University of Groningen

Asymmetric Multi-depot Vehicle Routing Problems

uit het Broek, Michiel; Schrotenboer, Albert; Jargalsaikhan, Bolor; Roodbergen, K.J.; Callegari Coelho, Leandro

Published in:
Operations Research

DOI:
[10.1287/opre.2020.2033](https://doi.org/10.1287/opre.2020.2033)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2021

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

uit het Broek, M., Schrotenboer, A., Jargalsaikhan, B., Roodbergen, K. J., & Callegari Coelho, L. (2021). Asymmetric Multi-depot Vehicle Routing Problems: Valid Inequalities and a Branch-and-cut Algorithm. *Operations Research*, 69(2), 380-409. <https://doi.org/10.1287/opre.2020.2033>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Operations Research

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

Asymmetric Multidepot Vehicle Routing Problems: Valid Inequalities and a Branch-and-Cut Algorithm

Michiel A. J. uit het Broek, Albert H. Schrotenboer, Bolor Jargalsaikhan, Kees Jan Roodbergen, Leandro C. Coelho

To cite this article:

Michiel A. J. uit het Broek, Albert H. Schrotenboer, Bolor Jargalsaikhan, Kees Jan Roodbergen, Leandro C. Coelho (2021) Asymmetric Multidepot Vehicle Routing Problems: Valid Inequalities and a Branch-and-Cut Algorithm. *Operations Research* 69(2):380-409. <https://doi.org/10.1287/opre.2020.2033>

Full terms and conditions of use: <https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions>

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2021, INFORMS

Please scroll down for article—it is on subsequent pages

With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes.

For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

Contextual Areas

Asymmetric Multidepot Vehicle Routing Problems: Valid Inequalities and a Branch-and-Cut Algorithm

Michiel A. J. uit het Broek,^a Albert H. Schrotenboer,^a Bolor Jargalsaikhan,^a Kees Jan Roodbergen,^a Leandro C. Coelho^{a,b,c}

^a Department of Operations, Faculty of Economics and Business, University of Groningen, 9712 CP Groningen, Netherlands; ^b CIRRELT and Université Laval, Québec, Québec G1V 0A6, Canada; ^c Research Chair in Integrated Logistics, Université Laval, Québec, Québec G1V 0A6, Canada

Contact: a.j.uit.het.broek@rug.nl, <https://orcid.org/0000-0002-7567-6385> (MAJB); a.h.schrotenboer@rug.nl,

<https://orcid.org/0000-0001-8365-2204> (AHS); b.jargalsaikhan@rug.nl (BJ); k.j.roodbergen@rug.nl,

<https://orcid.org/0000-0002-4083-2106> (KJR); leandro.coelho@cirrelt.ca, <https://orcid.org/0000-0002-9797-1019> (LCC)

Received: July 17, 2019

Accepted: March 1, 2020

Published Online in Articles in Advance:
February 16, 2021

Area of Review: Transportation

<https://doi.org/10.1287/opre.2020.2033>

Copyright: © 2021 INFORMS

Abstract. We present a generic branch-and-cut framework for solving routing problems with multiple depots and asymmetric cost structures, which determines a set of cost-minimizing (capacitated) vehicle tours that fulfill a set of customer demands. The backbone of the framework is a series of valid inequalities with corresponding separation algorithms that exploit the asymmetric cost structure in directed graphs. We derive three new classes of so-called D_k inequalities that eliminate subtours, enforce tours to be linked to a single depot, and impose bounds on the number of customers in a tour. In addition, other well-known valid inequalities for solving vehicle routing problems are generalized and adapted to be valid for routing problems with multiple depots and asymmetric cost structures. The framework is tested on four specific problem variants, for which we develop a new set of large-scale benchmark instances. The results show that the new inequalities can reduce root-node optimality gaps by up to 67.2% compared with existing approaches. Moreover, the complete framework is very effective and can solve instances of considerably larger size than reported in the literature. For instance, it solves asymmetric multidepot traveling-salesman-problem instances containing up to 400 customers and 50 depots, whereas to date only solutions of instances containing up to 300 customer nodes and 60 depots have been reported.

Funding: This work was supported by Nederlandse Organisatie voor Wetenschappelijk Onderzoek [Grants 438-13-216, 439-16-612] and Natural Sciences and Engineering Research Council of Canada [Grant 2014-05764].

Supplemental Material: Data are available at <https://doi.org/10.1287/opre.2020.2033>.

Keywords: branch-and-cut • valid inequalities • asymmetric • multidepot • vehicle routing • upper-bound procedure

1. Introduction

Many practically inspired routing, scheduling, and network design problems have an underlying structure corresponding to a directed graph with asymmetric costs and multiple depots. For instance, routing in inner cities is clearly asymmetric because of one-way streets and congestion, and it is often organized from various city hubs. Other examples include applications in home and healthcare (Cappanera and Scutellà 2014), crew and vehicle scheduling problems (Fischetti et al. 2001, Paraskevopoulos et al. 2017), rural postman problems (Fernández et al. 2017), and skill-based routing problems (Cappanera et al. 2013).

For symmetric routing problems, both with and without multiple depots, a thorough understanding of the combinatorial structure has enabled the development of effective exact-solution approaches (e.g., Letchford and Salazar-González 2006, Baldacci and

Mingozi 2009). In addition, it is recognized that understanding the combinatorial structure of the asymmetric counterpart of the well-known traveling-salesman problem (TSP) substantially enhances the computational efficiency of its solution approaches (Balas and Fischetti 2007, Fischetti et al. 2007). Motivated by these structural results and the characteristics of current emerging routing applications, we study stylized (i.e., without additional problem-specific constraints) asymmetric routing problems with more than a single vehicle and more than a single depot.

Stylized vehicle routing problems with *asymmetric costs* and *multiple depots* are defined on a directed graph $G = (V', A)$, where the node set V' is partitioned into a depot set $D = \{1, \dots, r\}$ and a customer set $V = \{r+1, \dots, r+n\}$. The arc set is defined as $A = \{(i, j) \mid i \neq j \in V \vee i \in D, j \in V \vee i \in V, j \in D\}$. An *asymmetric cost* $c_{ij} \geq 0$ is incurred when traveling along arc $(i, j) \in A$,

that is, c_{ij} need not equal c_{ji} . Each available vehicle makes a single tour, starting and ending at the same depot. The objective is to find a set of cost-minimizing tours that fulfill all customer demands. Specifically, we focus on the following four problem variants.

i. The asymmetric multidepot TSP (A-MDTSP): A single vehicle with unlimited capacity is available at each depot.

ii. The asymmetric multidepot multiple TSP (A-MDmTSP): There are m vehicles available at each depot. Vehicle capacity is unlimited, but the number of customers per tour is bounded by $[l^{\min}, l^{\max}]$.

iii. The asymmetric multidepot capacitated vehicle routing problem (A-MDCVRP): There are m vehicles available at each depot. Each vehicle has a limited capacity Q , and each customer has a demand $q_i > 0$ ($i \in V$).

iv. The asymmetric capacitated location routing problem (A-CLRP): This problem extends the A-MDCVRP by incorporating depot location decisions, that is, using a depot comes at a fixed cost $\tilde{c} \geq 0$.

We present a branch-and-cut framework that can address each of the four problem variants above. The design of our framework is generic in the sense that—with some problem-specific, nonstructural adaptations—it may be applied to other routing problems with multiple depots and asymmetric cost structures. We focus on the above four problem variants because they cover a wide range of classic routing characteristics. The uncapacitated A-MDTSP is our starting point because it follows most naturally from the extant literature of the asymmetric TSP (ATSP). The A-MDmTSP then arises by introducing multiple vehicles per depot and bounds on the number of customers in a tour. To show the general applicability of our framework, we introduce vehicle capacities in the A-MDCVRP and depot opening costs in the A-CLRP. To indicate that results apply generically to routing problems with multiple depots and asymmetric costs, we use the descriptor “asymmetric multidepot vehicle routing problem” (A-MDVRP). Some results are specific to one of the four problem variants, in which case this is indicated explicitly. Numerical experiments are limited to the four problem variants.

The first contribution of this paper is a series of valid inequalities (VIs), with accompanying separation algorithms, tailored to asymmetric cost structures and multiple depots. The new VIs originate from the well-known D_k and closed alternating trail (CAT) inequalities for the ATSP (Fischetti 1991, Fischetti and Toth 1997, Balas and Fischetti 2007, Fischetti et al. 2007) and make explicit use of the directedness of the underlying graph. This results in strengthened VIs, whereas applying exact methods for symmetric routing problems on undirected graphs to the A-MDVRP leaves this unexploited. We propose three classes of so-called

D_k inequalities that (i) eliminate subtours, (ii) impose bounds on the tour size, and (iii) eliminate paths that begin and arrive at different depots, that is, the proposed D_k inequalities are not only valid but also define the integer polytope of the A-MDVRP. Furthermore, CAT inequalities are adapted to a multidepot setting. To the best of our knowledge, there has been no study on the integer polytope of *asymmetric* routing problems since a series of works on the ATSP and its variations (Balas 1989, Fischetti 1991, Balas and Fischetti 1993, Fischetti et al. 1994, Fischetti and Toth 1997, Ascheuer et al. 2001, Mak and Ernst 2007). One exception is the literature on arc-routing problems in which every arc must be traversed instead of every node visited (e.g., Corberán et al. 2008, Bosco et al. 2013).

Our second contribution is the development of a generic branch-and-cut algorithm for the A-MDVRP that integrates the results from capacitated vehicle routing problems (Lysgaard et al. 2004; Letchford and Salazar-González 2006, 2015), location routing problems (Laporte et al. 1988, Belenguer et al. 2011), and symmetric multidepot TSPs (Benavent and Martínez 2013). Along with the newly proposed VIs, our branch-and-cut algorithm includes the following adapted asymmetric VIs: *subtour elimination constraints*, *classical path elimination constraints*, *CAT inequalities*, D_k^+ and D_k^- *constraints*, *strengthened comb inequalities*, *T- and T1-comb inequalities*, *framed capacity inequalities*, and *homogenous and large multistar inequalities*.

Third, complementary to the branch-and-cut algorithm, we use a compact formulation with so-called neighborhood-arc constraints (i.e., considering a subset of promising arcs only) to provide upper bounds in a computationally simple yet effective manner. This upper-bound procedure improves the overall efficiency of the branch-and-cut algorithm considerably. We refer to the joint use of the branch-and-cut algorithm and the upper-bound procedure as the *branch-and-cut framework*.

Fourth, we show the effectiveness of the branch-and-cut framework on the four problem variants as detailed above. For each problem variant, we provide insights into the effectiveness of the VIs. Furthermore, to obtain a general understanding of how asymmetry in the cost structure relates to computational difficulty, we perform extensive experiments on three classes of asymmetric cost structures. These classes have arc costs ranging from completely random ($c_{ij} \sim U(1; 1,000)$) to Euclidean with a small noise. The instances used are publicly available and can serve as benchmark instances for future research. Our implementation of the branch-and-cut framework used while writing this paper, the instances, and supporting documentation are available in the online supplement. Moreover, to foster research on asymmetric vehicle routing problems, we provide public access to our latest C++ implementation of the branch-and-cut framework.¹

We show that the VIs help to reduce root-node optimality gaps by up to 67.2% on our benchmark instances. Regarding the A-MDTSP, we can solve all instances of Bektaş et al. (2017) to optimality, of which two were not solved before, and we find an improved solution for one of the instances. This performance is driven by an average root-node reduction of 86.5% compared with Bektaş et al. (2017). Furthermore, our branch-and-cut framework can solve to optimality our newly developed benchmark instances for the A-MDTSP containing up to 400 customers and 50 depots, which is significantly larger than the 300-customer and 60-depot instances of Bektaş et al. (2017).

Regarding the other three problem variants, no benchmark instances exist for stylized routing problems with asymmetric costs on directed graphs (i.e., not without additional problem-specific constraints). Therefore, we tested our branch-and-cut framework on newly proposed benchmark instances, and it appears to be effective for all remaining problem variants. For example, A-MDmTSP instances with up to 100 customers and 20 depots can be solved to optimality. This size is comparable to state-of-the-art methods for the single-depot symmetric multiple TSP, which has a significantly smaller solution space.

Furthermore, we compare the branch-and-cut framework to a basic branch-and-cut algorithm, where the latter is our implementation of what can be considered state of the art. The branch-and-cut framework that exploits our new VIs shows considerable computational improvements over the basic algorithm. Specifically, the branch-and-cut framework solves considerably more instances to optimality than the basic algorithm, is on average faster for the instances that are solved to optimality by both algorithms, has smaller optimality gaps for instances that cannot be solved to optimality, and provides higher lower bounds for the instances for which no upper bound is found.

Finally, we briefly discuss the literature related to the capacitated variants that we consider herein and, in particular, to the A-CLRP. The location-routing-problem stream is vast, as one can see from numerous survey papers (Min et al. 1998, Nagy and Salhi 2007, Prodhon and Prins 2014, Drexler and Schneider 2015, Schneider and Drexler 2017). The number of exact-solution approaches tailored to solving capacitated location routing problems is limited. Furthermore, all the established benchmark instances have Euclidean distances. Although state-of-the-art approaches such as the one by Contardo et al. (2014) are typically of a branch-cut-and-price type, Contardo et al. (2014) also refine many VIs (for the symmetric case) to enhance the computational efficiency of their solution method. Such approaches are leading for multidepot capacitated routing problems as well, because these are typically studied in conjunction with capacitated location

routing problems. This indicates that fundamental knowledge about VIs in a branch-and-cut scheme could help develop decomposition-based methods.

The remainder of this paper is organized as follows. In Section 2, we provide a compact problem formulation that we use in our upper-bound procedure, and we present the basic formulation that is used in the branch-and-cut algorithm. In Section 3, we provide our series of VIs that are included in our branch-and-cut algorithm, and we discuss their corresponding separation algorithms in Section 4. In Section 5, we outline the branch-and-cut framework, consisting of the branch-and-cut algorithm and the upper-bound procedure. In Section 6, we present the computational experiments involving the branch-and-cut framework. We conclude and provide directions for further research in Section 7.

2. Problem Formulation

We present two mixed-integer programming (MIP) formulations for the A-MDVRP. The first formulation is a compact formulation used in our upper-bound procedure and consists of a *polynomial* number of variables and constraints. The second formulation, which we call the basic formulation, consists of exponentially many constraints. The latter formulation serves as the starting point for our branch-and-cut algorithm and the derivation of our new VIs.

2.1. Compact Formulation

Let $x_{ij} \in \{0, 1\}$ be the binary variable indicating whether arc $(i, j) \in A$ is traversed, and let $z_d \in \{0, 1\}$ be the binary variable denoting whether depot $d \in D$ is opened. In addition, continuous variables y_{ij} that model the vehicle load along arc $(i, j) \in A$ are used to ensure that the capacity of a vehicle is not exceeded and to eliminate subtours. Continuous variables u_i , equaling the originating depot of the vehicle that visits node $i \in V$, are used to ensure that tours start and end at the same depot.

We present the following compact MIP formulation P_c for the A-MDVRP:

$$\min \sum_{(i,j) \in A} c_{ij}x_{ij} + \tilde{c} \sum_{d \in D} z_d \quad (1)$$

$$\text{s.t.} \quad \sum_{j \in V'} x_{ij} = \sum_{j \in V'} x_{ji} = 1 \quad \forall i \in V, \quad (2)$$

$$\sum_{j \in V} x_{dj} = \sum_{j \in V} x_{jd} \leq m \quad \forall d \in D, \quad (3)$$

$$x_{di} \leq z_d \quad \forall i \in V, d \in D, \quad (4)$$

$$\sum_{i \in V'} (y_{ij} - y_{ji}) = q_j \quad \forall j \in V, \quad (5)$$

$$\sum_{d \in D} \sum_{j \in V} y_{dj} - \sum_{j \in V} q_j = 0, \quad (6)$$

$$y_{ij} - (Q - q_i)x_{ij} \leq 0 \quad \forall i \in V', j \in V, \quad (7)$$

$$\sum_{d \in D} dx_{di} \leq u_i \quad \forall i \in V, \quad (8)$$

$$\sum_{d \in D} dx_{id} \leq u_i \quad \forall i \in V, \quad (9)$$

$$|D| - \sum_{d \in D} (|D| - d)x_{di} \geq u_i \quad \forall i \in V, \quad (10)$$

$$|D| - \sum_{d \in D} (|D| - d)x_{id} \geq u_i \quad \forall i \in V, \quad (11)$$

$$u_i - u_j - M(1 - x_{ij} - x_{ji}) \leq 0 \quad \forall i, j \in V, i \neq j, \quad (12)$$

$$y_{dj} + M(1 - x_{dj}) \geq l^{\min} \quad \forall d \in D, j \in V, \quad (13)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A, \quad (14)$$

$$z_d \in \{0, 1\} \quad \forall d \in D, \quad (15)$$

$$u_i \geq 0 \quad \forall i \in V, \quad (16)$$

$$0 \leq y_{ij} \leq Q \quad \forall i, j \in V. \quad (17)$$

Objective (1) minimizes the sum of travel costs and depot opening costs. Constraints (2) ensure that every customer is visited exactly once. Constraints (3) limit the number of vehicles (or tours) at the depot, if necessary. Constraints (4) link the z_d and x_{ij} variables. Constraints (5)–(7) ensure that loads do not exceed the vehicle capacity and exclude subtours (Lahyani et al. 2018). Constraints (8)–(12) ensure that paths cannot start and end at different depots; to be precise, these constraints ensure that (i) the variables u_i take the same value for consecutive customers, and (ii) for customers connected from or to a depot, the value of u_i equals the depot index to which the customer is connected. Constraints (13) ensure that the lower bound on the tour size is respected. Constraints (14)–(17) define the domains of the variables.

The above compact MIP formulation P_c covers all four problem variants introduced in Section 1. For each of the four problem variants, Table 1 provides the appropriate parameter values to be used in this MIP formulation; an asterisk indicates that the particular parameter has no model-specific restrictions.

The compact MIP formulation could be used to solve any A-MDVRP with off-the-shelf MIP solvers such as Gurobi and CPLEX. However, we use it for our upper-bound procedure, for which purpose we consider a restricted version, using a subset of the arcs. The subset consists of (i) all arcs between depot nodes and customer nodes, and (ii) for every customer node i , the δ cheapest outgoing arcs connected

to other customer nodes. The resulting formulation is called the δ -compact formulation, which we refer to as P_δ . The upper-bound procedure is based predominantly on solving δ -compact formulations iteratively, and it runs parallel to the branch-and-cut algorithm to provide upper bounds quickly. The details of the upper-bound procedure are provided in Section 5.1.

2.2. Basic Formulation

In the following, we present a basic formulation that we use as the starting point of our branch-and-cut algorithm. Consider the following formulation P :

$$\min \sum_{(i,j) \in A} c_{ij}x_{ij} + \tilde{c} \sum_{d \in D} z_d \quad (18)$$

$$\text{s.t.} \quad \sum_{j \in V'} x_{ij} = \sum_{j \in V'} x_{ji} = 1 \quad \forall i \in V, \quad (19)$$

$$\sum_{j \in V} x_{dj} = \sum_{j \in V} x_{jd} \leq m \quad \forall d \in D, \quad (20)$$

$$x_{di} \leq z_d \quad \forall i \in V, d \in D, \quad (21)$$

$$\text{Depot fixing constraints,} \quad (22)$$

$$\text{Capacity/tour size constraints,} \quad (23)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in V, \quad (24)$$

$$z_d \in \{0, 1\} \quad \forall d \in D. \quad (25)$$

Except for (22) and (23), the objective and the constraints are equal to the compact formulation P_c . Note that the above formulation consists of x_{ij} and z_d variables only, whereas the compact formulation also uses the continuous variables y_{ij} and u_i .

In the remainder of this study, we use the following shorthand notation. We define $x(T) := \sum_{(i,j) \in T} x_{ij}$ for $T \subseteq A$, and $x(S_1, S_2) := \sum_{i \in S_1, j \in S_2, (i,j) \in A} x_{ij}$ for $S_1, S_2 \subseteq V'$. In addition, we denote $x(\gamma^+(S)) := x(S, S)$, $x(\delta^+(S)) := x(S, V' \setminus S)$, and $x(\delta^-(S)) := x(V' \setminus S, S)$ for $S \subset V'$. Thus, $\delta^+(S)$ and $\delta^-(S)$ denote all outgoing and incoming arcs of the node set S , respectively, and $x(\delta^+(S))$ and $x(\delta^-(S))$ denote their corresponding x_{ij} variables. For brevity, we write $\delta^+(i) := \delta^+(\{i\})$ and $\delta^-(i) := \delta^-(\{i\})$ for $i \in V'$.

Depot fixing constraints (22), which ensure that each tour starts and ends at the same depot, are present in all problem variants. The following depot fixing constraints are commonly used (e.g., Laporte et al. 1988, Belenguer et al. 2011) and are referred to as path elimination constraints:

$$\sum_{o \in I} x_{oi} + \sum_{o \in D \setminus I} x_{jo} + x(\gamma^+(S \cup \{i, j\})) \leq |S| + 2 \quad (26)$$

for all $S \subset V, I \subset D, i, j \in V \setminus S$.

However, these constraints do not exclude single customer tours starting and ending at different depots. In Section 3, we present our new class of depot

Table 1 Parameter Restrictions of the Various Problem Variants

Problem variant	Q	q_i	$l^{\min}$	$\tilde{c}$	m
A-MDTSP	n	1	1	0	1
A-MDmTSP	$l^{\max}$	1	*	0	*
A-MDCVRP	*	*	1	0	*
A-CLRP	*	*	1	*	*

fixing constraints based on D_k inequalities that can replace the above set of constraints (26). As will be explained later, our proposed constraints do exclude single customer tours starting and ending at different depots.

Capacity constraints (23) are present in the A-MDVRP variants to eliminate subtours and are particularly useful when vehicle capacity restrictions are considered (i.e., for the A-MDCVRP and the A-CLRP). In general, they have the following form:

$$x(\gamma(S)) \leq |S| - k(S) \quad \text{for all } S \subset V, \quad (27)$$

where $k(S)$ is the number of vehicles required to serve all customers in S . The number of vehicles $k(S)$ is the solution of a bin-packing problem and is commonly replaced by $\lceil \sum_{i \in S} q_i / Q \rceil$ (Lygaard et al. 2004). Letchford and Salazar-González (2006, 2015) proposed refinements to these capacity constraints. Note that the capacity constraints from the literature are evaluated mostly in the light of symmetric single-depot vehicle-routing problem instances, whereas we adapt these capacity constraints to asymmetric multidepot settings in Section 3.2.3.

Finally, tour-size constraints are required only for the A-MDmTSP variant, where they ensure that each tour includes at least $l^{\min}$ and at most $l^{\max}$ customers. In general, the upper-bound constraints can be imposed by using the above capacity constraints by setting $Q = l^{\max}$ and $q_i = 1$ for all $i \in V$. Tour-size constraints have been studied for single-depot symmetric vehicle routing problems (e.g., Bektaş et al. 2019, Gouveia et al. 2013), but not yet for the asymmetric multidepot settings such as ours. In Section 3.3.1, we present a class of D_k inequalities that can be used to enforce tour-size constraints in the asymmetric multidepot setting.

3. Model Constraints and Valid Inequalities

This section introduces the model constraints (Section 3.1) and the VIs (Section 3.2) that are used in the branch-and-cut algorithm. Further refinements for the A-MDmTSP are discussed in Section 3.3. The corresponding separation algorithms are discussed in Section 4.

In the remainder of this paper, we refer to inequalities that exhibit model defining characteristics as *constraints* and to inequalities that strengthen the linear hull but do not exhibit problem defining characteristics as *VIs*. For readability purposes, we specify actual problem variants only if the results presented below are not generally applicable to all variants, whereas we write “A-MDVRPs” if the results hold generally.

3.1. Model Constraints

We continue by describing two families of constraints that are valid for A-MDVRPs. First, we show that the D_k^+ and D_k^- constraints are also valid for the multidepot case and that those constraints can be used to eliminate subtours. Next, we extend those constraints to eliminate paths starting and ending at different depots, which we refer to as the D_k^+ and D_k^- depot fixing constraints.

3.1.1. D_k^+ and D_k^- Constraints. The extant literature presents D_k inequalities for the ATSP, that is, for single-tour single-depot routing problems (see Grötschel and Padberg 1985). To provide a complete overview of the inequalities that are valid for the A-MDVRP, we restate the D_k inequalities and give an independent proof showing that these are also valid in our setting.

In Figure 1, we show an example of a D_k^+ inequality subject to a customer sequence $\{i_1, i_2, i_3, i_4, i_k\} \subset V$. The figure depicts all arcs that are present in the left-hand side of (28), where the dashed arcs correspond to arcs that have coefficient 2.

Theorem 1. Let $\mathcal{J} = \{i_1, i_2, \dots, i_k\} \subseteq V$ be a sequence of distinct customer nodes. Then the inequalities

$$D_k^+ : x_{i_1 i_k} + \sum_{h=2}^k x_{i_h i_{h-1}} + 2 \sum_{h=2}^{k-1} x_{i_h i_h} + \sum_{h=3}^{k-1} x(\{i_2, \dots, i_{h-1}\}, i_h) \leq k-1, \quad (28)$$

$$D_k^- : x_{i_k i_1} + \sum_{h=2}^k x_{i_{h-1} i_h} + 2 \sum_{h=2}^{k-1} x_{i_h i_{h-1}} + \sum_{h=3}^{k-1} x(i_h, \{i_2, \dots, i_{h-1}\}) \leq k-1 \quad (29)$$

are valid cuts for the integer polytope of A-MDVRPs. Moreover, subtour elimination constraints are formulated correctly by (28) and/or (29).

Proof. We consider only the D_k^+ inequality because the proof for the D_k^- inequality follows the same line of reasoning. Notice that (28) reduces to cycle elimination

Figure 1. (Color online) Illustration of All Arcs Present in D_k^+ Inequality with Respect to Customer Sequence $\{i_1, i_2, i_3, i_4, i_k\} \subset V$

Note. The solid arcs are included once and the dashed ones twice.

constraints (i.e., $x_{i_1 i_k} + \sum_{h=2}^k x_{i_1 i_{h-1}} \leq k - 1$) if $2 \sum_{h=2}^{k-1} x_{i_1 i_h} + \sum_{h=3}^{k-1} x(\{i_2, \dots, i_{h-1}\}, i_h) = 0$. It follows immediately that the D_k^+ inequalities can be interpreted as lifted variants of such cycle elimination constraints; thus, (28) eliminates all subtours.

It is left to show that (28) is valid for the A-MDVRP. All arcs present in (28) with coefficient 2 originate from node i_1 , and thus, any feasible solution uses at most one arc with coefficient 2. In the remainder of this proof, we consider two cases: in the first case, no arcs with coefficient 2 are used; in the second case, exactly one of those arcs is used.

Case 1: Suppose $x_{i_1 i_h} = 0$ for all $h \in \{2, \dots, (k-1)\}$, that is, no arcs with coefficient 2 are used. Then (28) reduces to a summation of arcs with coefficient 1 between k customer nodes, which can be at most $k - 1$ because otherwise there would be subtours.

Case 2: Suppose $x_{i_1 i_h} = 1$ for some $h \in \{2, \dots, (k-1)\}$. Let us partition the customer node sequence $\mathcal{J}$ into $\mathcal{J}_1 = \{i_1, \dots, i_h\}$ and $\mathcal{J}_2 = \{i_{h+1}, \dots, i_k\}$. Let $\mathcal{A}_\ell$ (with $\ell = 1, 2$) be the complete arc set between nodes in $\mathcal{J}_\ell$, and let $\mathcal{A}_\ell$ be the set of all arcs (i_i, i_j) that are present in (28) with $i_j \in \mathcal{J}_\ell$. Note that $\mathcal{A}_1 \cup \mathcal{A}_2$ contains all arcs that are present in (28) and that any feasible solution uses at most $|\mathcal{J}_\ell| - 1$ arcs from $\mathcal{A}_\ell$ because otherwise there would be subtours.

Note that $x_{i_1 i_h} = 1$ implies $x_{i_{h+1} i_h} = 0$ and $x_{i_1 i_k} = 0$. Then, by the definition of $\mathcal{A}_1$, it holds that for each arc $(i_i, i_j) \in \mathcal{A}_1$ we have $i_i, i_j \in \mathcal{J}_1$, and thus, $\mathcal{A}_j \subset \mathcal{A}_1$. It follows that a feasible solution can traverse at most $|\mathcal{J}_1| - 1$ arcs from $\mathcal{A}_1$. Next, because $x_{i_1 i_k} = 0$, there are no arcs in $\mathcal{A}_2$ that point toward node i_k , and thus, a feasible solution traverses at most $|\mathcal{J}_2| - 1$ arcs from $\mathcal{A}_2$.

It follows that a feasible solution traverses at most $|\mathcal{J}_1| - 1 + |\mathcal{J}_2| - 1 = k - 2$ arcs from $\mathcal{A}_1 \cup \mathcal{A}_2$. All such arcs appear in (28) with weight 1, except for arc (i_1, i_h) , which appears with weight 2. Thus, the left-hand side of (28) is at most $k - 1$. We conclude that (28) eliminates all subtours and that it is valid for A-MDVRPs. $\square$

3.1.2. D_k^+ and D_k^- Depot Fixing Constraints. We further exploit the structure of (28) and (29) to derive the D_k^+ and D_k^- depot fixing constraints that eliminate paths starting and ending at different depots, and thus, we can model the depot fixing constraints (22) instead of (26).

An example of a D_k^+ depot fixing constraint (30) is given in Figure 2. In this figure, we depict all arcs that arise in the left-hand side of (30), where red dashed arcs correspond to arcs with coefficient 2, and blue dotted arcs represent the arcs from i_1 to the depot set $O \subset D$ and the arcs from the complement depot set $D \setminus O$ to i_k .

Theorem 2. Let $\mathcal{J} = \{i_1, i_2, \dots, i_k\} \subseteq V$ be a sequence of distinct customer nodes, and let $O \subset D$ be a subset of depots. The inequalities

Figure 2. (Color online) Illustration of All Arcs Present in D_k^+ Depot Constraint with Respect to Customer Sequence $\{i_1, i_2, i_3, i_4, i_k\}$ and Depot Subsets $O \subset D$ and $D \setminus O$

Notes. The solid arcs are included once and the dashed ones twice. The dotted arcs represent all arcs from $D \setminus O$ to i_k and all arcs from i_1 to O .

$$D_k^+ \text{ depot} : \sum_{s \in O} x_{i_1 s} + \sum_{s \in D \setminus O} x_{s i_k} + \sum_{h=2}^k x_{i_h i_{h-1}} + 2 \sum_{h=2}^k x_{i_1 i_h} + \sum_{h=3}^k x(\{i_2, \dots, i_{h-1}\}, i_h) \leq k, \quad (30)$$

$$D_k^- \text{ depot} : \sum_{s \in O} x_{s i_1} + \sum_{s \in D \setminus O} x_{i_k s} + \sum_{h=2}^k x_{i_{h-1} i_h} + 2 \sum_{h=2}^k x_{i_h i_1} + \sum_{h=3}^k x(i_h, \{i_2, \dots, i_{h-1}\}) \leq k \quad (31)$$

are valid cuts for the integer polytope of A-MDVRPs. Moreover, depot fixing constraints of A-MDVRPs are formulated correctly by (30) and/or (31).

Proof. We consider only the D_k^+ depot fixing constraint because the proof for the D_k^- one follows the same line of reasoning. First, let us show that path elimination constraints are formulated correctly by (30). By contradiction, assume that there is a tour $\{s_1, i_1, \dots, i_k, s_2\}$ with $s_1, s_2 \in D$, $s_1 \neq s_2$, and $i_1, \dots, i_k \in V$. Consider the sequence $\{i_k, \dots, i_1\}$ and $O := s_2$ in (30), and we obtain a contradiction. Thus, path elimination constraints are formulated correctly by (30).

In the remainder of this proof, we consider two cases to show that (30) is valid for the A-MDVRP: in the first case, no arcs with coefficient 2 are used; in the second case, exactly one of these arcs is used.

Case 1: Suppose $x_{i_1 i_h} = 0$ for all $h \in \{2, \dots, k\}$, that is, no arcs with coefficient 2 are used. Customer nodes can be connected to or from at most one depot, and it follows trivially that $\sum_{s \in O} x_{i_1 s} + \sum_{s \in D \setminus O} x_{s i_k} \leq 2$. Furthermore, a solution traverses at most $k - 1$ arcs between the k nodes in $\mathcal{J}$ because otherwise there would be subtours between these customer nodes; thus, $\sum_{h=2}^k x_{i_h i_{h-1}} + \sum_{h=3}^k x(\{i_2, \dots, i_{h-1}\}, i_h) \leq k - 1$. A violation for (30) occurs only if the previous two inequalities hold with equality. However, the only way that the last inequality holds with equality is by setting $x_{i_h i_{h-1}} = 1$ for $h \in \{2, \dots, k\}$, which implies that we must have $\sum_{s \in O} x_{i_1 s} + \sum_{s \in D \setminus O} x_{s i_k} \leq 1$ because otherwise there would be a path between two depots. It follows that, under this case, the left-hand side of (30) is at most k .

Case 2: Suppose $x_{i_1 i_h} = 1$ for some $h \in \{2, \dots, k\}$, that is, one of the arcs with coefficient 2 is used. Consider the sequence $\{i_1, i_2, \dots, i_k, i_{k+1}\}$, where we define $x_{i_1 i_{k+1}} := \sum_{s \in O} x_{i_1 s}$ and $x_{i_{k+1} i_k} := \sum_{s \in D \setminus O} x_{s i_k}$ to arrive at an inequality as denoted in Theorem 1. By following the same argument as that in the proof of Theorem 1, for the case $x_{i_1 i_h} = 1$ we obtain that (30) is a VI for the integer polytope of A-MDVRPs. $\square$

Note that, when $k = 1$ in Theorem 2, that is, $\mathcal{J} = \{i_1\}$, we have

$$\sum_{s \in O} x_{s i_1} + \sum_{s \in D \setminus O} x_{i_1 s} \leq 1,$$

which is a depot fixing constraint involving a single node $i_1 \in V$. Note that this differs from the classical path-eliminating constraints (26) because these are used typically when tours of a single size are forbidden. As Theorems 1 and 2 show, the modeling capabilities of those D_k -type constraints are rather extensive. As a direct consequence of the above theorems, we obtain the following corollary.

Corollary 1. *A-MDTSPs are formulated correctly as minimize (18) subject to (19)–(21), and completed with*

- i. *at least one of (28) and (29) to eliminate subtours, and*
- ii. *at least one of (30) and (31) for depot fixing constraints.*

3.2. Valid Inequalities

In the following, we present several strengthening VIs. First, we discuss CAT-type inequalities, introduced by Balas (1989) for the ATSP, and generalize these to the multidepot case. We then provide several comb-type inequalities and capacity inequalities, both generalized and adapted from the capacitated vehicle routing problem case.

3.2.1. CAT-Type Inequalities. We modify the CAT inequalities derived by Balas (1989) so that they are valid for the A-MDVRP. Before discussing the CAT inequalities, we must introduce the notion of incompatible arcs. Notice that this is different from Balas (1989). An arc $(i, j) \in A$ is incompatible in the following circumstances.

1. If $i, j \in V$, then (i, j) is incompatible with (i, u) , (u, j) , and (j, i) for any $u \in V'$.
2. If $i \in D$ and $j \in V$, then (i, j) is incompatible with (i) (u, j) for any $u \in V'$ and (ii) (j, o) for any $o \in D \setminus \{i\}$.
3. If $i \in V$ and $j \in D$, then (i, j) is incompatible with (i) (i, u) for any $u \in V'$ and (ii) (o, i) for any $o \in D \setminus \{j\}$.

A CAT inequality is described by a sequence of arcs $T = \{a_1, \dots, a_t\}$ such that $a_1, \dots, a_t \in A$, $t \geq 5$ and odd, and any arc in T is incompatible with its neighbor arcs in T and compatible with all other arcs in T . The neighbor arcs of a_1 are considered to be a_2 and a_t .

In other words, we consider that arcs $\{a_1, \dots, a_t\}$ are placed on a circle maintaining their order.

Furthermore, node i is called a source if $\delta^+(i) \cap T = 2$ or a sink if $\delta^-(i) \cap T = 2$. In addition, an arc $(i, j) \in A \setminus T$ is called a chord of type 1 if $i \in V$ is a source and $j \in V$ is a sink. We let R be the collection of all chords of type 1 as induced by the set T and the graph G .

Theorem 3. *Let $T = \{a_1, \dots, a_t\} \subset A$ with an odd $t \geq 5$ such that any arc in T is incompatible with its neighbor arcs and is compatible with the other arcs in T . Let R be the set of chords of type 1. Then, the CAT inequality*

$$x(T \cup R) \leq \frac{t-1}{2} \quad (32)$$

is valid for the integer polytope of A-MDVRPs.

Proof. Because $T \cap R = \emptyset$, we have $x(T \cup R) = x(T) + x(R)$. By the definition of “incompatibility,” any two neighboring arcs are mutually exclusive. In other words, the corresponding values of x for any two consecutive arcs cannot be 1 at the same time. Because t is odd, we obtain

$$x(T) \leq \frac{t-1}{2}. \quad (33)$$

Next, let us examine set R . Considering $(i, j) \in R$, we have $\delta^+(i) \cap T = 2$ and $\delta^-(j) \cap T = 2$ by definition. Thus, there are incompatible neighbor arcs (i, u) and (i, v) in T for some $u, v \in V'$. Arcs (i, u) and (i, v) have to be neighbors because all nonneighbor arcs are compatible by the construction of CAT. The same argument holds for arcs $(u', j) \in T$ and $(v', j) \in T$ for some $u', v' \in V'$.

By contradiction, suppose that $x_{ij} = 1$ and the equality holds for (33). If $x_{ij} = 1$, then we must have $x_{iu} = x_{iv} = x_{u'j} = x_{v'j} = 0$. Because $(i, j) \in R \subset A \setminus T$, these two neighbors do not overlap. So the corresponding x value of set T must be of the form $\{0, 0, \dots, 0, 0, \dots\}$. On the other hand, by using $x(T) = (t-1)/2$, there cannot be three consecutive zeros. Thus, the arcs in T must have the form $\{0, 0, 1, 0, 1, 0, 1, \dots, 0, 1\}$, which contradicts the pattern $\{0, 0, \dots, 0, 0, \dots\}$. Therefore, when $x(T) = (t-1)/2$, we have $x(R) = 0$, and thus, (32) holds.

Suppose now $x(R) = q$ with $q > 0$. As before, whenever $x_{ij} = 1$ for $(i, j) \in R$, we have $x_{iu} = x_{iv} = x_{u'j} = x_{v'j} = 0$. Excluding all these zero-valued arcs, there are $t - 4q$ remaining arcs of T that are separated into at most $2q$ parts. Because $t - 4q$ is odd, there are at most $(2q - 1)$ parts with an odd number of arcs and at least one part with an even number of arcs. The value of a part with odd $\tilde{t}$ arcs is at most $(\tilde{t} + 1)/2$, and the value of a part with even $\hat{t}$ arcs is at most $(\hat{t}/2)$. By summing all these separated parts, we obtain

$$x(T) + x(R) \leq \frac{(t - 4q) + (2q - 1)}{2} + q = \frac{t - 1}{2},$$

and we have shown that (32) holds for A-MDVRPs. $\square$

3.2.2. Comb-Type Inequalities. Fischetti (1991) defined asymmetric comb inequalities for the ATSP on a complete digraph (V, A) , where V is a node set and A is the arc set with $n(n-1)$ arcs. Define a set handle $H \subset V$ and an odd number of subsets called teeth $T_i \subset V$ with $i = 1, \dots, t$ and $t \geq 3$ such that

$$H \cap T_i \neq \emptyset, \quad T_i \setminus H \neq \emptyset, \quad \text{and} \quad T_i \cap T_j = \emptyset \quad (34)$$

for all $i, j = 1, \dots, t$.

Fischetti (1991) showed that the comb inequality

$$x(\delta^+(H)) + \sum_{j=1}^t x(\delta^+(T_j)) \geq \frac{3t+1}{2} \quad (35)$$

describes a facet of the ATSP polytope on a directed graph for $t \geq 7$.

The comb inequalities have been generalized to the multidepot multiple TSP with symmetric costs in Benavent and Martínez (2013) and are shown to be facets for $t \geq 3$. Similarly, we can reformulate the ATSP combs (35) such that they are valid for the A-MDVRP by imposing the additional constraint that all the depot nodes should be in the same element of the comb.

Proposition 1. *Let $G_d \subseteq G$ be a graph with only a single depot node $d \in D$. In graph G_d , we can derive the comb inequalities (35). Consider such an inequality and add all remaining depot nodes $d' \in D \setminus \{d\}$ to all the sets (i.e., handle and teeth) that contain d . This modified inequality (35) with $t \geq 3$ is valid for the A-MDVRP and is referred to as an ATSP comb.*

Proof. Every edge in the formulation of Benavent and Martínez (2013) is defined by x_{ij} and x_{ji} with $x_{ij} = x_{ji}$. In other words, both incoming and outgoing arcs are included in their formulation if the corresponding asymmetric case is considered. Reformulating the H -comb inequality of Benavent and Martínez (2013) for the asymmetric case, we derive

$$x(\delta^+(H)) + x(\delta^-(H)) + \sum_{j=1}^t x(\delta^+(T_j)) + \sum_{j=1}^t x(\delta^-(T_j)) \geq 3t + 1.$$

Because the numbers of incoming and outgoing arcs are the same, we obtain (35) directly. $\square$

In general, the symmetric case is translated into the asymmetric case simply by replacing every edge (symmetric case) by its two corresponding arcs

(asymmetric case). However, replacing every edge by the two corresponding arcs does not work for the generalized comb inequalities given by Benavent and Martínez (2013) because in their inequality edges are defined by both variables x_{ij} and x_{ji} with $x_{ij} = x_{ji}$ for any $i, j \in V$. For completeness, we reformulate the T - and H -comb inequalities proposed in Benavent and Martínez (2013) so that they are valid for the A-MDVRP.

Proposition 2. *Let the handle $H \subset V \cup D$ and an odd number of teeth $T_i \subset V$ with $i = 1, \dots, t$ and $t \geq 3$ be defined such that (34) is satisfied and such that $H \cap D \neq \emptyset$ and $D \setminus H \neq \emptyset$. The following H -comb inequality is valid for the A-MDVRP:*

$$x(\delta^+(H)) + \sum_{j=1}^t x(\delta^+(T_j)) \geq \frac{3t+1}{2}. \quad (36)$$

Proposition 3. *Let the handle $H \subset V \cup D$ and the teeth $T_i \subset V \cup I$ with $i = 1, \dots, t$ and $t \geq 1$ be defined such that (34) is satisfied. In addition, assume that (i) $T_i \cap I \neq \emptyset$ for $i = 1, \dots, t$, (ii) $H \setminus \bigcup_{i=1}^t T_i \neq \emptyset$, and (iii) $I \setminus \bigcup_{i=1}^t T_i \neq \emptyset$. Then the following T -comb inequality is valid for the A-MDVRP:*

$$x(\delta^+(H)) + \sum_{j=1}^t x(\delta^+(T_j)) \geq 2t + 2. \quad (37)$$

3.2.3. Capacity Inequalities. Capacity inequalities such as the generalized large multistar inequalities and the knapsack large multistar inequalities are defined for the symmetric single-depot capacitated vehicle routing problem. However, the inequalities involve only customer nodes, thereby allowing us to apply the transformation from edge variables (for the symmetric case) to arc variables (for the asymmetric case) in the following way: we substitute every edge variable x_e with the arc variables $x_{ij} + x_{ji}$, where i and j are the endpoints of e . Thus, the VIs for single-depot capacitated vehicle routing problems from Letchford and Salazar-González (2015) are directly applicable for multidepot settings.

In our branch-and-cut algorithm, we use framed capacity inequalities, homogeneous multistar inequalities, and large multistar inequalities. To keep our exposition concise, we restate the inequalities and refer to Letchford et al. (2002) and Lysgaard et al. (2004) for detailed information.

Homogeneous multistar inequalities are represented by a nucleus $S \subset V$, satellites $T \subset V \setminus S$, connectors $C \subset S$, and three integers A , B , and L and are given by

$$Bx(\delta(S)) - Ax(C, T) \geq L. \quad (38)$$

Generalized large multistar inequalities are represented by a nucleus $S \subset V$ and the remaining node set $\bar{S} := V \setminus S$ and are given by

$$Qx(N, N) + \sum_{j \in \bar{S}} q_j x(N, \{j\}) \leq Q|S| - \sum_{i \in S} q_i. \quad (39)$$

Framed capacity inequalities are defined by a frame $S \subseteq V$ and a partition $\Omega = \{S_1, \dots, S_p\}$ of S and are given by

$$x(\delta(S)) + \sum_{i=1}^p x(\delta(S_i)) \geq 2k(S, \Omega) + 2 \sum_{i=1}^p \left\lceil \sum_{i \in S_i} q_i / Q \right\rceil. \quad (40)$$

3.3. Specialized Model Constraints for A-MDmTSP

In this section, we study tour-size constraints and strengthened path elimination constraints for the A-MDmTSP.

3.3.1. Tour-Size Constraints. As mentioned previously, we can model the upper limit $l^{\max}$ on the tour size (i.e., the number of customers in a tour) by setting the customer demands equal to 1 and the vehicle capacity equal to $l^{\max}$. Furthermore, we propose the following simple constraints to impose the lower-bound restriction for multidepot routing problems:

$$x(\delta^+(S)) - \sum_{d \in D} \sum_{s \in S} (x_{ds} + x_{sd}) \geq 0 \quad (41)$$

for all $S \subset V$, $|S| \leq l^{\min}$.

On the other hand, D_k -type inequalities can also model the tour-size lower and upper limits. Consider a sequence $\{i_1, i_2, \dots, i_k, i_{k+1}\}$ in Theorem 1. From the construction of the left-hand side of (28), observe that the last node i_{k+1} can be either a depot or a customer node in a multidepot setting because there is exactly one incoming and one outgoing arc considered in (28) for i_{k+1} . Thus, we may replace i_{k+1} as the set of depots in the multidepot setting.

Theorem 4. Let $\{i_1, i_2, \dots, i_k\} \subseteq V$ be a sequence of distinct customer nodes. Then the inequalities

$$D_k^+ \text{-lim} : \sum_{s \in D} x_{i_1 s} + \sum_{s \in D} x_{s i_k} + \sum_{h=2}^k x_{i_h i_{h-1}} + 2 \sum_{h=2}^k x_{i_1 i_h} + \sum_{h=3}^k x(\{i_2, \dots, i_{h-1}\}, i_h) \leq k, \quad (42)$$

$$D_k^- \text{-lim} : \sum_{s \in D} x_{s i_1} + \sum_{s \in D} x_{i_k s} + \sum_{h=2}^k x_{i_{h-1} i_h} + 2 \sum_{h=2}^k x_{i_h i_1} + \sum_{h=3}^k x(i_h, \{i_2, \dots, i_{h-1}\}) \leq k \quad (43)$$

are valid cuts for the integer polytope of the A-MDmTSP for $k < l^{\min}$ or $k > l^{\max}$. Moreover, the constraints enforcing the lower and upper bounds on the tour size of the A-MDmTSP are formulated correctly by (42) or (43).

Proof. We prove that the proposed cuts are valid for the integer polytope of the A-MDmTSP by showing that the left-hand sides of (42) and (43) cannot exceed k for feasible solutions.

First, suppose $x_{i_1 i_h} = 0$ for all $h \in \{2, \dots, k\}$. As mentioned before, at most one incoming arc from depots and at most one outgoing arc to depots can be traversed in (42). In total, k customer nodes $\{i_1, \dots, i_k\}$ can have at most $k + 1$ traversed arcs on the left-hand side of (42) only if i_1 and i_k are connected to depots because of the construction of the left-hand side of (30). If i_1 and i_k are connected to different depots, then the tour is infeasible because tours must start and end at the same depot. Furthermore, if i_1 and i_k are connected to the same depot, the tour is still infeasible because this implies that the tour has size k with $k < l^{\min}$ or $k > l^{\max}$.

The remaining case where $x_{i_1 i_h} = 1$ for some $h \in \{2, \dots, k\}$ is shown as in the proof of Theorem 1 by considering i_{k+1} as the depot set D . We conclude that (42) is a VI for the integer polytope of the A-MDmTSP.

Now let us show that the tour-size constraints are formulated correctly by (42). By contradiction, assume that there is a tour $\{s, i_1, \dots, i_k, s\}$ with $s \in D$ and $i_1, \dots, i_k \in V$ such that $k < l^{\min}$ or $k > l^{\max}$. Consider the sequence $\{i_k, \dots, i_1\}$ in (42), and we obtain a contradiction. Thus, there is no tour with size $k < l^{\min}$ or $k > l^{\max}$. $\square$

Based on (42) and (43), we derive the following novel formulation of the A-MDmTSP.

Corollary 2. The A-MDmTSP is formulated correctly as min (18) subject to (19) and (21) and is completed with

- i. at least one of (28) and (29) to eliminate subtours,
- ii. at least one of (30) and (31) for depot fixing constraints, and
- iii. at least one of (42) and (43) for tour-size constraints.

Observe that the path elimination constraints (30) and (31) are redundant for the cases $k < l^{\min}$ and $k > l^{\max}$ if (42) and (43) are present, respectively. Thus, for example, when (42) and (30) are used together, it is sufficient to consider only (30) for the case $l^{\min} \leq k \leq l^{\max}$.

3.3.2. Path-Elimination Constraints. Without loss of generality, we assume that $l^{\min} \geq 2$ for the A-MDmTSP. Under this assumption, we can refine the path elimination constraints (26) further. This is summarized in the following proposition.

Proposition 4. Consider the A-MDmTSP with $l^{\min} \geq 2$. Then the path-elimination constraints (26) can be strengthened as follows:

$$\begin{aligned} \sum_{o \in I} x_{io} + \sum_{s \in D \setminus I} x_{sj} + \sum_{o \in I} x_{oi} + \sum_{s \in D \setminus I} x_{js} \\ + x(\gamma(S \cup \{i, j\})) \leq |S| + 2 \\ \text{for all } I \subset D, i, j \in V, S \subset V \setminus \{i, j\}. \end{aligned} \quad (44)$$

In other words, a feasible solution of the A-MDmTSP satisfies the above inequality.

Proof. Note that $\sum_{o \in I} x_{io} + \sum_{o \in I} x_{oi} \leq 1$ and $\sum_{s \in D \setminus I} x_{sj} + \sum_{s \in D \setminus I} x_{js} \leq 1$ hold because $l^{\min} \geq 2$. Moreover, $x(\gamma(S \cup \{i, j\})) \leq |S| + 1$ is a valid subtour elimination constraint. The equalities are attained for the above three inequalities only if the feasible solution contains a tour consisting of all nodes of $S \cup \{i, j\}$ and i, j are connected to different depots, which is infeasible. Thus, by combining the above three inequalities, we obtain (44). $\square$

Note that the above path elimination constraints do not remove tours with a single customer connected to different depots. Thus, we must add the following inequality:

$$\sum_{o \in I} x_{oh} + \sum_{o \in D \setminus I} x_{ho} \leq 1 \text{ for all } h \in V, I \subset D. \quad (45)$$

Now let us formulate the A-MDmTSP without using any D_k -type inequalities by including the above constraints.

Corollary 3. The A-MDmTSP is formulated correctly as min (18) subject to (19) and (21) and is completed with

- i. tour-size constraints (27) and (41) (subtours are also eliminated), and
- ii. path-elimination constraints (44) and (45).

4. Separation Algorithms

Efficient separation procedures for the model constraints and VIs introduced in Section 3 are crucial for the computational performance of the overall branch-and-cut framework. Failing to find violated inequalities quickly results in increased run times. In the following, we discuss the heuristic and exact procedures used for separating the VIs.

Let x^* be the linear programming (LP) solution for which the inequalities are required to be separated. We call x_a^* the capacity or weight of arc $a \in A$. Define $\mathcal{C}_s(x^*)$ as the set of *strongly* connected components of customer nodes in x^* . For each strongly connected component $C \in \mathcal{C}_s(x^*)$, a positive flow can be sent between all nodes $i, j \in C$. The set of *weakly* connected components $\mathcal{C}_w(x^*)$ of customer nodes equals the set

of strongly connected components on a modified graph $G' = (V, A')$, where each arc $(i, j) \in A'$ has capacity $x_{ij}^* := x_{ij}^* + x_{ji}^*$. For convenience, we refer to $\mathcal{C}_s(x^*)$ and $\mathcal{C}_w(x^*)$ by $\mathcal{C}_s$ and $\mathcal{C}_w$, respectively. Note that connected components are found using standard depth-first search methods.

Before discussing our new separation algorithms, we summarize briefly the separation algorithms described in the literature that are exploited in our branch-and-cut algorithm. The separation algorithms for the CAT inequalities (32) and the D_k^+ and D_k^- inequalities (28) and (29) are adapted versions of the separation procedures described by Fischetti and Toth (1997). For both the CAT inequalities and the D_k^+ and D_k^- inequalities, we begin by identifying the set of weakly connected components $\mathcal{C}_w$, after which separation is done for each component separately. For the CAT inequalities, we must impose our definition of arc incompatibility (see Section 3.2.1) so that the CAT inequalities remain valid. Regarding the D_k^+ and D_k^- inequalities, we refrain from using the heuristic procedure described by Fischetti and Toth (1997) and instead use the exact separation procedure to separate these inequalities, because its run time is negligible compared with the overall run time of the branch-and-cut algorithm.

The separation of D_k^+ and D_k^- tour-size constraints for the A-MDmTSP is done using the same procedure as that for the regular D_k^+ and D_k^- inequalities. However, we simply prune a partial tour as soon as the depth-first search method considers a tour larger than or equal to $l^{\min}$, and we automatically have a separation algorithm for the D_k^+ and D_k^- tour-size constraints. Note that $l^{\max}$ is enforced by setting the vehicle capacity equal to $l^{\max}$ and all customer demands q_i ($i \in V$) equal to 1.

Moreover, we separate the subtour elimination constraints (27), the large and homogeneous multistar inequalities, and the framed capacity inequalities (see Section 3.2.3) with the help of the procedures given by Lysgaard et al. (2004). However, these separation algorithms are tailored for capacitated routing problems with a single depot and a symmetric edge set. So that we can still use those separation algorithms, we modify the arc capacities to $\hat{x}_{ij}^* := x_{ij}^* + x_{ji}^*$. We ensure that $x_{ij} + x_{ji} < 1$ for all $i, j \in V$ by including explicitly all the subtour elimination constraints of size 2.

4.1. D_k^+ and D_k^- Depot Fixing Constraints

We now introduce the separation algorithm for the D_k depot constraints (30) as described by Theorem 2. We discuss only the separation of the D_k^+ depot constraint because the separation algorithm for the D_k^- depot constraint (31) can be obtained by swapping the indices.

The separation algorithm is based on a depth-first search with sophisticated pruning rules. This may sound similar to the separation of D_k^+ and D_k^- inequalities as described by Fischetti and Toth (1997), but the separation algorithm differs structurally as will be made clear in the following.

Let $\mathcal{J} = \{i_1, i_2, \dots, i_k\} \subseteq V$ be a sequence of distinct customer nodes. To be valid for the integer polytope of the A-MDVRP, it is sufficient to consider only sequences $\mathcal{J}$ for which $\sum_{s \in D} x_{i_1, s}^* + \sum_{s \in D} x_{s, i_k}^* > 1$. Furthermore, we consider only sequences $\mathcal{J}$ for which each customer is an element of the same connected component.

Observe that partitioning the set of depots into O and $O_2 := D \setminus O$ affects only the first two terms of (30) and is independent of the other terms. For a given LP solution x^* , we find the partitioning corresponding to the highest violation by greedily assigning depots into a set such that it contributes the most to the violation; that is, we assign a depot $s \in D$ to O if $x_{i_1, s}^* \geq x_{s, i_k}^*$ and to O_2 otherwise. For the optimal partitioning, we have

$$\sum_{s \in O} x_{i_1, s}^* + \sum_{s \in D \setminus O} x_{s, i_k}^* = \sum_{s \in D} \max\{x_{i_1, s}^*, x_{s, i_k}^*\}.$$

To find the sequence $\mathcal{J}$ with the largest violation $\phi(\mathcal{J})$, we use a depth-first search approach as in the case of the D_k inequalities without depots. For each weakly connected component $C \in \mathcal{C}_w$, we consider all pairs of customers nodes $i, j \in C$ for which $\sum_{s \in D} x_{i, s}^* + \sum_{s \in D} x_{s, j}^* > 1$. For any pair, we initialize the sequence $\mathcal{J} = \{i, j\}$ and explore all options iteratively under the restriction that node i remains first and node j remains last in the sequence. At each iteration, we calculate the degree of violation, and a VI is returned when a positive violation is observed.

We use a pruning rule to avoid enumerating all sequences completely. Let $\mathcal{B} = \{b_1, \dots, b_\eta\} \subseteq V \setminus \mathcal{J}$ be the sequence of η customer nodes that is inserted into $\mathcal{J}$ such that we obtain $\mathcal{J}^+ = \{i_1, \dots, i_{k-1}, b_1, \dots, b_\eta, i_k\}$. We let $\Delta(\mathcal{J} | \mathcal{B})$ denote the violation increase when $\mathcal{B}$ is inserted into $\mathcal{J}$. By deriving $\Delta(\mathcal{J} | \mathcal{B}) = \phi(\mathcal{J}^+) - \phi(\mathcal{J})$ explicitly, we obtain

$$\begin{aligned} \Delta(\mathcal{J} | \mathcal{B}) &= x_{i_k, b_\eta}^* + x_{b_1, i_{k-1}}^* - x_{i_k, i_{k-1}}^* - \eta \\ &\quad + \sum_{j=1}^{\eta} x_{i_1, b_j}^* + \sum_{h=1}^{k-1} \sum_{j=1}^{\eta} x_{i_h, b_j}^* + \sum_{h=1}^{\eta-1} \sum_{j=h+1}^{\eta} x_{b_h, b_j}^* \\ &\quad + \sum_{j=2}^{\eta} x_{b_h, b_{h-1}}^* + \sum_{h=1}^{\eta} x_{b_h, i_k}^*. \end{aligned} \quad (46)$$

Observe that the total weight of the incoming arcs into $\mathcal{B}$ is at most $x^*(V', \mathcal{B}) = \eta$ because $|\mathcal{B}| = \eta$.

Furthermore, $x^*(V', \mathcal{B}) = x^*(D, \mathcal{B}) + x^*(V, \mathcal{B})$ and thus $x^*(V, \mathcal{B}) = \eta - x^*(D, \mathcal{B})$. It follows that $x^*(V, \mathcal{B}) \leq \eta - \sum_{s \in D} x_{s, b_1}^*$. Consequently, we have

$$\begin{aligned} x_{i_k, b_\eta}^* + \sum_{h=1}^{k-1} \sum_{j=1}^{\eta} x_{i_h, b_j}^* + \sum_{h=1}^{\eta-1} \sum_{j=h+1}^{\eta} x_{b_h, b_j}^* + \sum_{h=2}^{\eta} x_{b_h, b_{h-1}}^* \\ \leq x^*(V, \mathcal{B}) \leq \eta - \sum_{s \in D} x_{s, b_1}^*. \end{aligned} \quad (47)$$

Substituting (47) into (46) gives

$$\Delta(\mathcal{J} | \mathcal{B}) \leq x_{b_1, i_{k-1}}^* - \sum_{s \in D} x_{s, b_1}^* - x_{i_k, i_{k-1}}^* + \sum_{j=1}^{\eta} x_{i_1, b_j}^* + \sum_{h=1}^{\eta} x_{b_h, i_k}^*. \quad (48)$$

The third term of the right-hand side of (48) is a known constant for any sequence $\mathcal{J}$. Furthermore, the fourth term (i.e., the sum of weights from node i_1 toward a node of the new inserted sequence $\mathcal{B}$) is at most $1 - \alpha$, where α is the current outgoing weight from node i_1 . The fifth term (i.e., the sum of weights from all nodes of $\mathcal{B}$ toward node i_k) is at most $1 - \beta$, where β is the current incoming weight into node i_k . For a given sequence $\mathcal{J}$, we have

$$\begin{aligned} \alpha(\mathcal{J}) &= \sum_{s \in D} x_{i_1, s}^* + \sum_{h=2}^k x_{i_1, i_h}^*, \\ \beta(\mathcal{J}) &= \sum_{s \in D} x_{s, i_k}^* + \sum_{h=1}^{k-1} x_{i_h, i_k}^*. \end{aligned}$$

Substituting the upper bounds $1 - \alpha(\mathcal{J})$ and $1 - \beta(\mathcal{J})$ into (25) gives

$$\begin{aligned} \Delta(\mathcal{J} | \mathcal{B}) &\leq x_{b_1, i_{k-1}}^* - \sum_{s \in D} x_{s, b_1}^* - x_{i_k, i_{k-1}}^* + 2 \\ &\quad - \alpha(\mathcal{J}) - \beta(\mathcal{J}). \end{aligned} \quad (49)$$

Upper bound (49) depends on only the known sequence $\mathcal{J}$ and the first node of the sequence $\mathcal{B}$. Thus, the upper bound can be exploited during the depth-first search to prune parts of the enumeration tree.

For a particular $\mathcal{J}$, let $\phi(\mathcal{J})$ be the current violation of (30). Moreover, let $\phi_{\max}$ be the largest violation of (30) found so far during the depth-first search. We know that if $\phi(\mathcal{J}) + \Delta(\mathcal{J} | \mathcal{B}) \leq \phi_{\max}$, then the customer sequence $\mathcal{J}$ and all its further extensions in the enumeration tree can be pruned. Using the upper bound (49) on $\Delta(\mathcal{J} | \mathcal{B})$, we obtain the following pruning rule:

$$\begin{aligned} x_{b_1, i_{k-1}}^* - \sum_{s \in D} x_{s, b_1}^* &\leq \phi_{\max} - \phi(\mathcal{J}) + \alpha(\mathcal{J}) + \beta(\mathcal{J}) \\ &\quad + x_{i_k, i_{k-1}}^* - 2. \end{aligned} \quad (50)$$

It is easy to incorporate this rule in the depth-first search by maintaining the values of $\alpha(\mathcal{J})$, $\beta(\mathcal{J})$, and $\phi(\mathcal{J})$ as follows:

$$\begin{aligned}\alpha(\mathcal{J}^+) &= \alpha(\mathcal{J}) + x_{i_1, b_1}^*, \\ \beta(\mathcal{J}^+) &= \beta(\mathcal{J}) + x_{b_1, i_k}^*, \\ \phi(\mathcal{J}^+) &= \phi(\mathcal{J}) + x_{i_k, b_1}^* + x_{b_1, i_{k-1}}^* - x_{i_k, i_{k-1}}^* + x_{i_1, b_1}^* \\ &\quad + x^*(\{i_1, \dots, i_{k-1}\}, b_1) + x_{b_1, i_k}^* - 1.\end{aligned}$$

4.2. Separating Path-Elimination Constraints

Separating the depot fixing constraints (26) and (44) has been done for symmetric edge sets by Belenguer et al. (2011). However, some structurally different steps must be taken for asymmetric arc sets. For completeness, we describe our separation procedure in full.

We begin by discussing the separation of inequalities (26), after which we discuss the changes required for separating inequalities (44). Recall that a violated inequality (26) is determined by two nodes $i, j \in V$, a subset $S \subset V \setminus \{i, j\}$, and a subset $I \subset D$.

For each $C \in \mathcal{C}_w$, we consider all pairs of customer nodes $i, j \in C$. Given i and j , we can determine $I \subseteq D$ such that $\sum_{o \in I} x_{oi}^* + \sum_{s \in D \setminus I} x_{js}^*$ is maximized. Note the following two points: (i) finding I is independent of finding $S \subset V$ such that $x(\gamma(S \cup \{i, j\}))$ is maximized, and (ii) if $x(\gamma(S \cup \{i, j\}))$ is smaller than unity, then there is no violated inequality (26) because we assume that the subtour elimination constraints are already satisfied.

Having determined candidate nodes i and j , we must find a subset $S \subset V$ to maximize $x(\gamma(S \cup \{i, j\}))$. We consider a modified graph $\tilde{G} = (\tilde{N}, \tilde{A})$ with $\tilde{N} = D \cup C_w \cup \{s^+\} \cup \{s^-\}$, where s^+ is an artificial source node and s^- is an artificial sink node. Let $\tilde{A} := \{(i, j) : i, j \in \tilde{N}\}$. Let c_{uv} be the arc capacity of any $(u, v) \in \tilde{A}$, where $c_{uv} := x_{uv}^* + x_{vu}^*$. Moreover, we create arcs from s^+ to both i and j , and we create arcs from each $o \in D$ to s^+ , all with a capacity that is large enough so that they are not selected in a minimum cut.

We then determine the minimum weighted (s^+, s^-) cut in graph $\tilde{G}$. All the nodes on the s^+ side of the minimum weighted cut will be part of the set S . If the found I , S , i , and j lead to a violated inequality (26), then we add it to the model.

In summary, this method is exact and finds for every i and j a depot fixing inequality with the largest violation in the form of inequality (26). The method can be adapted easily to separate constraints (44), because the only difference is how the sums corresponding to the connection of i and j to the depots are calculated, which are independent from the other aspects of the separation procedure. Therefore, no

structural changes are required except calculating the depot sums differently.

4.3. Comb Inequalities

Separation of ATSP comb inequalities (34) is done using the procedure due to Lysgaard et al. (2004) for finding strengthened comb inequalities. The procedure returns a handle $H \subset V'$ and a set of teeth $T_1, \dots, T_\pi \subset V'$. To check the returned inequalities for validity for the A-MDVRP, we verify whether $T_\ell \cup T_k = \emptyset$ for all $\ell, k \in \{1, \dots, \pi\}$ with $\ell \neq k$. As described in Proposition 1, we check which element contains the depot and insert all other depots in the same element. If (34) is violated, then we add the VI to the model.

To separate T -comb inequalities, we consider two different separation algorithms. First, we focus on T -comb inequalities with a single tooth T , which we refer to as $T1$ -comb inequalities. The second separation algorithm finds general T -comb inequalities (with multiple teeth). Both separation procedures are based on a greedy search to find violated $T1$ -comb inequalities, similar to that in Benavent and Martínez (2013) but with some minor adaptations. For completeness, we describe our heuristic separation procedure in full.

First, we obtain all weakly connected components $\mathcal{C}_w$, and in each of them we perform the following greedy search. For the $T1$ -comb inequalities, we search a tooth T that contains a single depot, and we add customers iteratively so that $\delta^+(x(T))$ is as small as possible. We stop when adding a customer results in $x(\delta^+(T)) < 2$. We initialize the handle $H = V \cap T$ and add customers iteratively so that $x(\delta^+(H))$ is as small as possible. Then, if $x(\delta^+(T)) + x(\delta^+(T)) < 4$, a violated $T1$ -comb inequality is found. Notice that we repeat this greedy search for each depot in each weakly connected component.

The T -comb inequalities are found by using a similar procedure, except that we must find multiple teeth instead of a single tooth. For each connected component, we create for each depot a tooth as for the $T1$ -comb inequalities. This results in a collection of teeth of size d , the number of depots.

Then the following procedure is repeated n_1 times. We take an arbitrary subset of teeth and check whether they (i) are not pairwise disjoint and (ii) span the complete connected component. In either case, no violated T -comb inequality can be found, and we continue with a new random selection of teeth.

What remains is to search for handle H . We initialize H with $\mathcal{C}_w \cap V$ and remove customers from H iteratively while checking for a violated T -comb inequality with the current set H and the randomly selected teeth. If no customers remain in H , then the search is terminated and we start over with a new random selection of teeth.

5. Branch-and-Cut Algorithm

Our branch-and-cut framework for solving A-MDVRPs consists of two simultaneously running algorithms. First, we have the branch-and-cut algorithm including the VIs and accompanying separation algorithms that are discussed throughout Sections 3 and 4. Second, we have designed a simple yet effective upper-bound procedure based on the δ -compact formulation P_δ .

Note that the following chosen parameter values and the order of calling VIs are the results of an extensive preliminary computational campaign. These values are kept fixed for all the different problem variants solved with the branch-and-cut framework. One might argue that using different parameter values for each problem variant would lead to better performance, but we chose not to do so because using parameter values that work well on all problem types enhances the general applicability of the branch-and-cut framework.

5.1. A Novel and Easy-to-Implement Upper-Bound Procedure

The upper-bound procedure is based on iteratively solving the δ -compact formulation P_δ as introduced in Section 2.1. Recall that this formulation is obtained from P_c by removing the $n - \delta$ most-expensive outgoing arcs to customers at each customer node.

For a given δ , we solve the δ -compact formulation with standard off-the-shelf commercial MIP solvers. We use Gurobi 8.0 because preliminary experiments showed that it provides high-quality upper bounds relatively quickly. When a new solution is found, we store it in a global variable that is accessible by the branch-and-cut algorithm (implemented in CPLEX 12.8). The branch-and-cut algorithm checks for improved upper bounds during its search, and whenever available it uses the solution provided by P_δ as the new incumbent solution.

The overall working of the upper-bound procedure is as follows. We solve δ -compact formulations iteratively, and we differentiate three different phases in our upper-bound procedure, where each phase is characterized by a series of values for δ and a maximum run time. In the first phase, we allow for a short run time for small values of δ to find an upper bound relatively quickly. In the second phase, we allow more time to solve P_δ , starting at the δ that has provided the best upper bound from the first phase. The third phase is characterized by long run times and large δ .

When we increase δ or change from one phase to the next, the new model is initialized with the best solution so far. Based on initial experiments, for the first phase we range δ from 5 to $\min\{10, |V|\}$ with a maximum run time of 20 s for each δ . In the second phase, we set the maximum run time to 300 s and let δ range

up to $\min\{25, |V|\}$. Finally, the run time is increased to 600 s, and we let δ range up to $\min\{50, |V|\}$.

Using the upper-bound procedure has three clear advantages. First, if the upper bound becomes sufficiently close to the lower bound, then variable fixing (i.e., pricing out variables by their reduced costs) will eliminate a significant number of variables, thereby decreasing the problem size. Second, the branch-and-bound tree will be smaller, making it easier and quicker to maintain. Third, CPLEX will normally dive into the branch-and-bound tree to find some incumbent solution. We noticed that this behavior of CPLEX is less expensive in terms of run time when the upper-bound procedure is used to provide new incumbent solutions.

5.2. Branch-and-Cut Implementation

The separation algorithms are coded in the user and lazy callback routines of CPLEX 12.8. The user callback is called for every fractional LP solution, whereas the lazy callback is called for every integer LP solution. Because the user callback is rather expensive in terms of run time, it is called only for the first N^{user} nodes of the branch-and-bound tree, and once these are processed it is called only once for every F^{user} branch-and-bound node. We examine the branch-and-bound nodes in a worst-bound-first manner, which implies that we call our separation algorithms on branch-and-bound nodes that determine the current global lower bound. Initial experiments showed that $N^{\text{user}} = 200$ and $F^{\text{user}} = 200$ are suitable values for all problem types.

The root node of the branch-and-bound tree is controlled additionally by the parameter N^{root} that specifies the maximum number of so-called cut loops (i.e., the number of times iterating between solving the LP relaxation and adding VIs). Although adding more VIs to the root node will increase the LP relaxation, it will also result in more complex rows of the simplex tableau that are notoriously more difficult to process efficiently, leading to increased run times for finding LP relaxation solutions in branch-and-bound nodes. Initial experiments showed that $N^{\text{root}} = 250$ is a suitable value for all problem types.

An overview of all the VIs included in the branch-and-cut algorithm is provided in Table 2. It specifies the order in which all separation routines are executed, which is kept fixed throughout all the numerical experiments presented herein. We employ a few common acceleration strategies to speed up the branch-and-cut algorithm. First, while calling the VIs in the order specified in Table 2, we terminate the callback procedure as soon as one of the VIs has found an inequality violating the current LP solution. Second, a VI is added only if it is violated by more than a predetermined threshold. Third, we limit the

Table 2. Branch-and-Cut Details Regarding the Order of Separation Procedures Being Called, Their Violation Threshold, and the Maximum Number of Added Inequalities per Call of the Separation Procedure

Order	Valid inequality	Threshold	Max number of cuts
1	Subtour elimination constraints (27)	0.1	100
2	Path-eliminating constraints (26) or (44)	0	*
3	D_k^+ and D_k^- depot constraints (30) and (31)	0	*
4	D_k^+ and D_k^- tour-size constraints (42) and (43)	0	*
5	CAT inequalities (32)	0	*
6	D_k^+ and D_k^- inequalities (28) and (29)	0	*
7	Framed capacity inequalities (40)	0.1	10
8	Homogeneous multistar inequalities (38)	0.1	10
9	Generalized large multistar inequalities (39)	0.1	10
11	ATSP-comb inequalities (35)	0.1	5
11	T1-comb inequalities (37)	0.1	5
12	T-comb inequalities (37)	0.1	5

Note. An asterisk indicates that there is no limit on the number of added inequalities.

number of VIs that we add during each call of their corresponding separation procedures.

In addition, we include all subtour elimination constraints of size two directly into our model formulation. We also consider subtour elimination constraints of sizes three and four, but including all of them would be computationally intractable. Therefore, for each customer node, we include the five vehicle subtours of shortest length. We add those to a pool of lazy constraints that are then checked automatically for violation at each integer LP solution. Moreover, we do not consider vehicle subtours of size four for instances with more than 100 customers, and we do not consider subtours of size three for instances with more than 200 customers. Finally, when solving A-MDTSP instances, we limit the maximum number of outgoing arcs from any depot to one.

6. Numerical Experiments

To assess the numerical performance of our branch-and-cut framework, we begin by comparing it with the (only yet existing) set of benchmark instances for A-MDVRPs designed by Bektaş et al. (2017). We then propose a new set of benchmark instances (Section 6.2) that we use to study the impact of the VIs on the root node (Section 6.3), to show the effectiveness of the upper-bound procedure (Section 6.4), and to examine the performance of the complete branch-and-cut framework (Section 6.5).

To study the effectiveness of the proposed branch-and-cut framework (i.e., the VIs and the upper-bound procedure), we compare it with a standard branch-and-cut algorithm consisting of subtour elimination constraints and the depot fixing constraints in the fashion of Laporte et al. (1988), which is a commonly observed strategy in the literature. (e.g., Belenguer et al. 2011) and can be considered state of the art. We refer to this standard branch-and-cut algorithm

without the upper-bound procedure and without the new VIs as the *basic algorithm*.

Our branch-and-cut framework is implemented in C++ and uses CPLEX 12.8 for the branch-and-cut algorithm and Gurobi 8.0 for the upper-bound procedure. All experiments are performed on an Intel Xeon E5 2680v3 CPU (2.5 GHz) with 24 GB of memory. Our framework uses six parallel running threads (which are commonly available on new desktop machines), of which five are assigned to the branch-and-cut algorithm and one to the upper-bound procedure. Besides the 24 GB of memory, we also allow both algorithms to store nodes on a hard drive with 50 GB of storage. Our implementation of the branch-and-cut framework used while writing this paper, the instances, and supporting documentation are available in the online supplement. Our latest C++ implementation of the algorithms (which might differ from the one used for this paper) and our benchmark instances are available at <https://doi.org/10.1287/opre.2020.2033>.

6.1. Comparison with Bektaş et al. (2017)

To the best of our knowledge, there is only one available set of benchmark instances that coincides exactly with a special case of the stylized *asymmetric* MDVRP that we consider in this study, namely, the A-MDTSP instances of Bektaş et al. (2017). We remark that there are ample benchmark instances available that consider problems with an A-MDVRP structure, but those require additional problem-specific constraints, which is considered to be beyond the present scope. For instance, Fischetti et al. (2007) studied crew and vehicle scheduling problems and Cappanera et al. (2013) studied asymmetric skill-based routing problems, which are both defined as an A-MDVRP with additional resource and path-elimination constraints.

Table 3. Results on the Benchmark Instances of Bektaş et al. (2017)

	Bektaş et al. (2017)			Branch-and-cut framework				
	Root	Obj.	Time (s)	Root	Obj.	Time A (s)	Time B (s)	ΔGap (%)
bgs-100-05-1a	865.00	879	5	879.00	879	5	3	100.0
bgs-100-10-1a	856.25	873	20	871.04	873	18	5	88.3
bgs-100-20-1a	848.85	861	25	857.72	861	11	5	73.0
bgs-100-05-2a	948.54	954	5	954.00	954	2	2	100.0
bgs-100-10-2a	945.37	952	5	952.00	952	2	2	100.0
bgs-100-20-2a	931.00	933	4	933.00	933	1	1	100.0
bgs-100-05-3a	893.41	904	13	904.00	904	10	4	100.0
bgs-100-10-3a	890.92	904	28	901.46	904	20	7	80.6
bgs-100-20-3a	871.07	879	30	877.02	879	8	5	75.0
bgs-200-10a	1,291.88	1,296	5	1,296.00	1,296	5	5	100.0
bgs-200-20a	1,290.14	1,295	5	1,295.00	1,295	5	4	100.0
bgs-200-40a	1,288.33	1,293	12	1,293.00	1,293	10	18	100.0
bgs-300-10a	1,609.54	*1,627	10,800	1,621.12	1,626	1,684	292	70.4
bgs-300-20a	1,604.09	1,622	10,387	1,615.56	1,622	1,921	220	64.0
bgs-300-30a	1,599.22	*1,617	10,800	1,610.64	1,617	2,387	173	64.2
bgs-300-40a	1,593.11	1,606	799	1,602.08	1,606	491	62	69.6
bgs-300-60a	1,583.37	1,592	533	1,590.72	1,592	868	166	85.2
Average	1,171.18	—	1,969	1,179.61	—	438	57	86.5

Notes. The results are taken from Bektaş et al. (2017) and are not reimplemented. The lower bounds of the two unsolved instances (marked with an asterisk symbol) are 1,624 and 1,614. The column ΔGap (%) shows the relative gap reduction between the root node relaxation and the optimal objective value of our framework compared with that of Bektaş et al. (2017). The columns time A and time B provide the run time in seconds that our framework requires when we use one or six cores, respectively. Note that on one core our framework alternates between the upper bounding procedure and the branch-and-cut algorithm.

The performance of our branch-and-cut framework on the instances of Bektaş et al. (2017) is shown in Table 3. For each instance, we present the root-node relaxation, the best upper bound, and the run time in seconds when our framework can use one or six cores (times A and B, respectively). Column ΔGap (%) presents the relative root-node gap reduction of our framework compared with the results presented by Bektaş et al. (2017), where we define the gap as the difference between the optimal objective value and the root-node relaxation. We remark that Bektaş et al. (2017) ran their experiments on a single core of an Intel i7-4790 CPU (3.6 GHz), which is faster than our Intel E5 2680v3 CPU (2.5 GHz). Moreover, when the framework runs on a single core, the computing capacity of this core is alternated between the upper-bound procedure and the branch-and-cut algorithm.

Our framework solves all instances within one hour (five minutes with six cores), thereby solving two instances that were not solved before (namely, bgs-300-10a and bgs-300-30a). We also improve the best known upper bound for instance bgs-300-30a, which is underlined and bold. The run times for small instances (up to 200 customers) are comparable, and for the larger instances we clearly outperform the algorithm of Bektaş et al. (2017). Furthermore, the developed VIs close the root-node gaps considerably for all instances (on average by 86.5% compared with Bektaş et al. (2017)), and eight instances are solved in the root node.

6.2. New Benchmark Instances

We consider three classes of asymmetric cost structures (called I, II, and III), which are in line with the ATSP instances of Fischetti and Toth (1997). Recall that c_{ij} is the traveling cost between nodes $i, j \in V'$, and let $U\{a; b\}$ denote the uniform integer distribution with support $\{a, a + 1, \dots, b - 1, b\}$. The three asymmetric cost-structure classes are generated as follows:

I. $c_{ij} \sim U\{1, 1,000\}$;

II. $c_{ij} := a_{ij} + b_{ij}$, where $a_{ij} = a_{ji} \sim U\{1; 1,000\}$ and $b_{ij} \sim U\{1; 20\}$;

III. $c_{ij} := d_{ij} + b_{ij}$, where d_{ij} is the floored Euclidean distance between i and j , and b_{ij} is as defined above.

For the class-III instances, we consider a $[0, 500] \times [0, 500]$ grid from which we draw the geographical locations of the nodes. For half of the nodes, we do this uniformly along the grid, whereas for the other half we create clusters of random sizes.

In total, we consider 12 problem categories, namely, four problem variants (A-MDTSP, A-MDmTSP, A-MDCVRP, and A-CLRP) each with three asymmetric cost structures (I, II, and III). We vary the number of customers and depots for each of the problem variants. The A-MDTSP instances have 200–400 customers with 30–50 depots, the A-MDmTSP instances have 50–100 customers with 5–20 depots, and the A-MDCVRP and A-CLRP instances have 40–85 customers and 5–20 depots. For each considered combination of the numbers of customers and depots, we create three replications.

Thus, we have 45×3 , 69×3 , 63×3 , and 63×3 instances for the A-MDTSP, A-MDmTSP, A-MDCVRP, and A-CLRP, respectively, giving a total of 720 instances. Each instance is named $XX_{n,r,x}$, where XX is the problem variant considered (i.e., A-MDTSP, A-MDmTSP, A-MDCVRP, or A-CLRP), n is the number of customers, r is the number of depots, and x is the replication index.

The other parameter values are drawn independently from a uniform distribution for each instance. For the A-MDmTSP, we draw the minimum tour length uniformly between 2 and 6 and the maximum tour length uniformly between 15 and 25. For the A-MDCVRP, we let the customer demand be random between 15 and 25, and the vehicle capacity is drawn uniformly between 150 and 300. For the A-CLRP, the capacity characteristics are equal to those of the A-MDCVRP, and the depot opening costs are generated uniformly between 500 and 3,000 for each instance. Finally, the number of vehicles per depot equals the number of customers, except for the A-MDTSP instances where it equals one by definition. An overview of all the instances and the detailed results of solving these with the basic algorithm and the branch-and-cut framework are provided in the appendix in Tables 8–15.

6.3. Valid Inequalities and Effect on Root Node

To examine the effect of the proposed VIs on the root-node relaxation, we consider six combinations of the VIs discussed in this study. To provide a fair comparison of the impact of the inequalities, we disable all standard CPLEX cuts that would otherwise be added by default. We remark that similar results are observed when these cuts are enabled.

Table 4 presents the average root-node relaxation when only the path-elimination constraints due to Laporte et al. (1988) are used (column *Path*), when

only the D_k depot inequalities are used (column D_k depot), and when both are used (column *Both*). To show the effect of all other inequalities discussed in this paper, we include the root-node results with and without separating the other inequalities for each of the three different depot fixing methods (referred to as with/without other VIs). Next, we define the root-node gap as the difference between the best known solution and the root-node relaxation. Column ΔGap (%) shows the average and maximum percentages of the root-node optimality gap reduction when we use our complete branch-and-cut framework (i.e., “With other VIs (Both)”) instead of the basic algorithm (i.e., “Without other VIs (Path)”).

Adding the proposed VIs reduces root-node optimality gaps by 16.5%, 8.3%, 7.8%, and 0.4% on average for the A-MDTSP, A-MDmTSP, A-MDCVRP, and A-CLRP instances, respectively. Furthermore, the asymmetry class affects the effectiveness of the proposed VIs considerably: the root-node gap is reduced on average by 1.5%, 9.5%, and 13.8% for the asymmetry classes I, II, and III, respectively. Thus, for the first three problem types, the effect of the added VIs is clearly visible in the root-node optimality gaps, whereas for the last problem type the strength of the VIs becomes clearer in the complete run of the branch-and-cut framework (discussed in Section 6.5). This is related mainly to the large root-node optimality gaps of the A-CLRP instances (mostly 20%–30%) and for the A-MDTSP-I (mostly larger than 75% because of the lack of reasonable upper bounds), whereas the other problem types typically have a root-node optimality gap smaller than 10%.

In addition, using the D_k depot inequalities instead of the path-elimination constraints results in higher root-node relaxations for 715 out of 720 instances. Including all other VIs (without vs. with other VIs) does (almost) not affect the difference between the

Table 4. Average Root Node Lower Bounds for Various Combinations of VIs

Type	Asy.	Without other VIs			With other VIs			ΔGap (%)	
		Path	D_k depot	Both	Path	D_k depot	Both	Avg.	Max.
A-MDTSP	I	1,395.0	1,395.6	1,395.6	1,395.4	1,396.0	1,396.0	1.6	17.4
	II	4,025.0	4,027.3	4,027.3	4,026.7	4,028.9	4,028.9	20.7	67.2
	III	7,715.8	7,729.3	7,729.3	7,734.1	7,747.1	7,747.1	27.2	49.4
	Avg.	4,378.6	4,384.1	4,384.1	4,385.4	4,390.7	4,390.7	16.5	—
A-MDmTSP	I	1,485.4	1,485.8	1,485.9	1,486.1	1,486.6	1,486.6	1.4	14.5
	II	2,530.9	2,537.6	2,537.7	2,533.3	2,540.0	2,540.0	10.1	32.3
	III	3,722.9	3,738.5	3,738.5	3,732.6	3,748.0	3,748.0	13.5	35.8
	Avg.	2,579.7	2,587.3	2,587.4	2,584.0	2,591.5	2,591.5	8.3	—
A-MDCVRP	I	1,486.7	1,488.9	1,489.0	1,488.0	1,490.2	1,490.3	2.5	28.3
	II	2,388.1	2,396.1	2,396.1	2,392.7	2,400.7	2,400.8	6.7	30.0
	III	3,482.1	3,497.5	3,497.5	3,493.9	3,507.0	3,507.0	14.3	50.5
	Avg.	2,452.3	2,460.8	2,460.9	2,458.2	2,466.0	2,466.0	7.8	—
A-CLRP	I	2,565.8	2,566.4	2,566.4	2,566.9	2,567.5	2,567.5	0.4	7.0
	II	3,495.3	3,497.2	3,497.2	3,497.0	3,498.6	3,498.6	0.4	1.6
	III	4,318.7	4,319.9	4,319.9	4,324.9	4,325.8	4,325.8	0.5	2.0
	Avg.	3,459.9	3,461.2	3,461.2	3,462.9	3,464.0	3,464.0	0.4	—

columns *Path* and D_k depot. We conclude that the D_k depot inequalities are computationally stronger than the traditional path-elimination constraints, but the two inequalities are not mutually redundant. Therefore, we choose to use both the traditional path-elimination constraints and the D_k depot constraints in the branch-and-cut framework.

To show the notable performance of the proposed VIs for some particular instances, we focus on instance A-MDTSP-II_20_200_1 with optimal objective value 3,240. Adding all proposed inequalities increases the root-node relaxation from 3,227.4 to 3,235.9, that is, the root-node optimality gap is closed by 67.2%. We remark that both algorithms solve the root node within 1 s.

6.4. Effectiveness of Upper-Bound Procedure

To show that the upper-bound procedure is effective, we compare the upper-bound performance of the basic algorithm with that of the branch-and-cut framework (with and without using the upper-bound procedure described in Section 5.1). We examine the average upper bound that the three methods find on a subset of instances, where we set the objective of instances without upper bound equal to 15,000. Two arbitrarily selected sets of instances are discussed because the results for all other problem variants are comparable to one of these.

Figure 3 (left) shows the results on the A-MDTSP instances with asymmetry class I. We see that the basic algorithm (black line) has difficulty finding feasible solutions and that adding the VIs (red line) does not ease the process of finding feasible solutions. We also observe that the framework with the upper-bound procedure (blue line) finds feasible solutions quickly in the beginning and that the upper bound continues to improve gradually over time.

Figure 3 (right) shows the results on the A-MDTSP instances with asymmetry class III. The basic algorithm (black line) is slightly faster at finding the first feasible solutions than is the framework without the upper-bound procedure (red line), but the latter has a better upper-bound performance over the whole run. The reason, which will be addressed in the next section, is that the basic algorithm runs out of memory, which is avoided by adding the VIs. Moreover, although initially the framework with the upper-bound procedure is slower at finding feasible solutions, we do observe that it finds the best upper bounds over the whole run.

We conclude that adding the upper-bound procedure to the branch-and-cut framework is efficient and results in a robust upper-bound performance of the branch-and-cut framework.

6.5. Performance of Branch-and-Cut Framework

We solved all benchmark instances of the 12 problem categories with both the branch-and-cut framework and the basic algorithm. Recall that the basic algorithm can be considered as what is known in the literature for solving the A-MDVRP, namely, it is a branch-and-cut algorithm with subtour elimination constraints and the traditional path-elimination constraints.

Tables 5–7 present various summary statistics regarding the basic algorithm, the branch-and-cut framework, and a comparison of both, respectively. For detailed information about an individual instance level, we refer to Tables 8–15 in the appendix. In Tables 5 and 6, the columns headed with $\#_{\leq x\%}$ indicate the number of instances with an optimality gap smaller than $x\%$, *LB* is the average lower bound of all instances, *Time* is the average run time of all instances, and *Nodes* is the number of processed branch-and-bound nodes.

Figure 3. (Color online) The Average Upper Bound over Time on the A-MDTSP Instances with Asymmetry Class I (Left) and the Asymmetry Class III (Right) for the Basic Algorithm, the Branch-and-Cut Framework, and the Branch-and-Cut Framework Without the Upper-Bound Procedure

Note. If no upper bound for a particular instance was available, we set it equal to 15,000.

Table 5. Summary Statistics of the Performance of the Basic Algorithm

Type	Asy.	#Inst	N_{sol}	$\#_{\leq 0\%}$	$\#_{\leq 1\%}$	$\#_{\leq 5\%}$	LB	Time (s)	Nodes (10^3)
A-MDTSP	I	45	14	7	8	10	1,403.82	15,358*	109
	II	45	41	34	39	40	4,041.23	5,579*	47
	III	45	45	28	44	45	7,819.48	7,834*	138
A-MDmTSP	I	69	46	30	30	33	1,535.71	10,893	79
	II	69	65	45	50	53	2,605.86	7,604	82
	III	69	69	48	50	61	3,883.00	6,273	81
A-MDCVRP	I	63	51	31	31	35	1,567.53	10,736	103
	II	63	44	18	19	20	2,469.99	13,270	95
	III	63	62	41	44	57	3,635.97	7,375	138
A-CLRP	I	63	55	35	35	36	2,986.40	9,227	54
	II	63	48	29	31	32	4,264.81	1,0293	54
	III	63	52	24	25	29	5,380.55	1,2080	131

Notes. N_{sol} gives the number of instances for which a feasible solution is found, $\#_{\leq x\%}$ indicates the number of instance with an optimality gap smaller than or equal to $x\%$, and $nodes$ is the number of processed branch-and-bound nodes.

*We remark that 33, 9, and 15 out of 45 A-MDTSP instances stopped prematurely by reaching the memory limit.

Table 6. Summary Statistics of the Performance of the Branch-and-Cut Framework

Type	Asy.	#Inst	$\#N_{\text{sol}}$	$\#_{\leq 0\%}$	$\#_{\leq 1\%}$	$\#_{\leq 5\%}$	LB	Time (s)	Nodes (10^3)
A-MDTSP	I	45	45	12	12	18	1,412.57	13,926*	155
	II	45	45	40	40	44	4,043.93	3,975*	21
	III	45	45	45	45	45	7,831.82	2,152	13
A-MDmTSP	I	69	69	41	44	57	1,557.17	7,996	84
	II	69	69	59	59	66	2,617.60	4,016	65
	III	69	69	55	61	69	3,898.03	4,599	85
A-MDCVRP	I	63	63	37	39	50	1,592.91	8,049	105
	II	63	63	26	28	44	2,517.48	11,419	155
	III	63	63	44	53	63	3,652.29	6,437	148
A-CLRP	I	63	63	43	46	58	3,018.52	6,993	66
	II	63	63	36	42	54	4,391.30	8,389	84
	III	63	63	25	26	39	5,580.83	11,383	133

Notes. N_{sol} gives the number of instances for which a feasible solution is found, $\#_{\leq x\%}$ indicates the number of instance with an optimality gap smaller than or equal to $x\%$, and $Nodes$ is the number of processed branch-and-bound nodes.

*We remark that 18 and 1 out of 45 A-MDTSP instances stopped prematurely by reaching the memory limit.

In Table 7, *Time* denotes the average run time for instances that are solved to optimality by both algorithms, *Gap* presents the average gap for instances that are not solved to optimality by one of the algorithms, and *LB* is the average lower bound for the remaining instances (i.e., instances for which at least one algorithm did not find a feasible solution).

The key observation is that our branch-and-cut framework solves considerably more instances to optimality for each problem type. For instance, our framework solves all 45 A-MDTSP-III instances to optimality, whereas the basic algorithm solves only 28 instances. The main reason is that the branch-and-cut framework provides higher lower bounds than does the basic algorithm, which is observed for all 12 problem categories.

Second, the basic algorithm has difficulty finding feasible solutions, which our framework overcomes

effectively by using the upper-bound procedure described in Section 5.1. The basic algorithm fails to find feasible solutions for 35, 27, 32, and 34 instances of the A-MDTSP, A-MDmTSP, A-MDCVRP, and A-CLRP, respectively, whereas the branch-and-cut framework finds feasible solutions to all instances. This is exemplified also by the number of instances with relatively small optimality gaps. Using the branch-and-cut framework instead of the basic algorithm increases the number of instances with optimality gaps smaller than 5% from 95 to 107, from 147 to 192, from 112 to 157, and from 97 to 151 for the four problem variants, respectively. The average optimality gap for the instances for which both algorithms find a feasible solution but that were not solved to optimality is 2.1% for the branch-and-cut framework and 63.3% for the basic algorithm, as shown in Table 7.

Table 7. Summary Statistics Comparing Both Algorithms

Type	Asy.	Basic algorithm			Branch-and-cut framework		
		Time (s)	Gap (%)	LB	Time (s)	Gap (%)	LB
A-MDTSP	I	1,017	23.2	1,396.7	263	1.5	1,407.6
	II	1,560	3.3	4,329.4	914	3.8	4,345.4
	III	1,662	0.6	—	777	0.0	—
A-MDmTSP	I	1,653	199.1	1,586.5	92	1.1	1,628.3
	II	2,059	98.8	2,659.7	262	0.8	2,705.7
	III	1,142	6.7	—	442	0.9	—
A-MDCVRP	I	3,239	157.2	1,545.0	374	3.2	1,596.0
	II	1,446	111.6	2,634.2	122	2.8	2,700.7
	III	1,674	5.0	4,253.9	1,146	1.2	4,339.7
A-CLRP	I	2,209	33.1	3,443.2	325	1.5	3,536.3
	II	1,257	49.2	4,638.2	317	2.0	4,999.8
	III	2,459	28.5	5,749.1	1,315	5.3	6,196.9
Avg.		1,815	63.3	2,663.6	550	2.1	2,777.7

Notes. Time is the average time of the instances that are solved by both algorithms, Gap the average gap of the instances where both algorithms found an upper bound but at least one did not prove optimality, and LB the average lower bound of the remaining instances, that is, at least one algorithm did not find an upper bound.

Comparing the run times of the basic algorithm and the branch-and-cut framework, we see that the branch-and-cut framework is on average faster than the basic algorithm on each of the 12 problem categories. We also observe that the difference in run-time performance is affected highly by the problem type and the class of asymmetry. For instance, on the 30 A-MDmTSP-I instances that are solved by both algorithms, the framework is on average almost 20 times faster (92 s compared with 1,653 s), whereas on the 48 A-MDmTSP-III instances, the framework is only 2.5 times faster. On average, the run time of the instances solved to optimality by both algorithms is 550 s for the branch-and-cut framework and 1,815 s for the basic algorithm, as denoted in Table 7.

In addition, because of adding all the VIs, the branch-and-bound tree is searched more effectively. This has two major consequences. First, for some problem categories, this leads to a reduction in the number of processed branch-and-bound nodes. For instance, a reduction from 138×10^3 to 13×10^3 is seen for problem category A-MDTSP-III. Second, searching the branch-

and-bound tree more effectively may result in the pruning of more branch-and-bound nodes and thereby the ability to process more nodes. In other words, memory is used more efficiently, which is exemplified by the number of instances stopped before the time limit because of reaching the memory limit of 75 GB (57 vs. 19).

Finally, the computational performance is clearly different among the asymmetry classes. Table 7 shows that the average gap on the instances that are not solved to optimality reduces especially on asymmetry class I (e.g., from 199.1% to 1.1% on the A-MDmTSP-I instances). Moreover, there seems to be no clear ordering in the difficulty of the asymmetry classes over the four considered problem variants. However, within each problem variant, a clear ordering in difficulty is observed. For the A-MDTSP and A-MDmTSP, it holds that, when distances are closer to the Euclidean one, both algorithms can provide higher-quality solutions. For the A-MDCVRP, this effect is not as clear because asymmetry class II seems to be most difficult there. Finally, for the A-CLRP class, the Euclidean arc costs with noise (i.e., class III) seem to be most difficult.

Table 8. Branch-and-Cut Framework Solutions to the A-MDTSP Instances

Instance A-MDTSP	I					II					III				
	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)
_200_20_0	1,395	1,395	0.00	1	81	3,094	3,094	0.00	0	44	6,320	6,320	0.00	2	182
_200_20_1	1,392	1,392	0.00	3	96	3,240	3,240	0.00	0	26	5,896	5,896	0.00	0	351
_200_20_2	1,574	1,574	0.00	35	407	3,567	3,567	0.00	0	61	5,748	5,748	0.00	0	152
_200_30_0	1,348	1,348	0.00	296	5,801	3,313	3,313	0.00	0	42	5,836	5,836	0.00	0	80
_200_30_1	1,437	1,437	0.00	12	187	3,470	3,470	0.00	0	53	6,102	6,102	0.00	0	95
_200_30_2	1,357	1,357	0.00	157	2,578	3,112	3,112	0.00	0	53	5,927	5,927	0.00	3	182
_200_40_0	1,464	1,644	12.31	397	18,000	3,149	3,149	0.00	4	246	5,861	5,861	0.00	0	78

Table 8. (Continued)

Instance A-MDTSP	I					II					III				
	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)
_200_40_1	1,472	1,497	1.67	656	18,000	3,450	3,450	0.00	9	305	6,131	6,131	0.00	3	156
_200_40_2	1,421	1,421	0.00	816	14,560	3,227	3,227	0.00	0	22	6,526	6,526	0.00	0	143
_300_20_0	1,542	1,542	0.00	3	339	3,676	3,676	0.00	0	136	7,660	7,660	0.00	0	1,351
_300_20_1	1,474	1,474	0.00	3	298	3,747	3,747	0.00	0	128	7,704	7,704	0.00	0	444
_300_20_2	1,401	1,401	0.00	17	430	3,833	3,833	0.00	0	139	7,535	7,535	0.00	2	751
_300_30_0	1,492	1,577	5.71	288	18,000	3,848	3,848	0.00	0	327	7,356	7,356	0.00	0	284
_300_30_1	1,526	1,571	2.97	524	18,000	4,245	4,245	0.00	5	825	7,565	7,565	0.00	3	462
_300_30_2	1,516	1,550	2.23	274	18,000	3,895	3,895	0.00	3	666	7,282	7,282	0.00	1	564
_300_40_0	1,467	1,636	11.56	144	18,000	4,247	4,247	0.00	10	1,220	7,277	7,277	0.00	0	217
_300_40_1	1,462	3,775	158.16	141	17,317	3,885	3,885	0.00	194	10,188	7,440	7,440	0.00	0	730
_300_40_2	1,364	1,971	44.50	147	18,000	3,943	3,943	0.00	16	1,411	7,323	7,323	0.00	19	1,169
_350_30_0	1,410	1,410	0.00	64	4,047	4,339	4,339	0.00	22	3,530	8,047	8,047	0.00	2	460
_350_30_1	1,360	29,717	2,085.31	110	15,453	4,152	4,152	0.00	1	295	8,169	8,169	0.00	1	1,322
_350_30_2	1,329	1,384	4.16	254	18,000	4,361	4,361	0.00	2	878	8,373	8,373	0.00	6	1,546
_350_40_0	1,295	15,111	1,067.23	107	14,108	4,250	4,250	0.00	77	15,537	8,253	8,253	0.00	84	5,988
_350_40_1	1,374	6,648	383.95	103	17,533	4,199	4,199	0.00	25	3,392	8,642	8,642	0.00	18	2,714
_350_40_2	1,357	8,599	533.61	103	17,605	4,082	4,082	0.00	3	837	8,214	8,214	0.00	7	1,187
_350_50_0	1,434	18,681	1,202.92	100	17,797	4,132	4,132	0.00	7	1,186	8,169	8,169	0.00	1	662
_350_50_1	1,483	7,051	375.48	102	17,286	4,245	4,245	0.00	3	1,080	7,843	7,843	0.00	0	2,120
_350_50_2	1,349	15,594	1,055.79	99	18,000	4,351	4,513	3.73	107	18,000	8,144	8,144	0.00	12	1,411
_375_30_0	1,472	3,363	128.46	95	14,744	4,163	4,163	0.00	2	1,340	8,306	8,306	0.00	21	3,917
_375_30_1	1,440	1,440	0.00	39	3,840	4,212	4,212	0.00	0	284	8,516	8,516	0.00	2	1,616
_375_30_2	1,429	8,531	496.94	93	15,939	4,393	4,393	0.00	0	826	8,668	8,668	0.00	20	3,387
_375_40_0	1,397	1,556	11.38	143	18,000	4,557	4,557	0.00	29	7,481	8,550	8,550	0.00	2	1,498
_375_40_1	1,440	4,788	232.45	90	14,078	4,346	4,346	0.00	7	1,898	8,850	8,850	0.00	0	1,183
_375_40_2	1,375	3,501	154.71	90	15,695	4,301	4,301	0.00	91	13,422	8,388	8,388	0.00	57	7,145
_375_50_0	1,419	1,638	15.47	98	18,000	4,310	4,310	0.00	0	116	8,885	8,885	0.00	1	742
_375_50_1	1,265	2,765	118.57	80	15,339	4,284	4,429	3.39	82	15,603	8,602	8,602	0.00	11	2,250
_375_50_2	1,353	21,952	1,522.36	86	18,000	4,298	4,298	0.00	76	9,953	8,549	8,549	0.00	76	7,998
_400_30_0	1,352	1,570	16.16	155	18,000	4,578	4,578	0.00	4	1,385	8,802	8,802	0.00	31	7,652
_400_30_1	1,437	12,767	788.65	82	12,945	4,429	4,429	0.00	9	2,183	9,311	9,311	0.00	49	9,334
_400_30_2	1,305	1,346	3.13	309	18,000	4,327	4,327	0.00	0	919	9,028	9,028	0.00	17	2,820
_400_40_0	1,420	1,445	1.78	258	18,000	4,534	4,731	4.35	52	18,000	8,539	8,539	0.00	3	1,550
_400_40_1	1,464	2,412	64.72	79	14,354	4,534	4,534	0.00	1	1,863	9,053	9,053	0.00	73	10,639
_400_40_2	1,404	7,563	438.59	80	15,330	4,446	5,421	21.92	35	18,000	8,766	8,766	0.00	31	3,769
_400_50_0	1,378	9,337	577.60	77	15,075	4,277	4,277	0.00	3	2,188	8,848	8,848	0.00	8	1,984
_400_50_1	1,391	3,417	145.65	80	15,859	4,497	4,656	3.53	53	18,000	8,520	8,520	0.00	5	1,710
_400_50_2	1,332	11,916	794.84	78	16,844	4,439	4,439	0.00	7	2,358	8,908	8,908	0.00	11	2,828

Table 9. Basic Algorithm Solutions to the A-MDTSP Instances

Instance A-MDTSP	I					II					III				
	LB	UB	Gap	Nodes	Time (s)	LB	UB	Gap	Nodes	Time (s)	LB	UB	Gap	Nodes	Time (s)
_200_20_0	1,395	1,395	0.00	13	323	3094	3094	0.00	0	25	6,320	6,320	0.00	81	662
_200_20_1	1,392	1,392	0.00	16	317	3240	3240	0.00	0	26	5,896	5,896	0.00	7	81
_200_20_2	1,574	1,574	0.00	60	607	3567	3567	0.00	1	41	5,748	5,748	0.00	3	58
_200_30_0	1,333	2,513	88.58	224	18,000	3313	3313	0.00	0	22	5,836	5,836	0.00	6	74
_200_30_1	1,437	1,437	0.00	149	3,070	3470	3470	0.00	1	48	6,102	6,102	0.00	13	142
_200_30_2	1,346	1,430	6.24	336	13,957	3112	3112	0.00	1	51	5,927	5,927	0.00	97	1,019
_200_40_0	1,444	—	—	229	16,887	3149	3149	0.00	24	413	5,861	5,861	0.00	4	63
_200_40_1	1,453	—	—	209	17,368	3450	3450	0.00	36	584	6,131	6,131	0.00	106	940
_200_40_2	1,404	—	—	187	17,502	3227	3227	0.00	0	45	6,526	6,526	0.00	13	110
_300_20_0	1,542	1,542	0.00	18	349	3676	3676	0.00	0	89	7,660	7,660	0.00	56	1,116
_300_20_1	1,474	1,474	0.00	19	507	3,747	3,747	0.00	0	45	7,704	7,704	0.00	3	166
_300_20_2	1,401	1,401	0.00	59	1,941	3,833	3,833	0.00	3	172	7,535	7,535	0.00	34	842
_300_30_0	1,479	—	—	115	4,649	3,848	3,848	0.00	6	346	7,356	7,356	0.00	14	259
_300_30_1	1,517	1,882	24.10	116	7,908	4,245	4,245	0.00	19	628	7,565	7,565	0.00	107	1,969
_300_30_2	1,504	—	—	121	5,742	3,895	3,895	0.00	9	303	7,282	7,282	0.00	142	2,307
_300_40_0	1,456	—	—	117	5,332	4,247	4,247	0.00	52	2,055	7,277	7,277	0.00	6	184

Table 9. (Continued)

Instance A-MDTSP	I					II					III				
	LB	UB	Gap	Nodes	Time (s)	LB	UB	Gap	Nodes	Time (s)	LB	UB	Gap	Nodes	Time (s)
_300_40_1	1,452	—	—	115	6,513	3,876	3,889	0.33	168	11,636	7,440	7,440	0.00	56	867
_300_40_2	1,354	—	—	112	7,135	3,943	3,943	0.00	89	3,235	7,294	7,339	0.61	221	6,216
_350_30_0	1,404	1,916	36.47	96	8,456	4,339	4,339	0.00	99	5,505	8,047	8,047	0.00	92	1,832
_350_30_1	1,351	—	—	90	4,081	4,152	4,152	0.00	2	271	8,169	8,169	0.00	190	4,395
_350_30_2	1,324	1,343	1.47	217	18,000	4,361	4,361	0.00	9	436	8,373	8,373	0.00	100	3,531
_350_40_0	1,288	—	—	166	18,000	4,229	—	—	91	5,632	8,230	8,262	0.39	202	9,147
_350_40_1	1,359	—	—	89	4,043	4,199	4,199	0.00	94	5,586	8,603	8,659	0.66	159	5,799
_350_40_2	1,347	—	—	88	5,595	4,082	4,082	0.00	62	5,237	8,187	8,224	0.45	384	9,233
_350_50_0	1,419	—	—	88	6,838	4,127	4,142	0.37	124	9,580	8,169	8,169	0.00	99	2,492
_350_50_1	1,474	—	—	87	7,995	4,245	4,245	0.00	83	7,111	7,843	7,843	0.00	7	321
_350_50_2	1,341	—	—	88	9,994	4,335	—	—	91	5,957	8,101	8,161	0.74	147	5,062
_375_30_0	1,463	—	—	80	5,040	4,163	4,163	0.00	8	438	8,282	8,312	0.37	130	5,943
_375_30_1	1,438	1,446	0.57	160	16,815	4,212	4,212	0.00	1	176	8,516	8,516	0.00	30	653
_375_30_2	1,418	—	—	79	4,846	4,393	4,393	0.00	3	231	8,637	8,687	0.58	156	6,967
_375_40_0	1,386	—	—	79	4,610	4,548	4,734	4.10	99	16,992	8,550	8,550	0.00	45	1,197
_375_40_1	1,433	—	—	80	4,301	4,346	4,346	0.00	45	2,858	8,850	8,850	0.00	126	4,230
_375_40_2	1,367	—	—	79	4,042	4,293	4,314	0.50	114	14,538	8,324	8,402	0.93	124	5,179
_375_50_0	1,405	—	—	79	5,141	4,310	4,310	0.00	6	385	8,885	8,885	0.00	248	5,336
_375_50_1	1,253	—	—	79	5,518	4,269	—	—	81	11,780	8,566	8,616	0.59	149	4,344
_375_50_2	1,340	—	—	79	4,812	4,290	4,311	0.48	148	18,000	8,514	8,575	0.72	108	4,679
_400_30_0	1,347	—	—	181	18,000	4,578	4,578	0.00	77	7,437	8,794	8,803	0.10	577	18,000
_400_30_1	1,430	—	—	72	3,269	4,429	4,429	0.00	29	1,424	9,264	9,345	0.88	104	4,677
_400_30_2	1,300	1,365	4.97	199	18,000	4,327	4,327	0.00	6	554	9,021	9,032	0.13	489	17,728
_400_40_0	1,412	—	—	71	4,250	4,529	4,556	0.59	175	18,000	8,539	8,539	0.00	284	6,373
_400_40_1	1,457	—	—	71	5,130	4,534	4,534	0.00	20	1,046	8,992	9,104	1.25	96	3,757
_400_40_2	1,395	—	—	70	6,355	4,433	5,189	17.06	84	6,668	8,726	8,774	0.55	115	4,884
_400_50_0	1,370	—	—	69	7,329	4,277	4,277	0.00	52	4,992	8,824	8,862	0.43	167	4,006
_400_50_1	1,380	—	—	70	5,618	4,484	—	—	71	6,578	8,505	8,530	0.29	644	18,000
_400_50_2	1,318	—	—	69	7,743	4,439	4,439	0.00	34	1,211	8,908	8,908	0.00	235	5,303

Table 10. Branch and Cut Framework Solutions to A-MDmTSP Instances

Instance A-MDmTSP	I					II					III				
	LB	UB	Gap	Nodes	Time (s)	LB	UB	Gap	Nodes	Time (s)	LB	UB	Gap	Nodes	Time (s)
_50_5_0	1,294	1,294	0.00	0	4	1,893	1,893	0.00	0	3	2,856	2,856	0.00	0	9
_50_5_1	1,187	1,187	0.00	0	8	2,487	2,487	0.00	0	3	2,901	2,901	0.00	2	14
_50_5_2	1,356	1,356	0.00	1	12	1,938	1,938	0.00	0	1	3,286	3,286	0.00	9	42
_60_5_0	1,736	1,736	0.00	16	209	2,613	2,613	0.00	8	40	3,250	3,250	0.00	90	525
_60_5_1	1,636	1,636	0.00	24	192	2,424	2,424	0.00	0	5	3,431	3,431	0.00	0	14
_60_5_2	1,650	1,650	0.00	107	1,206	2,507	2,507	0.00	14	113	3,883	3,883	0.00	169	5,070
_70_5_0	1,956	1,997	2.07	166	18,000	2,568	2,568	0.00	98	1,324	3,601	3,601	0.00	1	42
_70_5_1	1,428	1,428	0.00	8	86	2,661	2,661	0.00	21	235	3,511	3,529	0.51	384	18,000
_70_5_2	1,651	1,651	0.00	17	112	2,757	2,757	0.00	0	10	3,559	3,559	0.00	0	7
_75_5_0	1,630	1,630	0.00	1	22	2,590	2,590	0.00	2	18	3,851	3,851	0.00	7	105
_75_5_1	1,802	1,802	0.00	1	25	2,744	2,744	0.00	2	23	3,761	3,761	0.00	4	108
_75_5_2	1,827	1,827	0.00	44	2,090	2,420	2,420	0.00	10	149	4,024	4,024	0.00	141	3,239
_80_5_0	1,483	1,513	2.05	139	18,000	2,413	2,413	0.00	43	1,127	4,296	4,296	0.00	9	176
_80_5_1	1,425	1,465	2.84	201	18,000	2,622	2,622	0.00	48	703	4,248	4,248	0.00	4	383
_80_5_2	1,719	1,719	0.00	31	596	2,778	2,778	0.00	161	7,479	4,099	4,161	1.52	117	18,000
_85_5_0	1,717	1,752	2.07	157	18,000	2,686	2,686	0.00	11	122	4,219	4,243	0.57	272	18,000
_85_5_1	1,478	1,549	4.84	109	18,000	2,843	2,890	1.65	153	18,000	4,136	4,136	0.00	1	48
_85_5_2	2,041	2,387	16.95	99	18,000	2,588	2,588	0.00	209	13,906	4,264	4,328	1.50	204	18,000
_90_5_0	1,761	2,215	25.77	82	18,000	2,923	2,977	1.86	174	18,000	4,279	4,279	0.00	13	628
_90_5_1	1,530	1,530	0.00	7	134	2,732	2,732	0.00	0	24	4,167	4,167	0.00	57	2,196
_90_5_2	1,494	1,494	0.00	52	3,106	2,879	2,879	0.00	25	412	4,095	4,191	2.34	203	18,000
_95_5_0	1,592	1,592	0.00	169	17,004	2,827	2,827	0.00	8	261	4,521	4,521	0.00	4	284
_95_5_1	1,486	1,504	1.23	184	18,000	2,833	2,833	0.00	24	370	4,410	4,410	0.00	22	322

Table 10. (Continued)

Instance A-MDmTSP	I					II					III				
	LB	UB	Gap	Nodes	Time (s)	LB	UB	Gap	Nodes	Time (s)	LB	UB	Gap	Nodes	Time (s)
_95_5_2	1,452	1,528	5.23	130	18,000	2,706	2,706	0.00	125	4,394	4,682	4,682	0.00	65	8,927
_100_5_0	1,791	1,852	3.43	100	18,000	3,293	3,293	0.00	43	1,285	4,291	4,291	0.00	2	127
_100_5_1	1,653	1,680	1.66	140	18,000	2,958	3,027	2.33	104	18,000	4,542	4,542	0.00	0	83
_100_5_2	1,860	1,942	4.41	109	18,000	3,124	3,251	4.08	133	18,000	4,375	4,462	1.98	132	18,000
_50_10_0	1,681	1,681	0.00	0	9	2,680	2,680	0.00	0	2	2,595	2,595	0.00	0	3
_50_10_1	1,410	1,410	0.00	0	9	2,079	2,079	0.00	0	4	2,868	2,868	0.00	1	7
_50_10_2	1,372	1,372	0.00	1	10	1,902	1,902	0.00	0	3	3,013	3,013	0.00	0	3
_60_10_0	1,354	1,354	0.00	1	17	2,421	2,421	0.00	0	4	3,140	3,140	0.00	0	7
_60_10_1	1,772	1,772	0.00	0	12	2,372	2,372	0.00	0	2	3,430	3,430	0.00	0	15
_60_10_2	1,873	1,873	0.00	29	219	2,528	2,528	0.00	0	3	3,200	3,200	0.00	23	160
_70_10_0	1,578	1,578	0.00	0	20	2,286	2,286	0.00	0	7	3,684	3,684	0.00	6	56
_70_10_1	1,538	1,538	0.00	16	70	2,479	2,479	0.00	1	14	3,332	3,332	0.00	2	41
_70_10_2	1,415	1,415	0.00	9	19	2,555	2,555	0.00	0	16	3,819	3,819	0.00	0	11
_75_10_0	1,350	1,350	0.00	6	40	2,630	2,630	0.00	118	2,909	3,545	3,545	0.00	9	91
_75_10_1	1,513	1,638	8.23	163	18,000	2,807	2,807	0.00	16	300	3,987	3,987	0.00	22	185
_75_10_2	1,773	1,900	7.18	178	18,000	2,725	2,725	0.00	449	15,969	3,610	3,610	0.00	373	6,396
_80_10_0	1,344	1,344	0.00	5	35	2,525	2,525	0.00	5	73	3,721	3,721	0.00	1	42
_80_10_1	1,499	1,499	0.00	22	333	2,379	2,379	0.00	13	62	3,976	3,976	0.00	416	7,246
_80_10_2	1,559	1,559	0.00	26	478	2,672	2,672	0.00	2	23	3,831	3,831	0.00	0	32
_85_10_0	1,563	1,589	1.69	191	18,000	2,772	2,772	0.00	10	109	4,230	4,241	0.25	365	18,000
_85_10_1	1,590	1,599	0.58	226	18,000	2,634	2,634	0.00	195	12,114	3,850	3,926	1.97	187	18,000
_85_10_2	1,650	1,781	7.92	111	18,000	2,644	2,644	0.00	230	12,465	4,268	4,268	0.00	361	7,709
_90_10_0	1,738	2,345	34.95	54	18,000	2,605	2,758	5.88	135	18,000	4,066	4,066	0.00	5	210
_90_10_1	1,457	1,457	0.00	121	4,623	2,521	2,521	0.00	22	235	4,058	4,058	0.00	35	987
_90_10_2	1,293	1,293	0.00	7	76	2,742	2,742	0.00	3	43	4,176	4,259	1.98	207	18,000
_95_10_0	1,590	1,626	2.26	126	18,000	2,832	2,832	0.00	185	11,034	4,505	4,533	0.62	179	18,000
_95_10_1	1,603	1,613	0.60	169	18,000	3,096	3,096	0.00	12	744	4,154	4,173	0.47	408	18,000
_95_10_2	1,726	1,889	9.44	100	18,000	2,610	2,666	2.13	165	18,000	4,039	4,039	0.00	6	207
_100_10_0	1,657	2,204	33.02	52	18,000	3,130	3,167	1.19	137	18,000	4,733	4,733	0.00	189	9,344
_100_10_1	1,575	2,062	30.91	42	18,000	2,794	3,037	8.71	116	18,000	4,330	4,391	1.40	161	18,000
_100_10_2	1,605	1,639	2.13	150	18,000	2,880	2,880	0.00	30	729	4,403	4,403	0.00	1	129
_80_20_0	1,352	1,352	0.00	51	258	2,075	2,075	0.00	1	26	3,744	3,744	0.00	1	57
_80_20_1	1,321	1,321	0.00	47	179	2,813	2,813	0.00	32	197	3,641	3,641	0.00	2	52
_80_20_2	1,584	1,584	0.00	393	4,762	2,635	2,635	0.00	48	373	3,891	3,891	0.00	60	909
_85_20_0	1,451	1,451	0.00	23	114	2,360	2,360	0.00	15	229	3,922	3,922	0.00	0	55
_85_20_1	1,480	1,480	0.00	13	154	2,659	2,659	0.00	82	1,356	3,688	3,688	0.00	0	29
_85_20_2	1,390	1,390	0.00	149	1,258	2,520	2,520	0.00	115	1,171	4,033	4,033	0.00	75	1,986
_90_20_0	1,460	1,460	0.00	22	134	2,439	2,439	0.00	1	30	3,991	3,991	0.00	51	520
_90_20_1	1,377	1,377	0.00	25	69	2,610	2,610	0.00	26	239	3,759	3,759	0.00	0	24
_90_20_2	1,538	1,538	0.00	380	6,228	2,678	2,678	0.00	74	1,248	3,998	3,998	0.00	0	36
_95_20_0	1,554	2,449	57.54	51	18,000	2,625	2,680	2.09	243	18,000	4,151	4,263	2.69	194	18,000
_95_20_1	1,603	1,603	0.00	31	157	2,660	2,660	0.00	9	109	4,202	4,202	0.00	126	3,633
_95_20_2	1,231	1,234	0.28	202	18,000	2,613	2,613	0.00	13	208	4,145	4,159	0.34	308	18,000
_100_20_0	1,594	1,916	20.21	63	18,000	2,794	3,103	11.08	456	18,000	4,299	4,299	0.00	55	1,528
_100_20_1	1,533	1,533	0.00	217	3,589	2,390	2,390	0.00	4	34	4,077	4,077	0.00	3	68
_100_20_2	1,266	1,287	1.64	208	18,000	2,638	2,638	0.00	80	2,982	4,321	4,321	0.00	55	1,210

Table 11. Basic Algorithm Solutions to A-MDmTSP Instances

Instance A-MDmTSP	I					II					III				
	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)
_50_5_0	1,294	1,294	0.00	0	0	1,893	1,893	0.00	0	1	2,856	2,856	0.00	0	2
_50_5_1	1,187	1,187	0.00	0	2	2,487	2,487	0.00	0	0	2,901	2,901	0.00	4	14
_50_5_2	1,356	1,356	0.00	3	17	1,938	1,938	0.00	0	1	3,286	3,286	0.00	4	7
_60_5_0	1,736	1,736	0.00	27	940	2,613	2,613	0.00	10	49	3,250	3,250	0.00	124	1,024
_60_5_1	1,636	1,636	0.00	65	1,078	2,424	2,424	0.00	3	14	3,431	3,431	0.00	1	5
_60_5_2	1,625	1,726	6.20	272	18,000	2,507	2,507	0.00	53	1,484	3,862	3,883	0.53	444	18,000

Table 11. (Continued)

Instance A-MDmTSP	I					II					III				
	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)
_70_5_0	1,897	—	—	91	18,000	2,568	2,568	0.00	138	11,223	3,601	3,601	0.00	31	360
_70_5_1	1,428	1,428	0.00	39	544	2,661	2,661	0.00	34	412	3,480	3,561	2.34	206	18,000
_70_5_2	1,651	1,651	0.00	119	5,559	2,757	2,757	0.00	1	8	3,559	3,559	0.00	0	1
_75_5_0	1,630	1,630	0.00	7	55	2,590	2,590	0.00	13	286	3,851	3,851	0.00	22	291
_75_5_1	1,802	1,802	0.00	3	71	2,744	2,744	0.00	11	117	3,761	3,761	0.00	13	559
_75_5_2	1,812	1,854	2.34	115	18,000	2,420	2,420	0.00	50	1,279	4,024	4,024	0.00	240	5,919
_80_5_0	1,443	2,318	60.63	73	18,000	2,413	2,413	0.00	99	3,575	4,296	4,296	0.00	16	641
_80_5_1	1,378	4,100	197.53	96	18,000	2,614	2,623	0.36	163	18,000	4,248	4,248	0.00	33	1,466
_80_5_2	1,688	—	—	81	18,000	2,750	2,815	2.36	170	18,000	3,998	5,396	34.97	64	18,000
_85_5_0	1,665	4,112	146.97	64	18,000	2,686	2,686	0.00	33	1,582	4,209	4,252	1.03	260	18,000
_85_5_1	1,409	—	—	56	18,000	2,792	6,613	136.83	52	18,000	4,136	4,136	0.00	2	70
_85_5_2	1,994	—	—	57	18,000	2,537	5,508	117.11	74	18,000	4,207	4,546	8.07	96	18,000
_90_5_0	1,715	—	—	50	18,000	2,873	21,436	646.23	58	18,000	4,279	4,279	0.00	30	1,749
_90_5_1	1,530	1,530	0.00	44	2,469	2,732	2,732	0.00	1	21	4,126	4,189	1.52	137	18,000
_90_5_2	1,468	6,521	344.30	58	18,000	2,879	2,879	0.00	109	9,996	4,063	4,369	7.54	89	18,000
_95_5_0	1,546	1,944	25.73	59	18,000	2,817	2,951	4.77	111	18,000	4,521	4,521	0.00	17	1,750
_95_5_1	1,444	6,638	359.65	70	18,000	2,833	2,833	0.00	65	4,404	4,410	4,410	0.00	21	439
_95_5_2	1,411	—	—	49	18,000	2,674	2,787	4.24	81	18,000	4,647	4,831	3.97	101	18,000
_100_5_0	1,755	14,142	705.90	47	18,000	3,266	3,594	10.04	97	18,000	4,291	4,291	0.00	13	505
_100_5_1	1,623	6,316	289.16	56	18,000	2,924	5,091	74.09	42	18,000	4,542	4,542	0.00	2	144
_100_5_2	1,817	—	—	46	18,000	3,087	3,967	28.52	55	18,000	4,307	4,841	12.40	76	18,000
_50_10_0	1,681	1,681	0.00	1	5	2,680	2,680	0.00	0	1	2,595	2,595	0.00	0	0
_50_10_1	1,410	1,410	0.00	1	3	2,079	2,079	0.00	2	8	2,868	2,868	0.00	1	2
_50_10_2	1,372	1,372	0.00	3	5	1,902	1,902	0.00	0	3	3,013	3,013	0.00	0	2
_60_10_0	1,354	1,354	0.00	4	12	2,421	2,421	0.00	2	5	3,140	3,140	0.00	1	7
_60_10_1	1,772	1,772	0.00	6	33	2,372	2,372	0.00	0	3	3,432	3,432	0.00	2	20
_60_10_2	1,873	1,873	0.00	80	1,727	2,528	2,528	0.00	1	5	3,200	3,200	0.00	36	242
_70_10_0	1,578	1,578	0.00	3	24	2,286	2,286	0.00	5	15	3,684	3,684	0.00	36	545
_70_10_1	1,538	1,538	0.00	32	253	2,479	2,479	0.00	4	13	3,332	3,332	0.00	7	80
_70_10_2	1,415	1,415	0.00	53	173	2,555	2,555	0.00	11	114	3,819	3,819	0.00	0	5
_75_10_0	1,350	1,350	0.00	40	1,137	2,630	2,630	0.00	202	16,654	3,545	3,545	0.00	13	162
_75_10_1	1,437	—	—	79	18,000	2,807	2,807	0.00	176	9,026	3,987	3,987	0.00	126	1,411
_75_10_2	1,717	—	—	73	18,000	2,677	2,843	6.21	193	18,000	3,610	3,610	0.00	359	13,348
_80_10_0	1,344	1,344	0.00	6	47	2,525	2,525	0.00	27	293	3,721	3,721	0.00	9	74
_80_10_1	1,499	1,499	0.00	94	4,187	2,379	2,379	0.00	28	175	3,954	3,994	1.00	460	18,000
_80_10_2	1,529	2,828	85.00	97	18,000	2,672	2,672	0.00	7	16	3,831	3,831	0.00	3	30
_85_10_0	1,519	3,914	157.71	94	18,000	2,772	2,772	0.00	24	119	4,173	4,280	2.57	240	18,000
_85_10_1	1,549	—	—	73	18,000	2,579	3,216	24.70	92	18,000	3,807	3,933	3.30	114	18,000
_85_10_2	1,609	—	—	67	18,000	2,630	2,647	0.67	223	18,000	4,228	4,294	1.55	194	18,000
_90_10_0	1,698	—	—	56	18,000	2,564	1,5309	497.19	60	18,000	4,066	4,066	0.00	26	725
_90_10_1	1,432	1,489	3.97	160	18,000	2,521	2,521	0.00	59	1,079	4,058	4,058	0.00	90	4,591
_90_10_2	1,293	1,293	0.00	140	10,600	2,742	2,742	0.00	23	475	4,090	4,856	18.72	79	18,000
_95_10_0	1,561	—	—	51	18,000	2,790	3,209	15.01	103	18,000	4,429	4,733	6.86	71	18,000
_95_10_1	1,570	—	—	75	18,000	3,068	9,418	206.94	64	18,000	4,125	4,197	1.75	292	18,000
_95_10_2	1,695	—	—	55	18,000	2,572	—	—	64	18,000	4,039	4,039	0.00	49	1,868
_100_10_0	1,626	—	—	42	18,000	3,084	9,222	199.02	58	18,000	4,678	4,810	2.81	129	18,000
_100_10_1	1,540	—	—	46	18,000	2,751	—	—	53	18,000	4,225	4,594	8.73	93	18,000
_100_10_2	1,578	12,140	669.57	59	18,000	2,870	2,891	0.72	146	18,000	4,403	4,403	0.00	9	116
_80_20_0	1,352	1,352	0.00	162	1,060	2,075	2,075	0.00	11	71	3,744	3,744	0.00	2	17
_80_20_1	1,321	1,321	0.00	180	2,651	2,813	2,813	0.00	53	1,138	3,641	3,641	0.00	8	63
_80_20_2	1,548	—	—	157	18,000	2,635	2,635	0.00	400	7,002	3,891	3,891	0.00	124	812
_85_20_0	1,451	1,451	0.00	207	3,936	2,360	2,360	0.00	242	2,618	3,922	3,922	0.00	3	25
_85_20_1	1,480	1,480	0.00	119	7,238	2,659	2,659	0.00	226	5,543	3,688	3,688	0.00	0	5
_85_20_2	1,378	1,393	1.09	499	18,000	2,520	2,520	0.00	254	4,840	4,033	4,033	0.00	216	10,659
_90_20_0	1,460	1,460	0.00	156	3,435	2,439	2,439	0.00	50	821	3,991	3,991	0.00	64	515
_90_20_1	1,377	1,377	0.00	103	360	2,610	2,610	0.00	49	530	3,759	3,759	0.00	0	5
_90_20_2	1,495	—	—	132	18,000	2,672	2,687	0.58	455	18,000	3,998	3,998	0.00	0	13
_95_20_0	1,521	—	—	54	18,000	2,578	—	—	89	18,000	4,077	4,749	16.49	99	18,000
_95_20_1	1,603	1,603	0.00	175	1,954	2,660	2,660	0.00	25	318	4,201	4,202	0.03	280	18,000
_95_20_2	1,194	—	—	85	18,000	2,613	2,613	0.00	80	1,792	4,092	4,274	4.46	132	18,000
_100_20_0	1,561	—	—	47	18,000	2,737	—	—	132	18,000	4,299	4,299	0.00	55	1,445
_100_20_1	1,507	3,468	130.13	130	18,000	2,390	2,390	0.00	96	5,506	4,077	4,077	0.00	3	36
_100_20_2	1,241	—	—	94	18,000	2,630	2,649	0.72	322	18,000	4,321	4,321	0.00	121	3,019

Table 12. Branch-and-Cut Framework Solutions to A-MDCVRP Instances

Instance A-MDCVRP	I					II					III				
	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)
_40_5_0	1,548	1,548	0.00	1	7	2,248	2,248	0.00	0	3	2,838	2,838	0.00	1	6
_40_5_1	1,700	1,700	0.00	1	17	2,235	2,235	0.00	1	12	2,890	2,890	0.00	0	5
_40_5_2	1,549	1,549	0.00	0	4	2,219	2,219	0.00	5	37	3,035	3,035	0.00	2	10
_45_5_0	1,707	1,707	0.00	0	10	2,164	2,164	0.00	2	13	3,264	3,264	0.00	0	6
_45_5_1	1,760	1,760	0.00	2	9	2,488	2,488	0.00	0	7	3,080	3,080	0.00	74	537
_45_5_2	1,427	1,427	0.00	19	85	2,340	2,340	0.00	263	5,216	3,456	3,456	0.00	6	109
_50_5_0	1,568	1,568	0.00	73	332	2,235	2,290	2.48	492	18,000	3,692	3,692	0.00	674	10,255
_50_5_1	1,370	1,370	0.00	64	439	2,802	2,802	0.00	219	3,186	3,285	3,305	0.62	727	18,000
_50_5_2	1,381	1,381	0.00	4	26	2,036	2,036	0.00	7	25	3,378	3,378	0.00	2	23
_55_5_0	1,627	1,627	0.00	50	506	2,387	2,387	0.00	107	1,572	3,313	3,413	3.03	391	18,000
_55_5_1	2,179	2,314	6.21	218	18,000	2,970	3,184	7.19	257	18,000	4,369	4,369	0.00	227	7,013
_55_5_2	1,735	1,735	0.00	49	240	2,880	2,880	0.00	63	551	3,705	3,736	0.83	334	18,000
_60_5_0	1,936	2,031	4.89	255	18,000	2,913	3,124	7.24	226	18,000	3,488	3,488	0.00	118	1,492
_60_5_1	1,656	1,656	0.00	59	737	2,516	2,516	0.00	249	5,565	3,656	3,656	0.00	0	25
_60_5_2	1,561	1,561	0.00	8	55	2,536	2,590	2.13	288	18,000	3,767	3,767	0.00	0	53
_65_5_0	1,483	1,483	0.00	6	63	2,483	2,483	0.00	48	436	3,936	3,936	0.00	60	684
_65_5_1	1,831	1,877	2.51	184	18,000	2,929	3,097	5.75	225	18,000	4,074	4,219	3.55	207	18,000
_65_5_2	1,628	1,628	0.00	134	2,905	2,636	2,636	0.00	336	15,570	3,826	3,826	0.00	10	171
_70_5_0	1,953	2,011	2.95	236	18,000	2,724	2,934	7.73	190	18,000	3,928	3,928	0.00	80	2,209
_70_5_1	1,492	1,492	0.00	49	930	2,877	2,877	0.00	363	11,674	3,853	3,853	0.00	0	51
_70_5_2	1,920	2,078	8.23	149	18,000	3,129	3,480	11.22	138	18,000	4,200	4,200	0.00	0	46
_75_5_0	1,672	1,672	0.00	18	149	2,753	2,753	0.00	150	7,744	4,043	4,134	2.25	197	18,000
_75_5_1	2,017	2,117	4.94	208	18,000	3,148	3,429	8.94	137	18,000	4,340	4,447	2.47	114	18,000
_75_5_2	1,857	1,857	0.00	161	14,348	2,499	2,567	2.72	200	18,000	4,472	4,528	1.25	163	18,000
_40_10_0	1,516	1,516	0.00	0	4	2,595	2,595	0.00	5	16	2,558	2,558	0.00	0	2
_40_10_1	1,420	1,420	0.00	0	5	1,776	1,776	0.00	0	3	2,736	2,736	0.00	2	11
_40_10_2	1,560	1,560	0.00	21	82	2,180	2,180	0.00	8	31	2,418	2,418	0.00	54	507
_45_10_0	1,376	1,376	0.00	2	10	1,817	1,817	0.00	3	13	2,779	2,779	0.00	321	2,098
_45_10_1	1,640	1,640	0.00	15	81	2,376	2,467	3.81	445	18,000	3,010	3,010	0.00	6	58
_45_10_2	1,293	1,293	0.00	0	6	2,230	2,230	0.00	2	11	2,920	2,920	0.00	0	3
_50_10_0	1,682	1,682	0.00	0	7	2,699	2,699	0.00	5	84	2,623	2,623	0.00	0	5
_50_10_1	1,498	1,498	0.00	58	413	2,249	2,396	6.53	258	18,000	3,142	3,142	0.00	0	12
_50_10_2	1,372	1,372	0.00	2	21	1,948	1,948	0.00	8	43	2,949	2,949	0.00	0	4
_55_10_0	1,465	1,465	0.00	101	1,405	2,328	2,344	0.68	534	18,000	3,789	3,814	0.67	348	18,000
_55_10_1	1,556	1,556	0.00	24	147	2,556	2,556	0.00	96	821	3,427	3,427	0.00	0	33
_55_10_2	1,323	1,323	0.00	12	24	2,488	2,488	0.00	1	11	2,882	2,882	0.00	0	20
_60_10_0	1,481	1,495	0.95	401	18,000	2,528	2,671	5.65	269	18,000	3,395	3,395	0.00	0	25
_60_10_1	1,817	1,817	0.00	63	1,150	2,338	2,338	0.00	12	62	3,737	3,737	0.00	11	189
_60_10_2	2,020	2,348	16.24	146	18,000	2,788	2,953	5.93	200	18,000	3,482	3,482	0.00	9	158
_65_10_0	1,520	1,588	4.50	195	18,000	2,147	2,251	4.83	239	18,000	3,699	3,730	0.85	764	18,000
_65_10_1	1,314	1,314	0.00	128	2,095	2,115	2,148	1.58	306	18,000	3,643	3,657	0.39	511	18,000
_65_10_2	1,721	1,958	13.80	177	18,000	2,644	2,761	4.42	229	18,000	3,474	3,474	0.00	564	12,694
_70_10_0	1,644	1,644	0.00	90	3,664	2,322	2,344	0.95	320	18,000	3,922	3,979	1.45	570	18,000
_70_10_1	1,668	2,007	20.35	112	18,000	2,798	3,298	17.89	108	18,000	3,980	4,026	1.16	288	18,000
_70_10_2	1,459	1,487	1.90	404	18,000	2,611	2,669	2.22	199	18,000	4,157	4,157	0.00	7	238
_75_10_0	1,350	1,350	0.00	6	52	2,670	2,840	6.37	189	18,000	3,578	3,578	0.00	3	105
_75_10_1	1,567	1,874	19.60	99	18,000	2,902	3,344	15.21	111	18,000	4,531	4,531	0.00	15	476
_75_10_2	1,765	1,810	2.58	189	18,000	2,718	2,832	4.19	195	18,000	3,690	3,762	1.96	225	18,000
_80_10_0	1,366	1,366	0.00	38	881	2,538	2,614	3.01	226	18,000	3,928	3,928	0.00	18	605
_80_10_1	1,630	1,805	10.71	102	18,000	2,506	2,675	6.72	117	18,000	4,370	4,403	0.76	340	18,000
_80_10_2	1,582	1,612	1.92	147	18,000	2,669	2,669	0.00	34	663	4,051	4,051	0.00	177	5,021
_85_10_0	1,665	2,037	22.33	81	18,000	2,797	3,032	8.38	79	18,000	5,193	5,328	2.60	77	18,000
_85_10_1	1,596	1,653	3.56	158	18,000	2,774	3,286	18.44	91	18,000	4,064	4,064	0.00	18	1,506
_85_10_2	1,641	1,675	2.04	154	18,000	2,634	2,697	2.40	130	18,000	4,487	4,501	0.31	329	18,000
_75_20_0	1,498	1,642	9.61	113	18,000	2,606	2,684	2.99	196	18,000	3,989	4,024	0.88	182	18,000
_75_20_1	1,670	1,683	0.80	255	18,000	2,648	2,689	1.54	273	18,000	3,462	3,462	0.00	1	54
_75_20_2	1,433	1,628	13.64	142	18,000	2,496	2,797	12.06	133	18,000	4,077	4,077	0.00	36	849
_80_20_0	1,358	1,358	0.00	419	6,704	2,067	2,089	1.06	171	18,000	3,875	3,875	0.00	0	36
_80_20_1	1,363	1,899	39.34	97	18,000	2,893	3,263	12.80	106	18,000	3,833	3,833	0.00	102	1,067
_80_20_2	1,584	1,647	3.97	305	18,000	2,563	2,659	3.73	165	18,000	3,983	3,983	0.00	334	7,184

Table 12. (Continued)

Instance A-MDCVRP	I					II					III				
	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)
_85_20_0	1,479	1,929	30.39	82	18,000	2,390	2,861	19.73	70	18,000	4,353	4,438	1.94	148	18,000
_85_20_1	1,529	1,801	17.76	105	18,000	2,616	2,984	14.08	78	18,000	4,054	4,086	0.78	266	18,000
_85_20_2	1,375	1,375	0.00	181	1,468	2,435	2,522	3.59	146	18,000	3,968	3,968	0.00	169	7,860

Table 13. Basic Algorithm Solutions to A-MDCVRP Instances

Instance A-MDCVRP	I					II					III				
	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)
_40_5_0	1,548	1,548	0.00	1	2	2,248	2,248	0.00	0	1	2,838	2,838	0.00	2	1
_40_5_1	1,700	1,700	0.00	15	77	2,235	2,235	0.00	2	8	2,890	2,890	0.00	1	3
_40_5_2	1,549	1,549	0.00	1	2	2,219	2,219	0.00	9	20	3,035	3,035	0.00	4	4
_45_5_0	1,707	1,707	0.00	2	7	2,164	2,164	0.00	9	28	3,264	3,264	0.00	0	2
_45_5_1	1,760	1,760	0.00	9	40	2,488	2,488	0.00	2	5	3,080	3,080	0.00	77	281
_45_5_2	1,427	1,427	0.00	93	1,337	2,294	2,370	3.30	353	18,000	3,456	3,456	0.00	23	231
_50_5_0	1,568	1,568	0.00	213	10,857	2,114	2,903	37.33	174	18,000	3,692	3,692	0.00	731	12,087
_50_5_1	1,370	1,370	0.00	170	6,060	2,785	2,802	0.61	439	18,000	3,228	3,370	4.39	325	18,000
_50_5_2	1,381	1,381	0.00	6	59	2,036	2,036	0.00	13	103	3,378	3,378	0.00	2	25
_55_5_0	1,627	1,627	0.00	107	3,085	2,339	2,538	8.53	208	18,000	3,283	3,426	4.35	331	18,000
_55_5_1	2,088	6,492	210.92	130	18,000	2,848	—	—	107	18,000	4,369	4,369	0.00	511	15,747
_55_5_2	1,735	1,735	0.00	93	2,880	2,880	2,880	0.00	229	10,559	3,678	3,779	2.74	235	18,000
_60_5_0	1,851	7,559	308.37	116	18,000	2,788	6,575	135.85	93	18,000	3,488	3,488	0.00	88	950
_60_5_1	1,656	1,656	0.00	151	7,347	2,462	2,797	13.63	130	18,000	3,656	3,656	0.00	1	19
_60_5_2	1,561	1,561	0.00	68	1,380	2,478	2,824	13.95	164	18,000	3,767	3,767	0.00	9	170
_65_5_0	1,483	1,483	0.00	46	1,753	2,483	2,483	0.00	98	3,536	3,936	3,936	0.00	68	2,416
_65_5_1	1,747	9,186	425.86	97	18,000	2,841	6,130	115.80	84	18,000	4,037	4,260	5.53	285	18,000
_65_5_2	1,606	1,655	3.08	211	18,000	2,571	3,168	23.22	102	18,000	3,826	3,826	0.00	35	579
_70_5_0	1,895	2,819	48.75	114	18,000	2,647	7,575	186.17	89	18,000	3,928	3,928	0.00	81	4,170
_70_5_1	1,461	2,192	50.04	118	18,000	2,794	4,202	50.41	106	18,000	3,853	3,853	0.00	1	29
_70_5_2	1,854	—	—	91	18,000	3,058	—	—	74	18,000	4,200	4,200	0.00	2	44
_75_5_0	1,672	1,672	0.00	57	2,746	2,692	4,906	82.22	82	18,000	3,981	4,393	10.34	104	18,000
_75_5_1	1,962	5,967	204.11	92	18,000	3,054	—	—	73	18,000	4,254	—	—	86	18,000
_75_5_2	1,815	4,383	141.55	87	18,000	2,448	16,598	577.95	73	18,000	4,405	4,549	3.26	156	18,000
_40_10_0	1,516	1,516	0.00	1	1	2,595	2,595	0.00	14	32	2,558	2,558	0.00	0	0
_40_10_1	1,420	1,420	0.00	0	0	1,776	1,776	0.00	1	2	2,736	2,736	0.00	1	1
_40_10_2	1,560	1,560	0.00	49	262	2,180	2,180	0.00	12	29	2,418	2,418	0.00	62	354
_45_10_0	1,376	1,376	0.00	4	11	1,817	1,817	0.00	3	17	2,779	2,779	0.00	314	2,270
_45_10_1	1,640	1,640	0.00	167	4,924	2,272	2,956	30.13	223	18,000	3,010	3,010	0.00	13	41
_45_10_2	1,293	1,293	0.00	0	2	2,230	2,230	0.00	7	15	2,920	2,920	0.00	0	1
_50_10_0	1,682	1,682	0.00	3	18	2,699	2,699	0.00	12	79	2,623	2,623	0.00	0	1
_50_10_1	1,498	1,498	0.00	287	6,292	2,164	2,707	25.12	195	18,000	3,142	3,142	0.00	1	6
_50_10_2	1,372	1,372	0.00	6	12	1,948	1,948	0.00	33	253	2,949	2,949	0.00	0	2
_55_10_0	1,465	1,465	0.00	249	12,748	2,248	2,614	16.30	192	18,000	3,735	3,858	3.28	321	18,000
_55_10_1	1,556	1,556	0.00	78	1,308	2,556	2,556	0.00	282	10,435	3,427	3,427	0.00	2	28
_55_10_2	1,323	1,323	0.00	14	51	2,488	2,488	0.00	9	21	2,882	2,882	0.00	3	28
_60_10_0	1,430	1,500	4.92	212	18,000	2,451	3,867	57.78	122	18,000	3,395	3,395	0.00	4	12
_60_10_1	1,790	2,684	49.98	147	18,000	2,338	2,338	0.00	48	882	3,737	3,737	0.00	20	142
_60_10_2	1,930	6,833	254.00	124	18,000	2,687	—	—	104	18,000	3,482	3,482	0.00	23	158
_65_10_0	1,462	5,511	276.95	116	18,000	2,085	7,677	268.25	92	18,000	3,679	3,744	1.77	798	18,000
_65_10_1	1,314	1,314	0.00	248	15,457	2,063	2,234	8.31	177	18,000	3,631	3,658	0.76	308	18,000
_65_10_2	1,660	—	—	105	18,000	2,565	—	—	95	18,000	3,474	3,474	0.00	321	8,304
_70_10_0	1,644	1,644	0.00	174	16,460	2,260	2,483	9.84	107	18,000	3,881	4,124	6.27	334	18,000
_70_10_1	1,600	—	—	85	18,000	2,708	—	—	75	18,000	3,935	4,078	3.64	201	18,000
_70_10_2	1,407	5,546	294.25	165	18,000	2,544	19,978	685.16	78	18,000	4,157	4,157	0.00	19	293
_75_10_0	1,350	1,350	0.00	32	974	2,611	6,210	137.84	94	18,000	3,578	3,578	0.00	4	71
_75_10_1	1,502	—	—	81	18,000	2,830	—	—	81	18,000	4,531	4,531	0.00	61	908
_75_10_2	1,715	5,683	231.30	94	18,000	2,659	—	—	81	18,000	3,626	3,878	6.96	183	18,000
_80_10_0	1,366	1,366	0.00	80	4,228	2,488	—	—	78	18,000	3,928	3,928	0.00	128	4,584

Table 13. (Continued)

Instance A-MDCVRP	I					II					III				
	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)
_80_10_1	1,580	—	—	65	18,000	2,442	7,002	186.68	68	18,000	4,314	4,487	4.01	154	18,000
_80_10_2	1,539	—	—	79	18,000	2,638	3,132	18.73	94	18,000	4,051	4,051	0.00	196	5,178
_85_10_0	1,612	—	—	58	18,000	2,735	—	—	57	18,000	5,089	6,823	34.08	82	18,000
_85_10_1	1,561	—	—	79	18,000	2,712	—	—	58	18,000	4,044	4,064	0.49	83	18,000
_85_10_2	1,605	3,439	114.22	99	18,000	2,597	2,938	13.13	92	18,000	4,449	4,516	1.51	231	18,000
_75_20_0	1,460	2,954	102.32	93	18,000	2,552	7,553	195.95	93	18,000	3,947	4,077	3.28	251	18,000
_75_20_1	1,627	2,776	70.64	126	18,000	2,597	—	—	92	18,000	3,462	3,462	0.00	6	63
_75_20_2	1,388	—	—	101	18,000	2,440	—	—	70	18,000	4,077	4,077	0.00	142	5,206
_80_20_0	1,334	1,381	3.53	331	18,000	2,024	—	—	80	18,000	3,875	3,875	0.00	2	30
_80_20_1	1,318	—	—	99	18,000	2,841	—	—	72	18,000	3,833	3,833	0.00	205	4,162
_80_20_2	1,549	6,904	345.76	114	18,000	2,522	—	—	98	18,000	3,955	3,986	0.78	473	18,000
_85_20_0	1,434	—	—	62	18,000	2,324	—	—	43	18,000	4,286	4,494	4.84	139	18,000
_85_20_1	1,491	—	—	70	18,000	2,572	—	—	59	18,000	4,011	4,073	1.54	242	18,000
_85_20_2	1,363	1,404	2.99	470	18,000	2,386	—	—	71	18,000	3,937	3,996	1.49	192	18,000

Table 14. Branch-and-Cut Framework Solutions to A-CLRP Instances

Instance A-CLRP	I					II					III				
	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)
_40_5_0	2,281	2,281	0.00	0	6	4,222	4,222	0.00	0	6	4,297	4,297	0.00	0	24
_40_5_1	3,048	3,048	0.00	65	1,113	3,836	3,836	0.00	7	62	4,426	4,426	0.00	0	9
_40_5_2	2,831	2,831	0.00	0	10	2,962	2,962	0.00	0	3	4,110	4,110	0.00	0	7
_45_5_0	2,769	2,769	0.00	0	9	3,932	3,932	0.00	13	58	4,273	4,273	0.00	1	19
_45_5_1	2,613	2,613	0.00	0	1	4,757	4,757	0.00	0	10	4,533	4,533	0.00	1	25
_45_5_2	2,581	2,581	0.00	1	20	4,622	4,622	0.00	83	1,371	5,052	5,052	0.00	49	1,581
_50_5_0	3,331	3,331	0.00	5	67	4,654	4,789	2.91	441	18,000	6,945	7,030	1.23	201	18,000
_50_5_1	2,524	2,524	0.00	3	31	4,718	4,718	0.00	5	101	4,623	4,623	0.00	123	2,516
_50_5_2	3,249	3,249	0.00	1	21	3,688	3,688	0.00	2	30	4,641	4,641	0.00	0	15
_55_5_0	2,693	2,693	0.00	1	28	4,369	4,369	0.00	8	83	5,231	5,231	0.00	4	119
_55_5_1	3,855	3,855	0.00	165	6,592	5,408	5,408	0.00	146	3,768	6,934	7,020	1.23	155	18,000
_55_5_2	3,091	3,091	0.00	0	8	4,436	4,436	0.00	17	215	5,971	6,037	1.10	170	18,000
_60_5_0	3,770	3,902	3.49	199	18,000	5,063	5,109	0.91	193	18,000	6,409	6,409	0.00	26	879
_60_5_1	3,193	3,193	0.00	242	15,647	4,500	4,532	0.71	267	18,000	6,718	6,772	0.81	484	18,000
_60_5_2	3,467	3,467	0.00	0	20	3,625	3,625	0.00	7	217	5,663	5,663	0.00	2	96
_65_5_0	2,850	2,850	0.00	18	226	3,564	3,564	0.00	2	101	6,208	6,412	3.29	164	18,000
_65_5_1	3,061	3,061	0.00	98	6,025	4,974	4,985	0.23	300	18,000	6,262	6,262	0.00	95	3,405
_65_5_2	3,670	3,670	0.00	43	1,025	4,867	4,867	0.00	7	332	5,275	5,275	0.00	297	6,070
_70_5_0	3,487	3,487	0.00	130	8,191	5,114	5,169	1.07	167	18,000	6,212	6,308	1.55	220	18,000
_70_5_1	2,598	2,598	0.00	9	176	5,132	5,132	0.00	20	517	5,066	5,066	0.00	1	62
_70_5_2	3,728	3,782	1.46	126	18,000	6,487	6,809	4.96	96	18,000	6,383	6,788	6.34	85	18,000
_75_5_0	2,534	2,534	0.00	0	31	4,754	4,754	0.00	36	1,524	5,892	6,421	8.99	202	18,000
_75_5_1	3,696	3,778	2.22	141	18,000	5,400	5,473	1.34	209	18,000	6,290	6,733	7.04	80	18,000
_75_5_2	2,750	2,750	0.00	1	45	3,778	3,778	0.00	2	190	6,517	6,604	1.33	116	18,000
_40_10_0	2,788	2,788	0.00	0	2	4,064	4,064	0.00	0	3	4,144	4,144	0.00	15	45
_40_10_1	2,256	2,256	0.00	0	3	2,722	2,722	0.00	0	10	3,963	3,963	0.00	4	32
_40_10_2	2,854	2,854	0.00	2	18	3,617	3,617	0.00	7	61	3,981	3,981	0.00	114	1,822
_45_10_0	2,451	2,451	0.00	0	5	3,135	3,135	0.00	0	11	4,724	4,724	0.00	124	2,123
_45_10_1	3,464	3,464	0.00	9	158	4,856	4,856	0.00	70	572	5,327	5,551	4.20	511	18,000
_45_10_2	2,643	2,643	0.00	4	36	3,761	3,761	0.00	1	11	4,943	4,943	0.00	227	2,423
_50_10_0	2,548	2,548	0.00	0	3	3,945	3,945	0.00	0	5	4,256	4,256	0.00	73	1,892
_50_10_1	2,729	2,729	0.00	0	11	4,623	4,623	0.00	54	630	5,596	5,596	0.00	37	1,458
_50_10_2	2,520	2,520	0.00	212	6,231	3,255	3,255	0.00	4	65	4,557	4,610	1.16	329	18,000
_55_10_0	3,679	3,679	0.00	310	11,890	4,136	4,280	3.47	301	18,000	6,011	6,168	2.60	257	18,000
_55_10_1	2,731	2,731	0.00	3	47	5,219	5,219	0.00	85	2,233	5,376	5,376	0.00	57	692
_55_10_2	2,298	2,298	0.00	5	53	3,537	3,537	0.00	7	93	4,538	4,538	0.00	158	5,011
_60_10_0	3,002	3,002	0.00	6	179	4,659	4,678	0.42	333	18,000	5,394	5,676	5.24	294	18,000
_60_10_1	3,046	3,046	0.00	219	13,404	3,671	3,671	0.00	15	482	5,096	5,096	0.00	26	1,691

Table 14. (Continued)

Instance A-CLRP	I					II					III				
	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)
_60_10_2	3,955	4,011	1.42	142	18,000	5,257	5,753	9.43	76	18,000	5,711	5,836	2.18	320	18,000
_65_10_0	2,762	2,785	0.84	244	18,000	4,089	4,089	0.00	3	180	5,868	6,923	17.98	193	18,000
_65_10_1	2,811	2,811	0.00	2	64	3,163	3,163	0.00	113	3,540	5,430	5,430	0.00	20	1,110
_65_10_2	2,934	3,039	3.57	156	18,000	5,318	5,356	0.72	331	18,000	5,924	6,593	11.28	258	18,000
_70_10_0	3,107	3,107	0.00	94	6,898	4,157	4,157	0.00	26	1,465	5,882	6,041	2.70	181	18,000
_70_10_1	3,778	4,087	8.19	89	18,000	5,597	6,840	22.21	27	18,000	6,026	6,646	10.28	200	18,000
_70_10_2	2,836	2,836	0.00	43	1,055	4,229	4,280	1.19	224	18,000	6,510	6,582	1.10	243	18,000
_75_10_0	2,970	2,970	0.00	1	55	3,825	3,825	0.00	5	370	5,061	5,430	7.28	146	18,000
_75_10_1	3,503	4,021	14.79	69	18,000	5,080	5,402	6.33	111	18,000	7,048	7,856	11.46	105	18,000
_75_10_2	2,818	2,818	0.00	14	503	3,916	3,934	0.47	249	18,000	5,181	5,709	10.20	109	18,000
_80_10_0	2,410	2,410	0.00	1	80	4,447	4,447	0.00	12	603	5,930	6,380	7.58	174	18,000
_80_10_1	3,564	3,643	2.20	88	18,000	4,010	4,119	2.72	90	18,000	6,462	7,276	12.59	65	18,000
_80_10_2	2,898	2,981	2.85	142	18,000	4,294	4,401	2.50	100	18,000	6,564	7,588	15.60	82	18,000
_85_10_0	3,410	3,547	4.03	85	18,000	6,206	6,713	8.17	41	18,000	6,743	7,617	12.96	27	18,000
_85_10_1	3,289	3,359	2.12	108	18,000	4,842	4,932	1.86	76	18,000	5,899	6,313	7.02	157	18,000
_85_10_2	3,092	3,196	3.37	75	18,000	4,003	4,186	4.57	121	18,000	6,562	6,885	4.93	176	18,000
_75_20_0	2,980	3,007	0.91	122	18,000	4,205	4,621	9.91	38	18,000	5,874	6,258	6.53	177	18,000
_75_20_1	2,570	2,570	0.00	0	30	4,117	4,117	0.00	209	14,554	5,170	5,527	6.90	216	18,000
_75_20_2	3,770	3,815	1.19	103	18,000	4,870	5,394	10.75	52	18,000	6,218	7,537	21.22	62	18,000
_80_20_0	2,329	2,334	0.21	231	18,000	3,732	3,891	4.27	91	18,000	5,470	6,144	12.32	167	18,000
_80_20_1	3,049	3,291	7.95	89	18,000	5,260	6,503	23.64	12	18,000	6,035	7,040	16.66	106	18,000
_80_20_2	3,019	3,230	6.98	90	18,000	4,565	4,838	5.98	126	18,000	5,854	6,488	10.82	105	18,000
_85_20_0	3,828	4,220	10.23	28	18,000	5,035	6,115	21.45	30	18,000	6,877	7,650	11.24	117	18,000
_85_20_1	3,225	3,297	2.22	96	18,000	4,326	4,463	3.18	79	18,000	5,613	6,149	9.56	95	18,000
_85_20_2	2,580	2,580	0.00	6	496	4,048	4,048	0.00	102	9,046	5,841	6,205	6.24	129	18,000

Table 15. Basic Algorithm Solutions to A-CLRP Instances

Instance A-CLRP	I					II					III				
	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)
_40_5_0	2,281	2,281	0.00	1	8	4,222	4,222	0.00	0	3	4,297	4,297	0.00	6	25
_40_5_1	3,048	3,048	0.00	148	5,714	3,836	3,836	0.00	18	60	4,426	4,426	0.00	2	5
_40_5_2	2,831	2,831	0.00	1	4	2,963	2,963	0.00	0	1	4,110	4,110	0.00	0	2
_45_5_0	2,769	2,769	0.00	1	15	3,932	3,932	0.00	17	181	4,273	4,273	0.00	0	5
_45_5_1	2,613	2,613	0.00	0	1	4,757	4,757	0.00	0	3	4,533	4,533	0.00	9	39
_45_5_2	2,581	2,581	0.00	12	192	4,622	4,622	0.00	122	3,346	5,016	5,062	0.92	207	18,000
_50_5_0	3,331	3,331	0.00	24	588	4,375	7,955	81.82	151	18,000	6,764	7,123	5.30	262	18,000
_50_5_1	2,524	2,524	0.00	17	280	4,718	4,718	0.00	19	426	4,623	4,623	0.00	77	1,212
_50_5_2	3,249	3,249	0.00	3	70	3,688	3,688	0.00	7	147	4,641	4,641	0.00	0	17
_55_5_0	2,693	2,693	0.00	15	490	4,369	4,369	0.00	15	209	5,231	5,231	0.00	19	571
_55_5_1	3,764	4,297	14.18	116	18,000	5,264	13,565	157.68	94	18,000	6,852	7,443	8.62	148	18,000
_55_5_2	3,091	3,091	0.00	3	176	4,436	4,436	0.00	73	3,374	5,919	6,019	1.69	240	18,000
_60_5_0	3,684	4,714	27.95	122	18,000	4,844	9,199	89.92	90	18,000	6,409	6,409	0.00	182	7,595
_60_5_1	3,122	3,331	6.69	126	18,000	4,428	5,029	13.58	124	18,000	6,570	6,773	3.08	550	18,000
_60_5_2	3,467	3,467	0.00	4	171	3,625	3,625	0.00	40	1,559	5,663	5,663	0.00	31	364
_65_5_0	2,850	2,850	0.00	68	3,897	3,564	3,564	0.00	21	964	6,200	6,484	4.58	128	18,000
_65_5_1	3,008	3,273	8.80	108	18,000	4,723	—	—	75	18,000	6,262	6,262	0.00	150	6,129
_65_5_2	3,670	3,670	0.00	117	13,063	4,867	4,867	0.00	28	1,395	5,275	5,275	0.00	230	3,991
_70_5_0	3,409	4,036	18.39	93	18,000	4,618	—	—	57	18,000	6,051	6,669	10.22	103	18,000
_70_5_1	2,598	2,598	0.00	37	2,786	5,132	5,132	0.00	51	2,471	5,066	5,066	0.00	4	92
_70_5_2	3,637	—	—	71	18,000	5,994	—	—	42	18,000	5,959	7,891	32.42	100	18,000
_75_5_0	2,534	2,534	0.00	10	420	4,754	4,754	0.00	56	5,387	5,446	6,948	27.58	124	18,000
_75_5_1	3,634	4,956	36.37	77	18,000	5,347	5,554	3.87	110	18,000	5,972	8,567	43.46	72	18,000
_75_5_2	2,750	2,750	0.00	14	361	3,778	3,778	0.00	11	1,107	5,844	8,244	41.08	98	18,000
_40_10_0	2,788	2,788	0.00	0	1	4,064	4,064	0.00	0	1	4,144	4,144	0.00	12	28
_40_10_1	2,256	2,256	0.00	0	1	2,722	2,722	0.00	0	2	3,963	3,963	0.00	8	44
_40_10_2	2,854	2,854	0.00	14	178	3,617	3,617	0.00	19	405	3,981	3,981	0.00	214	2,198

Table 15. (Continued)

Instance A-CLRP	I					II					III				
	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)	LB	UB	Gap	Nod.	Time (s)
_45_10_0	2,451	2,451	0.00	0	4	3,135	3,135	0.00	4	29	4,724	4,724	0.00	303	5,897
_45_10_1	3,464	3,464	0.00	37	824	4,856	4,856	0.00	99	2,927	5,085	5,658	11.27	555	18,000
_45_10_2	2,643	2,643	0.00	24	506	3,761	3,761	0.00	2	14	4,943	4,943	0.00	398	4,207
_50_10_0	2,548	2,548	0.00	0	3	3,945	3,945	0.00	0	16	4,256	4,256	0.00	156	2,383
_50_10_1	2,729	2,729	0.00	0	3	4,623	4,623	0.00	148	6,014	5,596	5,596	0.00	220	5,821
_50_10_2	2,520	2,520	0.00	249	12,896	3,255	3,255	0.00	19	393	4,515	4,879	8.06	213	18,000
_55_10_0	3,569	4,080	14.32	132	18,000	4,032	5,368	33.15	127	18,000	5,853	8,128	38.87	181	18,000
_55_10_1	2,731	2,731	0.00	41	1,517	5,219	5,219	0.00	114	2,661	5,376	5,376	0.00	98	2,093
_55_10_2	2,298	2,298	0.00	15	212	3,537	3,537	0.00	10	100	4,538	4,538	0.00	89	3,459
_60_10_0	3,002	3,002	0.00	53	3,246	4,323	6,782	56.90	95	18,000	5,498	5,703	3.73	457	18,000
_60_10_1	2,991	3,259	8.95	173	18,000	3,671	3,671	0.00	44	2,817	5,096	5,096	0.00	93	5,666
_60_10_2	3,849	6,248	62.33	87	18,000	4,958	7,458	50.41	81	18,000	5,339	7,514	40.74	165	18,000
_65_10_0	2,708	2,849	5.22	120	18,000	4,089	4,089	0.00	8	417	5,762	7,276	26.27	140	18,000
_65_10_1	2,811	2,811	0.00	18	977	3,081	3,655	18.62	96	18,000	5,430	5,430	0.00	83	7,157
_65_10_2	2,841	4,647	63.54	77	18,000	4,894	—	—	75	18,000	5,265	7,044	33.79	164	18,000
_70_10_0	3,088	3,119	1.01	151	18,000	4,052	5,756	42.05	69	18,000	5,580	6,527	16.98	96	18,000
_70_10_1	3,667	—	—	45	18,000	5,202	—	—	46	18,000	5,660	—	—	126	18,000
_70_10_2	2,836	2,836	0.00	93	11,093	3,990	—	—	58	18,000	6,074	7,614	25.35	122	18,000
_75_10_0	2,970	2,970	0.00	3	164	3,801	4,558	19.92	100	18,000	4,906	5,523	12.57	110	18,000
_75_10_1	3,425	—	—	64	18,000	4,722	—	—	49	18,000	6,595	9,309	41.14	104	18,000
_75_10_2	2,818	2,818	0.00	77	16,242	3,839	5,611	46.15	77	18,000	4,898	5,847	19.37	103	18,000
_80_10_0	2,410	2,410	0.00	5	353	4,439	4,465	0.59	88	18,000	5,229	12,652	141.97	92	18,000
_80_10_1	3,503	4,479	27.88	51	18,000	3,854	5,269	36.70	59	18,000	5,979	—	—	78	18,000
_80_10_2	2,858	3,053	6.84	95	18,000	4,139	7,753	87.32	44	18,000	5,996	—	—	75	18,000
_85_10_0	3,342	—	—	50	18,000	5,530	—	—	33	18,000	6,383	—	—	31	18,000
_85_10_1	3,238	5,516	70.35	60	18,000	4,371	—	—	44	18,000	5,514	—	—	77	18,000
_85_10_2	3,040	3,987	31.15	44	18,000	3,950	5,518	39.70	47	18,000	5,917	8,782	48.41	86	18,000
_75_20_0	2,923	4,843	65.67	46	18,000	4,012	—	—	42	18,000	5,467	—	—	82	18,000
_75_20_1	2,570	2,570	0.00	16	848	4,098	4,126	0.67	190	18,000	5,043	5,770	14.42	208	18,000
_75_20_2	3,631	—	—	51	18,000	4,514	—	—	50	18,000	5,835	—	—	91	18,000
_80_20_0	2,273	3,881	70.77	56	18,000	3,585	—	—	42	18,000	5,333	6,760	26.76	141	18,000
_80_20_1	2,960	—	—	51	18,000	4,946	—	—	30	18,000	5,666	—	—	95	18,000
_80_20_2	2,949	6,250	111.97	33	18,000	4,309	6,661	54.57	40	18,000	5,325	—	—	67	18,000
_85_20_0	3,751	—	—	38	18,000	4,570	—	—	35	18,000	6,198	—	—	48	18,000
_85_20_1	3,134	—	—	35	18,000	4,219	8,495	101.36	49	18,000	5,217	—	—	51	18,000
_85_20_2	2,568	2,824	9.97	82	18,000	3,902	—	—	34	18,000	5,392	11,256	108.74	57	18,000

7. Conclusion

In this paper, we have studied a class of routing problems that are characterized by an asymmetric cost structure and the presence of multiple depots. Based on the derivation of new VIs with model-describing capabilities, we have developed a generic branch-and-cut framework that can solve multiple variants of the A-MDVRP efficiently.

The backbone of the branch-and-cut framework is a branch-and-cut algorithm that exploits, among others, a series of generalizations of so-called D_k inequalities that result in a set of constraints that eliminate subtours, ensure that tours start and end at the same depot, and impose limits on the maximum number of customers in a tour. Next to the branch-and-cut algorithm, the framework employs an upper-bound procedure that is based on iteratively solving a compact MIP formulation of a restricted version of the A-MDVRP.

The computational efficiency of the branch-and-cut framework is assessed on known and new benchmark sets. On existing instances of the A-MDTSP, our algorithm has proved optimality to all instances, significantly closing the gaps at the root node compared with a state-of-the-art approach. On average, our method uses only 4.3% of the run time.

When solving a large set of newly developed benchmark instances for four problem variants of the A-MDVRP, the derived VIs appear to be effective for three of the problem variants. For instance, root-node optimality gaps are closed by up to 67.2% and on average by 10.2%. To further assess the performance of the branch-and-cut framework, we developed a basic branch-and-cut algorithm based on traditional methods to model and solve A-MDVRPs. Overall, the framework significantly outperforms the basic algorithm because it solves more instances to optimality

because of improved lower and upper bounds. In addition, the new framework is considerably faster and uses less memory. The performance is exemplified by the ability to solve asymmetric multidepot TSP instances of up to 400 customers and 50 depots. (Previously, only solutions to instances of 300 customers and 60 depots were available in the literature.)

Our implementation of the branch-and-cut framework and the newly constructed benchmark instances are shared openly to stimulate future research in this field. Such future research might take our branch-and-cut framework as a starting point to solve new and practically inspired routing problems with asymmetric cost structures and multiple depots. Particular problems that might be suitable are the skill-based vehicle routing problems (e.g., Cappanera et al. 2013) and vehicle- and crew-scheduling problems (e.g., Fischetti et al. 2007).

Furthermore, because the leading algorithms for the capacitated vehicle routing problem (as an example) are of a branch-cut-and-price type, it would be worthwhile investigating the effectiveness of the VIs within such approaches. Especially for the A-MDCVRP and A-CLRP variants, this might lead to new leading branch-cut-and-price algorithms if the inequalities can significantly increase the root-node relaxations for the associated set-partitioning-like mathematical formulations.

Appendix. Solutions to Benchmark Instances

In Tables 8–15, we provide the statistics of solving the benchmark instances with the branch-and-cut framework and with the basic algorithm. Each table is dedicated to a particular problem variant, with each three columns of a table indicating the asymmetry class.

References

- Ascheuer N, Fischetti M, Grötschel M (2001) Solving the asymmetric travelling salesman problem with time windows by branch-and-cut. *Math. Programming* 90(3):475–506.
- Balas E (1989) The asymmetric assignment problem and some new facets of the traveling salesman polytope on a directed graph. *SIAM J. Discrete Math.* 2(4):425–451.
- Balas E, Fischetti M (1993) A lifting procedure for the asymmetric traveling salesman polytope and a large new class of facets. *Math. Program.* 58(1-3):325–352.
- Balas E, Fischetti M (2007) Polyhedral theory for the asymmetric traveling salesman problem. Gutin G, Punnen AP, eds. *The Traveling Salesman Problem and Its Variations* (Springer, Boston), 117–168.
- Baldacci R, Mingozzi A (2009) A unified exact method for solving different classes of vehicle routing problems. *Math. Program.* 120(2):347–380.
- 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, Martínez-Sykora A, Salazar-González JJ (2019) Balanced vehicle routing: Polyhedral analysis and branch-and-cut algorithm. *Eur. J. Oper. Res.* 273(2):452–463.
- Belenguer JM, Benavent E, Prins C, Prodhon C, Calvo RW (2011) A branch-and-cut method for the capacitated location-routing problem. *Comput. Oper. Res.* 38(6):931–941.
- Benavent E, Martínez A (2013) Multi-depot multiple TSP: A polyhedral study and computational results. *Ann. Oper. Res.* 207(1):7–25.
- Bosco A, Laganà D, Musmanno R, Vocaturo F (2013) Modeling and solving the mixed capacitated general routing problem. *Optim. Lett.* 7(7):1451–1469.
- Cappanera P, Scutellà MG (2014) Joint assignment, scheduling, and routing models to home care optimization: A pattern-based approach. *Transportation Sci.* 49(4):830–852.
- Cappanera P, Gouveia L, Scutellà MG (2013) Models and valid inequalities to asymmetric skill-based routing problems. *EURO J Transportation Logist.* 2(1):29–55.
- Contardo C, Cordeau JF, Gendron B (2014) An exact algorithm based on cut-and-column generation for the capacitated location-routing problem. *INFORMS J. Comput.* 26(1):88–102.
- Corberán A, Plana I, Sanchis JM (2008) The windy general routing polyhedron: A global view of many known arc routing polyhedra. *SIAM J. Discrete Math.* 22(2):606–628.
- Drexel M, Schneider M (2015) A survey of variants and extensions of the location-routing problem. *Eur. J. Oper. Res.* 241(2):283–308.
- Fernández E, Laporte G, Rodríguez-Pereira J (2017) A branch-and-cut algorithm for the multidepot rural postman problem. *Transportation Sci.* 52(2):353–369.
- Fischetti M (1991) Facets of the asymmetric traveling salesman polytope. *Math. Oper. Res.* 16(1):42–56.
- Fischetti M, Toth P (1997) A polyhedral approach to the asymmetric traveling salesman problem. *Management Sci.* 43(11):1520–1536.
- Fischetti M, Lodi A, Toth P (2007) Exact methods for the asymmetric traveling salesman problem. Gutin G, Punnen AP, eds. *The Traveling Salesman Problem and Its Variations* (Springer), 169–205.
- Fischetti M, Toth P, Vigo D (1994) A branch-and-bound algorithm for the capacitated vehicle routing problem on directed graphs. *Oper. Res.* 42(5):846–859.
- Fischetti M, Lodi A, Martello S, Toth P (2001) A polyhedral approach to simplified crew scheduling and vehicle scheduling problems. *Management Sci.* 47(6):833–850.
- Gouveia L, Riera-Ledesma J, Salazar-González JJ (2013) Reverse multistar inequalities and vehicle routing problems with a lower bound on the number of customers per route. *Networks* 61(4):309–321.
- Grötschel M, Padberg MW (1985) Polyhedral theory. Lawler EL, Lenstra JK, Rinnooy Kan AHG, Shmoys DB, eds. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization* (Wiley, New York), 251–305.
- Lahyani R, Coelho LC, Renaud J (2018) Alternative formulations and improved bounds for the multi-depot fleet size and mix vehicle routing problem. *OR Spectrum* 40(1):125–157.
- Laporte G, Nobert Y, Taillefer S (1988) Solving a family of multi-depot vehicle routing and location-routing problems. *Transportation Sci.* 22(3):161–172.
- Letchford AN, Salazar-González JJ (2006) Projection results for vehicle routing. *Math. Program.* 105(2-3):251–274.
- Letchford AN, Salazar-González JJ (2015) Stronger multi-commodity flow formulations of the capacitated vehicle routing problem. *Eur. J. Oper. Res.* 244(3):730–738.
- Letchford AN, Eglese RW, Lysgaard J (2002) Multistars, partial multistars and the capacitated vehicle routing problem. *Math. Programming* 94(1):21–40.
- Lysgaard J, Letchford AN, Eglese RW (2004) A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Math. Programming* 100(2):423–445.
- Mak V, Ernst AT (2007) New cutting-planes for the time-and/or precedence-constrained ATSP and directed VRP. *Math. Methods Oper. Res.* 66(1):69–98.
- Min H, Jayaraman V, Srivastava R (1998) Combined location-routing problems: A synthesis and future research directions. *Eur. J. Oper. Res.* 108(1):1–15.
- Nagy G, Salhi S (2007) Location-routing: Issues, models and methods. *Eur. J. Oper. Res.* 177(2):649–672.

- Paraskevopoulos DC, Laporte G, Repoussis PP, Tarantilis CD (2017) Resource constrained routing and scheduling: Review and research prospects. *Eur. J. Oper. Res.* 263(3):737–754.
- Prodhon C, Prins C (2014) A survey of recent research on location-routing problems. *Eur. J. Oper. Res.* 238(1):1–17.
- Schneider M, Drexel M (2017) A survey of the standard location-routing problem. *Ann. Oper. Res.* 259(1-2):389–414.

Michiel uit het Broek is a postdoctoral researcher in operations research at the University of Groningen, Netherlands. His research interests range from optimal dynamic decision making on production and maintenance to heuristics and approximate methods focused on routing and scheduling problems.

Albert Schrottenboer is a postdoctoral researcher in operations research at the University of Groningen, Netherlands. His research interests include the design of exact and

heuristic methods for logistics problems arising in e-commerce and the energy transition

Bolor Jargalsaikhan is an applied mathematician and consultant at ORTEC. She is passionate about solving problems faced by industries using optimization. This paper was developed while she was a postdoc at the University of Groningen, Netherlands.

Kees Jan Roodbergen is a professor of quantitative logistics at the University of Groningen, Netherlands. His research interests include e-commerce logistics, warehouse optimization, vehicle routing, and inventory management.

Leandro Coelho is a professor of logistics optimization at Université Laval, Canada, and the Canada research chair in integrated logistics. His interests include logistics and vehicle routing problems. This paper was developed while he was a visiting professor at the University of Groningen, Netherlands.