generation procedure for the two new sets of instances even though it is the same as the one used by Bektaş et al. (2017).

We start by generating symmetric cost instances, where node coordinates are generated by uniformly placing as many points as necessary in a 200×200 continuous grid, comprising 10 instances with 50 clients and 9 instances with 150 clients. For the former, we initially generate an instance with 50 clients and 10 depots. Then, we remove 5 depots from this instance to originate an instance with 50 clients and 5 depots. By repeating this process a four more times, we obtain a total of 50-client instances. Similarly, we generate three groups of three instances with 150 clients and either 10, 20 or 30 depots. All instances are named as bgs-$|C|$-$|D|$-t, where t denotes the instance number, similarly to the instances taken from Bektaş et al. (2017).

By transforming the symmetric instances we generate an equal number of asymmetric instances, differentiated in name by the added suffix "a". As explained in Bektaş et al. (2017), the transformation consists in increasing or decreasing the cost of an arc (i, j) by a random 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,

Table 1: F-MDRP Symmetric instances with 50 clients and 5 or 10 depots

	50-5			50-10		
Model	LP gap	Root gap	t	LP gap	Root gap	t
DL/ARC	5.50%	4.28%	26	4.85%	3.87%	30
DL/CDA-DDL	5.50%	4.39%	32	4.85%	3.87%	52
DL/NODE-DL	5.50%	4.70%	17	4.85%	4.01%	17
DL/NODE	5.50%	4.08%	21	4.85%	4.18%	21
DL/3I	5.50%	4.41%	218	4.85%	3.74%	604
SCF/ARC	16.28%	2.73%	21	12.65%	2.84%	21
SCF/CDA-DDL	**5.21%**	**2.25%**	31	**4.60%**	**2.30%**	45
SCF/NODE-DL	**5.21%**	2.41%	**9**	**4.60%**	2.35%	**9**
SCF/NODE	16.28%	2.58%	16	12.65%	2.91%	13
SCF/3I	16.28%	2.54%	119	12.65%	2.88%	259

*Indicated in boldface are the minimum values observed under each column.

and the cost of the arc in the opposite direction is decreased by the specified percentage.

The computational results reported here are taken with (and thus, may depend on it) a specific computational setting which we describe. The computer used in the tests has an Intel Core i7-4790 3.6GHz processor and 8GB of RAM. We used the commercial solver CPLEX version 12.9 with default settings, apart from specifying the use of a single thread, which we called using its C++ library.

5.2 Results

The results are shown in Tables 1–4 for instances with 50 clients, in Tables 5–8 for instances with 100 clients and in Tables 9–12 for instances with 150 clients. The first column of each table indicates the formulation tested, each named using the convention A/B, where A is the acronym for the set of subtour elimination constraints and B for the path elimination constraints that appear in the model. The remaining columns show average results across five or three instances, respectively for 50 client instances, and 100 and 150 client instances. In particular, for each formulation, the column titled 'LP gap' is the percentage gap between the LP relaxation value of the formulation and its optimal value. The former is obtained by solving a relaxed formulation where the integrality constraints on the binary variables are replaced with those that require the binary variables to be in the interval $[0, 1]$. In this case, there are no cuts added by the solver. The column titled 'Root gap' displays the percentage gap between the value of the LP relaxation after cuts have been added by the solver (i.e., before the first branching decision) and the optimal value. Column 't' is the computation time (in seconds) required to identify an optimal solution. A time limit of 3600 seconds was imposed on all instances, except for the large ones tested in Tables 9–12 which were run under a time limit of 10800 seconds. The minimum value(s) observed under each column of each of the tables is indicated in boldface.

5.2.1 Results for instances with 50 clients

The results with 50 clients show that the indication provided by the quality of the lower bound values does not necessarily correspond to the best model to be used to obtain the optimal solution. Observe that even though the linear programming relaxation gaps may differ substantially from model to model, the root node gaps for all models are quite similar, removing from the evaluation of the comparison the theoretical LP bound discussion. For instance, the SCF/ARC model for the symmetric 50 client instances has an average LP gap of 16.28% whereas the SCF/NODE-DL only has 5.21%. Nevertheless, the average root gap is 2.51% and 2.31%, respectively for these two models. Using the same two models as an example in the asymmetric instances, an average LP gap of 4.46% and 3.31% is observed, respectively. However the average root gaps exhibit a reversed relation, that is, for the SCF/ARC model it is of 2.42% and for the SCF/NODE-DL model of 2.67%.

Additionally, even though the formulations including path elimination constraints based on arc-depot assignment formulations have better theoretical linear programming relaxations values, these models are the slowest at obtaining the optimal solutions. For this reason, we will not be providing results for these two formulations subsequently.

Table 2: F-MDRP Asymmetric instances with 50 clients and 5 or 10 depots

	50-5a			50-10a		
Model	LP gap	Root gap	t	LP gap	Root gap	t
DL/ARC	3.37%	2.23%	7	2.77%	1.56%	9
DL/CDA-DDL	3.38%	2.08%	8	2.39%	1.60%	9
DL/NODE-DL	3.41%	1.82%	**3**	3.42%	1.64%	**2**
DL/NODE	3.42%	1.86%	4	3.43%	1.73%	4
DL/3I	3.33%	**1.65%**	38	**1.67%**	**1.13%**	50
SCF/ARC	4.43%	2.44%	8	3.88%	1.75%	12
SCF/CDA-DDL	**3.23%**	2.31%	19	2.37%	1.64%	28
SCF/NODE-DL	3.28%	2.47%	6	3.40%	1.72%	6
SCF/NODE	4.46%	2.56%	7	4.49%	1.90%	6
SCF/3I	4.24%	2.42%	69	2.83%	1.24%	23

Table 3: L-MDRP Symmetric instances with 50 clients and 5 or 10 depots

	50-5			50-10		
Model	LP gap	Root gap	t	LP gap	Root gap	t
DL/ARC	6.82%	5.28%	51	7.23%	5.41%	57
DL/CDA-DDL	6.82%	5.18%	107	7.23%	5.22%	151
DL/NODE-DL	6.82%	5.55%	41	7.23%	5.32%	38
DL/NODE	6.82%	5.23%	52	7.23%	5.14%	40
DL/3I	6.82%	5.21%	557	7.23%	4.91%	931
SCF/ARC	17.84%	2.77%	29	18.37%	3.12%	24
SCF/CDA-DDL	**6.25%**	2.48%	29	**6.90%**	2.83%	69
SCF/NODE-DL	**6.25%**	**2.35%**	**12**	**6.90%**	**2.73%**	**14**
SCF/NODE	17.84%	2.75%	20	18.37%	3.19%	24
SCF/3I	17.84%	2.57%	116	18.37%	2.96%	207

Table 4: L-MDRP Asymmetric instances with 50 clients and 5 or 10 depots

	50-5a			50-10a		
Model	LP gap	Root gap	t	LP gap	Root gap	t
DL/ARC	5.45%	2.96%	14	5.03%	2.60%	12
DL/CDA-DDL	5.45%	2.88%	16	5.03%	**2.48%**	31
DL/NODE-DL	5.45%	**2.75%**	**10**	5.03%	2.87%	11
DL/NODE	5.45%	2.94%	12	5.03%	2.54%	**7**
DL/3I	5.45%	2.90%	159	5.03%	2.74%	520
SCF/ARC	6.85%	3.17%	14	6.50%	2.84%	18
SCF/CDA-DDL	**4.93%**	3.00%	30	**4.72%**	2.77%	61
SCF/NODE-DL	**4.93%**	3.01%	13	**4.72%**	2.82%	20
SCF/NODE	6.85%	3.07%	12	6.51%	2.79%	16
SCF/3I	6.85%	3.04%	91	6.50%	2.73%	532

Table 5: F-MDRP Symmetric instances with 100 clients and 5, 10 or 20 depots

	100-5			100-10			100-20		
Model	LP gap	Root gap	t	LP gap	Root gap	t	LP gap	Root gap	t
DL/ARC	6.09%	4.88%	3094	6.72%	6.02%	3599	6.75%	5.90%	3599
DL/CDA-DDL	6.09%	5.21%	2408	6.72%	6.11%	3598	6.75%	6.02%	3595
DL/NODE-DL	6.09%	5.33%	2144	6.72%	6.06%	3299	6.75%	5.83%	3569
DL/NODE	6.09%	5.04%	2468	6.72%	6.30%	3523	6.75%	6.05%	3116
SCF/ARC	19.25%	4.47%	**393**	18.53%	4.63%	1490	16.28%	4.76%	2555
SCF/CDA-DDL	**5.85%**	3.68%	1580	**6.53%**	**4.11%**	2416	**6.57%**	**4.18%**	3115
SCF/NODE-DL	**5.85%**	**3.41%**	703	**6.53%**	4.18%	**1260**	**6.57%**	4.24%	**1703**
SCF/NODE	19.25%	3.91%	606	18.53%	4.61%	2020	16.28%	4.98%	2393

Table 6: F-MDRP Asymmetric instances with 100 clients and 5, 10 or 20 depots

	100-5a			100-10a			100-20a		
Model	LP gap	Root gap	t	LP gap	Root gap	t	LP gap	Root gap	t
DL/ARC	3.73%	**2.31%**	654	3.78%	2.44%	898	4.26%	2.05%	1949
DL/CDA-DDL	3.73%	2.42%	781	3.45%	**2.00%**	1332	2.92%	1.83%	1017
DL/NODE-DL	3.73%	2.48%	**191**	3.96%	2.39%	335	4.61%	**2.03%**	**490**
DL/NODE	3.73%	2.47%	203	3.96%	2.09%	**306**	4.61%	2.15%	668
SCF/ARC	6.03%	3.08%	421	5.77%	2.71%	1392	5.92%	2.66%	1238
SCF/CDA-DDL	**3.66%**	2.99%	663	**3.40%**	2.64%	2197	**2.89%**	2.28%	2051
SCF/NODE-DL	**3.66%**	3.00%	200	3.91%	2.82%	359	4.57%	2.51%	576
SCF/NODE	6.03%	3.03%	356	5.84%	2.71%	533	6.27%	2.51%	514

Observe also that, because some path elimination constraint systems imply the two-client subtour elimination constraints (19) whereas others do not, the LP values obtained by models with weaker path elimination constraints may still provide an overall better LP bound. This is observed in the SCF/CDA-DDL or SCF/NODE-DL models when compared to all other models with the SCF formulation as subtour elimination constraints. This is not observed when the DL inequalities are used as subtour elimination constraints since these already imply constraints (19).

When comparing results from instances with a different number of depots, 5 and 10 depots for the whole class of 50 client instances, we notice two different behaviours. The arc and node current based models take the same computational time to reach the optimal solution. However, the client-depot and arc-depot assignment models can take much longer solving instances with 10 depots than instances with 5 depots. This is expected since the size of the latter two models depends directly on the number of depots, whereas in the former two models this is not the case.

As for the (lifted) NODE-DL model when compared to the NODE model, we see that the difference is not substantial, however, the former appears to be better on average. A more precise evaluation will be given in the next subsection, when describing the results for the 100 client instances. Finally, the conclusions stated above hold for the F-MDRP and L-MDRP, although the L-MDRP instances take longer to solve to optimality, on average, and the LP gaps are in general worse, regardless of the model used.

5.2.2 Results for instances with 100 clients

Most of the observations on the 50 client instances hold on the 100 client instances. The average computational times show that the client-depot assignment systems are unable to solve most of the 100 client instances and for this reason they were not run for the 150 client instances. The NODE based models are also worse than the NODE-DL based models, something which was not clear in the 50 client instances, and so we also omit them from the results taken from the 150 client instances.

These results give us a much better view of the models which work better in practice. It is clear that, in terms of path elimination constraints, the arc and node current based models are better than the LP stronger depot assignment based models. Additionally, it appears that, on average, the NODE-DL system is the best overall,

Table 7: L-MDRP Symmetric instances with 100 clients and 5, 10 or 20 depots

	100-5			100-10			100-20		
Model	LP gap	Root gap	t	LP gap	Root gap	t	LP gap	Root gap	t
DL/ARC	6.79%	5.81%	3597	7.23%	6.08%	3287	7.91%	6.63%	2618
DL/CDA-DDL	6.79%	5.91%	3578	7.23%	6.01%	3599	7.91%	6.85%	3598
DL/NODE-DL	6.79%	6.04%	3156	7.23%	6.38%	2733	7.91%	6.79%	2527
DL/NODE	6.79%	5.85%	3238	7.23%	6.05%	3591	7.91%	6.55%	2553
SCF/ARC	20.59%	4.55%	1621	20.87%	4.78%	1361	21.46%	5.58%	1190
SCF/CDA-DDL	**6.40%**	4.04%	2697	**6.97%**	4.06%	3360	**7.69%**	5.05%	3600
SCF/NODE-DL	**6.40%**	**3.90%**	**1114**	**6.97%**	**4.05%**	**459**	**7.69%**	**4.71%**	1190
SCF/NODE	20.59%	4.85%	1736	20.87%	4.77%	1465	21.46%	5.60%	**870**

Table 8: L-MDRP Asymmetric instances with 100 clients and 5, 10 or 20 depots

	100-5a			100-10a			100-20a		
Model	LP gap	Root gap	t	LP gap	Root gap	t	LP gap	Root gap	t
DL/ARC	3.73%	**2.06%**	**126**	4.00%	**2.04%**	496	3.45%	1.33%	249
DL/CDA-DDL	3.73%	2.42%	700	4.00%	2.08%	1041	3.42%	1.34%	793
DL/NODE-DL	3.73%	2.16%	155	4.00%	2.24%	282	3.45%	**1.24%**	**44**
DL/NODE	3.73%	2.17%	245	4.00%	2.06%	**149**	3.45%	1.30%	45
SCF/ARC	6.21%	3.00%	333	6.49%	3.19%	459	5.47%	2.50%	340
SCF/CDA-DDL	**3.58%**	2.70%	999	**3.90%**	3.03%	3185	**3.36%**	2.49%	2386
SCF/NODE-DL	**3.58%**	2.73%	515	**3.90%**	3.03%	1056	3.39%	2.43%	340
SCF/NODE	6.21%	2.81%	1041	6.49%	3.15%	729	5.48%	2.48%	488

since it is only outperformed by the ARC system in the L-MDRP variant for asymmetric instances with 10 depots. Nevertheless, a much stronger and conclusive analysis will be made in the next section with respect to the results for the 150 client instances.

Regarding subtour elimination constraints, the larger 100 client instance set puts in evidence a different behaviour between using the SCF or the DL models as subtour elimination constraints. For the symmetric instances, the DL based models perform worse than the SCF based models. For instance, the DL/ARC model has an average computational time of 3094 seconds in the 100 client and 5 depot F-MDRP symmetric instances, whereas the SCF/ARC only has 393 seconds. However, for asymmetric instances, the DL based formulations are much more competitive and even outperform the SCF formulations in some cases. We will omit the DL based models for symmetric instances in the 150 client instances.

5.2.3 Results for instances with 150 clients

Most of the instances with 150 clients were unable to be optimally solved within the time limits imposed. For this reason, the results presented in this section in Tables 9–12 include an additional column titled 'Final gap' that shows the average gap between the final upper and lower bounds at the time of termination.

The results suggest that, with respect to path elimination constraints, the node current system is overall faster than the arc current system in identifying optimal solutions. In particular, the latter was only faster on average in the L-MDRP instances with 20 depots, both symmetric and asymmetric. Once again, even though the node current system has weaker linear programming relaxation bounds than the arc current system for symmetric cost

Table 9: F-MDRP Symmetric instances with 150 clients and 10, 20 or 30 depots

	150-10				150-20				150-30			
Model	LP gap	Root gap	Final gap	t	LP gap	Root gap	Final gap	t	LP gap	Root gap	Final gap	t
SCF/ARC	16.40%	4.02%	**1.40%**	10633	14.67%	3.57%	5.74%	10799	14.65%	3.21%	7.18%	10799
SCF/NODE-DL	**5.81%**	**3.43%**	2.31%	**5570**	**4.66%**	**2.76%**	**0.58%**	**9674**	**4.63%**	**2.58%**	**0.16%**	**10003**

Table 10: F-MDRP Asymmetric instances with 150 clients and 10, 20 or 30 depots

	150-10a				150-20a				150-30a			
Model	LP gap	Root gap	Final gap	t	LP gap	Root gap	Final gap	t	LP gap	Root gap	Final gap	t
DL/ARC	**2.65%**	1.56%	0.24%	5408	**2.71%**	1.71%	0.41%	5509	**3.20%**	**1.84%**	0.47%	5999
DL/NODE-DL	2.71%	**1.47%**	0.00%	**1808**	2.90%	**1.66%**	0.41%	**3768**	3.71%	2.09%	0.00%	**1862**
SCF/ARC	4.57%	1.79%	0.00%	4547	4.52%	1.84%	16.03%	7730	4.70%	1.86%	1.83%	5558
SCF/NODE-DL	2.69%	1.75%	0.00%	3837	2.89%	1.90%	**0.23%**	4318	3.71%	2.18%	0.00%	2989

Table 11: L-MDRP Symmetric instances with 150 clients and 10, 20 or 30 depots

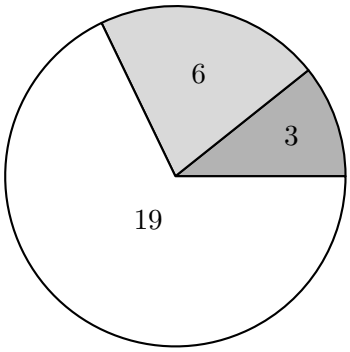
	150-10				150-20				150-30			
Model	LP gap	Root gap	Final gap	t	LP gap	Root gap	Final gap	t	LP gap	Root gap	Final gap	t
SCF/ARC	19.01%	5.53%	3.87%	10799	18.60%	4.56%	**0.48%**	**7169**	19.26%	4.95%	**0.94%**	10799
SCF/NODE-DL	**7.54%**	**4.81%**	**2.29%**	**10596**	**6.42%**	**3.54%**	0.71%	9803	**6.98%**	**3.73%**	1.41%	10799

instances, this difference becomes much less significant as far as the the root gap is concerned.

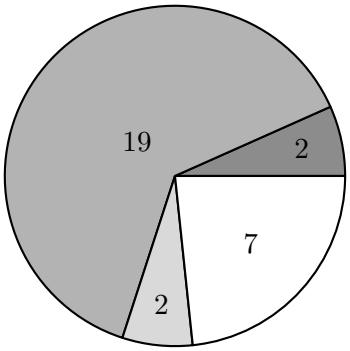
We present an aggregate summary of the results for the 28 instances tested in the subsections above in Figure 2, where each pie chart shows the number of instances for which the corresponding model was faster, or yielded the smallest final gap in case of instances not optimally solved within the time limit. Ties were counted twice. The figure indicates that the models using NODE-DL for path elimination are better in most cases. Models that use SCF for subtour elimination are better for symmetric instances, whereas those using DL are better for asymmetric instances.

Table 12: L-MDRP Asymmetric instances with 150 clients and 10, 20 or 30 depots

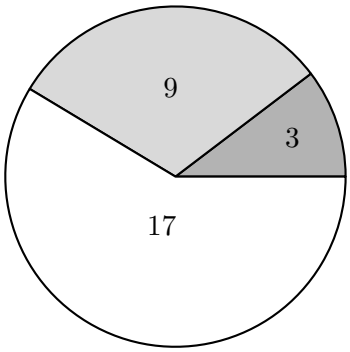
	150-10a				150-20a				150-30a			
Model	LP gap	Root gap	Final gap	t	LP gap	Root gap	Final gap	t	LP gap	Root gap	Final gap	t
DL/ARC	3.04%	1.91%	0.00%	2904	2.61%	1.50%	0.00%	1603	2.72%	1.46%	0.00%	1471
DL/NODE-DL	3.04%	**1.87%**	0.00%	**957**	2.61%	**1.45%**	0.00%	**466**	2.72%	**1.34%**	0.00%	**627**
SCF/ARC	5.08%	2.26%	0.59%	6062	4.75%	2.03%	0.00%	1790	4.91%	2.05%	0.00%	4387
SCF/NODE-DL	**2.92%**	2.19%	0.00%	4951	**2.58%**	1.94%	0.23%	5907	**2.70%**	1.97%	0.00%	4107



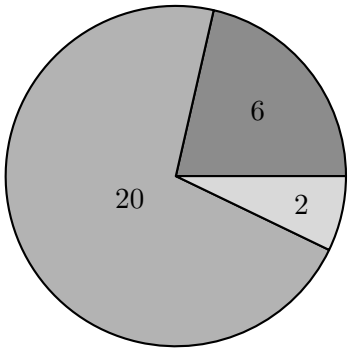
(a) F-MDRP symmetric.



(b) F-MDRP asymmetric.



(c) L-MDRP symmetric.



(d) L-MDRP asymmetric.

Figure 2: Aggregate statistics for all the tested instances in the paper.

6 Extensions to a more general case

In this section we discuss the extensions of the models and results above to a more general case of the MDRP, in which each client needs to be supplied with a given amount of commodity using capacitated vehicles and where there exist multiple vehicles ready to be dispatched at each depot to supply the customers with their requests.

We first recall that the models introduced earlier in the paper are based on two separate and independent generic sets of constraints, namely constraints (6) for subtour elimination and constraints (7) for path elimination, the latter being the main focus of this paper. We also observe that the theoretical results of Section 4.2 do not depend on the constraints used for subtour elimination, and would not change if subtour elimination constraints are extended to model additional requirements.

We now outline the changes needed to model the more general case of the MDRP:

1. For capacitated instances, one way to model the capacity restrictions is by solely modifying the subtour elimination constraints that were used to model (6). For illustrative purposes, consider the capacitated MDRP where each vehicle has a maximum capacity of Q and each client has a unit demand. The changes made in the formulations presented in the paper are simply to replace $|C|$ with Q in constraints (13), (14) and (17), or replace $|C| - |D| + 1$ with Q in constraints (15), (16) and (18). Modifications required for non-unit demands are also straightforward.

2. For the extension with k_d vehicles at each depot $d \in D$, the change required is to redefine the y variables as $y_d \in \{0, 1, 2, \ldots, k_d\}$ for the L-MDRP, and as $y_d \in \{1, 2, \ldots, k_d\}$ for the F-MDRP, as explained in Bektaş

et al. (2017).

In either of the two cases above, the theoretical results presented in Section 4.2 concerning the comparison of different sets of path elimination constraints and the relationships between them as shown in Figure 1 will still hold. In fact, the results will apply to formulations of the MDRP following the structure (1)-(10) and that have the same set of constraints modelling subtour elimination.

To serve as a preliminary investigation on the computational performance of the models, we have conducted experiments with a simpler variant of the more general case of the MDRP where each client has a unit demand, and where a single vehicle exists in each depot with vehicle capacities shown with Q. We tested four formulations, each of which uses either the capacitated version of the Desrochers and Laporte (DL-CAP) or capacitated Single Commodity Flow (SCF-CAP) constraints for subtour elimination, and either the arc current (ARC) or node current (NODE-DL) formulations for path elimination. The results on instances with 25 clients and five depots, and with $Q = 6, 8$ and 10 suggest that the formulations SCF-CAP/ARC and SCF-CAP/NODE-DL consistently outperform either DL-CAP/ARC or DL-CAP/NODE-DL in terms of LP gap, Root gap, Final gap and computational time, on both the F-MDRP and the L-MDRP. Moreover, SCF-CAP/NODE-DL is almost always faster than SCF-CAP/ARC. Comparing the latter two on instances with 50 clients, five depots and with $Q = 12, 16$ and 20 yields similar findings. Further details are given in the Appendix.

The main findings from the results is that the problem becomes more difficult to solve, as expected, but the main trend observed in the uncapacitated version of the problem does not change. Whilst it is inevitably difficult to arrive at a general conclusion about the computational tractability of the MDRP in general (as this will depend on the underlying parameters of the instances tested) the results above suggest that asymmetric instances of up to 50 clients, and more so those that allow larger capacity limits, are within the reach of the compact formulations described in this paper.

To conclude, we emphasize that the theoretical relations between the models are kept unchanged only because the extra requirements are not being modelled as part of the path elimination constraints. Otherwise, the relationship of the different models shown in Figure 1 may no longer hold. As an example, we observe that the capacity restrictions can also be included in the path elimination part of the models, for instance, by limiting the number of clients assigned to a depot in the formulations of Section 4.1.3. The interplay between all of the possible ways of including the capacity restrictions, either in the subtour elimination constraints or in the path elimination constraints or in both, requires further analysis including new theoretical comparisons.

7 Concluding remarks

This paper presented compact formulations for several variants of the multi-depot routing problem, including improvements of those that have previously been proposed, and other formulations that are new. Computational testing suggested that, on the instances tested in this paper with up to 150 clients and 30 depots, the NODE-DL formulation has generally performed better in terms of time-to-optimality, which makes it preferable in solving other instances of the multi-depot routing problems described in the paper. The same formulation, however, has not been able to produce results on larger-scale instances due to excessive requirements on computational memory. In comparison, the NODE-SCF formulation has resulted in a comparatively smaller number of branch-and-bound nodes on average. Moreover, the fact such formulations based on arc flows are more amenable to decomposition would make the NODE-SCF formulation more attractive in devising tailored methods for solving larger-scale multi-depot routing problem instances, which offers one avenue for further research.

The theoretical and computational comparisons undertaken here have shown that the findings from the theoretical analysis are not necessarily aligned with the computational results. In particular, although the computational results support the theoretical findings in terms of the relative strength of the linear programming relaxation bounds of the formulations, there are also cases where this strength does not necessarily indicate superiority in terms of solution time to optimality. In other words, a formulation with a stronger theoretical LP relaxation bound (e.g., the $3I^+$ formulation) is not necessarily faster, in comparison to one with a weaker LP relaxation bound (e.g., NODE-DL or ARC formulations), in optimally solving instances in practice. The reasons behind this phenomenon are not just attributable to the number of variables and constraints of the formulations, and can further be affected by the particular features of modern solvers, such as automatic cut generation features built within such solvers, the details of which are often not fully known to the user. Our observations indicate that, some of the formulations presented here that have very different linear programming relaxation bounds still yield

the same, or very similar, root relaxation bounds when processed by the same solver, a phenomenon that has also been reported in the computational testing of integer programming formulations of similar problems (see, e.g. Gadegaard & Lysgaard 2020). In addition, the computational performance of a formulation will also be highly dependent on the settings of the solver used, although for our comparison purposes the default values have been used throughout.

The models described here can be extended to formulate a more general case of the MDRP with capacitated vehicles and/or with several vehicles at each depot. In either case, the theoretical results presented on the relationship between the different path elimination constraints do not change, in the way that they were extended here. The computational experience with the extended formulations suggests that the models are able to solve unit-demand capacitated MDRP instances of up to 50 clients to optimality for both the L-MDRP and the F-MDRP.

Whilst this work has exclusively focused on the development and comparison of compact formulations for multi-depot routing problems, it has also identified exponential-size set of inequalities that can be used for eliminating paths or subtours, or both. The study of such inequalities and the relevant separation problems, along with testing their computational performance against the state-of-the-art in solving multi-depot routing problems represents yet another avenue for further research.

Acknowledgements

The authors thank the two anonymous reviewers for useful comments that have led to an improvement in the paper and for suggesting the additional analysis conducted on the capacitated variant of the problem.

References

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.

Applegate, D. L., Bixby, R. E., Chvátal, V. & Cook, W. J. (2006), *The traveling salesman problem: a computational study*, Princeton University Press.

Bektaş, T. (2012), 'Formulations and Benders decomposition algorithms for multidepot salesmen problems with load balancing', *European Journal of Operational Research* **216**(1), 83–93.

Bektaş, T., Gouveia, L. & Santos, D. (2017), 'New path elimination constraints for multi-depot routing problems', *Networks* **70**(3), 246–261.

Bektaş, T., Gouveia, L. & Santos, D. (2019), 'Revisiting the Hamiltonian p-median problem: a new formulation on directed graphs and a branch-and-cut algorithm', *European Journal of Operational Research* **276**(1), 40–64.

Belenguer, J.-M., Benavent, E., Prins, C., Prodhon, C. & Wolfler Calvo, R. (2011), 'A branch-and-cut method for the capacitated location-routing problem', *Computers & Operations Research* **38**(6), 931–941.

Benati, S., Puerto, J. & Rodríguez-Chía, A. M. (2017), 'Clustering data that are graph connected', *European Journal of Operational Research* **261**(1), 43–53.

Benavent, E. & Martínez-Sykora, A. (2013), 'Multi-depot multiple TSP: a polyhedral study and computational results', *Annals of Operations Research* **207**(1), 7–25.

Burger, M., Su, Z. & De Schutter, B. (2018), 'A node current-based 2-index formulation for the fixed-destination multi-depot travelling salesman problem', *European Journal of Operational Research* **265**(2), 463–477.

Desrochers, M. & Laporte, G. (1991), 'Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints', *Operations Research Letters* **10**(1), 27–36.

Erdoğan, G., Laporte, G. & Rodríguez-Chía, A. M. (2016), 'Exact and heuristic algorithms for the Hamiltonian p-median problem', *European Journal of Operational Research* **253**(2), 280–289.

Fernández, E. & Rodríguez-Pereira, J. (2016), 'Multi-depot rural postman problems', *TOP* pp. 1–33.

Gadegaard, S. & Lysgaard, J. (2020), 'A symmetry-free polynomial formulation of the capacitated vehicle routing problem', *Discrete Applied Mathematics* (In press).

Gavish, B. & Graves, S. C. (1978), 'The travelling salesman problem and related problems', *Working paper OR-078-78, Operations Research Center Working Papers, Massachusetts Institute of Technology, Operations Research Center* .

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, pp. 223—-254.

Gollowitzer, S., Gouveia, L., Laporte, G., Pereira, D. L. & Wojciechowski, A. (2014), 'A comparison of several models for the Hamiltonian p-median problem', *Networks* **63**(4), 350–363.

Gouveia, L. & Pesneau, P. (2006), 'On extended formulations for the precedence constrained asymmetric traveling salesman problem', *Networks* **48**(2), 77–89.

Gouveia, L., Pesneau, P., Ruthmair, M. & Santos, D. (2018), 'Combining and projecting flow models for the (precedence constrained) asymmetric traveling salesman problem', *Networks* **71**(4), 451–465.

Gouveia, L. & Pires, J. M. (1999), 'The asymmetric travelling salesman problem and a reformulation of the Miller-Tucker-Zemlin constraints', *European Journal of Operational Research* **112**(1), 134–146.

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.

Lawler, E. L., Lenstra, J. K., Kan, A. H. G. R. & Shmoys, D. B. (1985), *The traveling salesman problem: a guided tour of combinatorial optimization*, John Wiley & Sons.

Miller, C. E., Tucker, A. W. & Zemlin, R. A. (1960), 'Integer programming formulation of traveling salesman problems', *Journal of the Association for Computing Machinery* **7**(4), 326–329.

Öncan, T., Altınel, 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.

Toth, P. & Vigo, D. (2014), *Vehicle routing: problems, methods, and applications*, Society for Industrial and Applied Mathematics.

A Results for the capacitated unit demand variants

The Appendix presents the preliminary results obtained for the variants of the F-MDRP and L-MDRP with a single capacitated vehicle per depot with capacity Q and unit demands for every client. For this purpose, we used the 50 client and five depot instances, as well as new ones with 25 clients and five depots which were generated using a procedure similar to the one explained in Section 5.1. Tables 13 and 14 show the results for the 25 client and five depot instances for the F-MDRP and the L-MDRP, respectively, and Tables 15 and 16 show the results for the 50 client and five depot instances. All tables follow the same format as previous ones, with the addition of a column indicating the value of Q.

Table 13: F-MDRP Instances with 25 clients and 5 depots

		25-5				25-5a			
Model	Q	LP gap	Root gap	Final gap	t	LP gap	Root gap	Final gap	t
DL-CAP/ARC	6	19.77%	18.51%	5.12%	2761	9.14%	7.79%	**0.00%**	85
DL-CAP/NODE-DL		19.77%	18.30%	4.89%	2752	9.29%	7.93%	**0.00%**	56
SCF-CAP/ARC		18.29%	9.23%	**0.00%**	759	9.38%	**6.38%**	**0.00%**	37
SCF-CAP/NODE-DL		**11.93%**	**8.86%**	**0.00%**	**664**	**8.01%**	6.49%	**0.00%**	**25**
DL-CAP/ARC	8	11.98%	9.47%	1.86%	756	4.79%	3.58%	**0.00%**	3
DL-CAP/NODE-DL		11.98%	10.90%	1.55%	767	4.96%	3.48%	**0.00%**	**2**
SCF-CAP/ARC		17.44%	5.73%	**0.00%**	324	6.54%	**3.17%**	**0.00%**	5
SCF-CAP/NODE-DL		**8.57%**	**5.53%**	**0.00%**	**121**	**4.64%**	3.50%	**0.00%**	**2**
DL-CAP/ARC	10	7.88%	5.29%	**0.00%**	14	**4.00%**	2.67%	**0.00%**	**1**
DL-CAP/NODE-DL		7.88%	6.80%	**0.00%**	8	4.18%	2.81%	**0.00%**	**1**
SCF-CAP/ARC		16.40%	3.01%	**0.00%**	**3**	6.01%	**2.61%**	**0.00%**	**1**
SCF-CAP/NODE-DL		**5.81%**	**2.95%**	**0.00%**	**3**	4.04%	2.80%	**0.00%**	**1**

Table 14: L-MDRP Instances with 25 clients and 5 depots

		25-5				25-5a			
Model	Q	LP gap	Root gap	Final gap	t	LP gap	Root gap	Final gap	t
DL-CAP/ARC	6	28.76%	27.38%	11.27%	3599	23.59%	21.98%	4.38%	2380
DL-CAP/NODE-DL		28.76%	26.90%	12.00%	3599	23.59%	21.96%	4.79%	2839
SCF-CAP/ARC		19.82%	8.89%	**0.00%**	1048	13.46%	**6.48%**	**0.00%**	40
SCF-CAP/NODE-DL		**13.70%**	**8.87%**	**0.00%**	**701**	**11.93%**	6.60%	**0.00%**	**26**
DL-CAP/ARC	8	20.05%	18.37%	2.63%	1238	14.30%	12.27%	**0.00%**	634
DL-CAP/NODE-DL		20.05%	18.72%	2.61%	1026	14.30%	12.56%	1.53%	789
SCF-CAP/ARC		20.12%	6.74%	**0.00%**	764	9.41%	**3.41%**	**0.00%**	11
SCF-CAP/NODE-DL		**11.72%**	**6.43%**	**0.00%**	**297**	**7.07%**	3.48%	**0.00%**	**8**
DL-CAP/ARC	10	16.27%	14.36%	**0.00%**	390	12.12%	10.14%	**0.00%**	337
DL-CAP/NODE-DL		16.27%	14.16%	**0.00%**	574	12.12%	10.05%	**0.00%**	198
SCF-CAP/ARC		21.27%	6.12%	**0.00%**	12	9.94%	**4.43%**	**0.00%**	**5**
SCF-CAP/NODE-DL		**10.90%**	**6.10%**	**0.00%**	**11**	**7.46%**	4.48%	**0.00%**	8

Table 15: F-MDRP Instances with 50 clients and 5 depots

		50-5				50-5a			
Model	Q	LP gap	Root gap	Final gap	t	LP gap	Root gap	Final gap	t
SCF-CAP/ARC	12	18.08%	**9.09%**	6.83%	10798	7.66%	5.19%	**0.00%**	2959
SCF-CAP/NODE-DL		**12.75%**	9.10%	**5.56%**	10799	**6.79%**	**5.17%**	**0.00%**	**1742**
SCF-CAP/ARC	16	16.41%	7.52%	**1.89%**	10335	6.22%	4.21%	**0.00%**	297
SCF-CAP/NODE-DL		**9.40%**	**7.22%**	2.50%	**10102**	**5.22%**	**4.20%**	**0.00%**	**272**
SCF-CAP/ARC	20	17.02%	6.86%	**0.79%**	7501	5.24%	3.21%	**0.00%**	69
SCF-CAP/NODE-DL		**8.97%**	**6.62%**	1.21%	**6164**	**4.12%**	**3.13%**	**0.00%**	**68**

Table 16: L-MDRP Instances with 50 clients and 5 depots

		50-5				50-5a			
Model	Q	LP gap	Root gap	Final gap	t	LP gap	Root gap	Final gap	t
SCF-CAP/ARC	12	18.94%	9.28%	7.27%	10799	8.96%	**5.30%**	**0.00%**	2905
SCF-CAP/NODE-DL		**13.57%**	**9.15%**	**6.06%**	**10795**	**8.08%**	5.34%	**0.00%**	**2232**
SCF-CAP/ARC	16	18.10%	7.61%	**2.14%**	10798	8.10%	**4.71%**	**0.00%**	698
SCF-CAP/NODE-DL		**10.94%**	**7.33%**	2.61%	**10743**	**6.84%**	**4.71%**	**0.00%**	**421**
SCF-CAP/ARC	20	18.73%	7.01%	0.90%	**7166**	7.65%	4.14%	**0.00%**	196
SCF-CAP/NODE-DL		**10.44%**	**6.90%**	**0.72%**	8782	**6.09%**	**4.06%**	**0.00%**	**72**